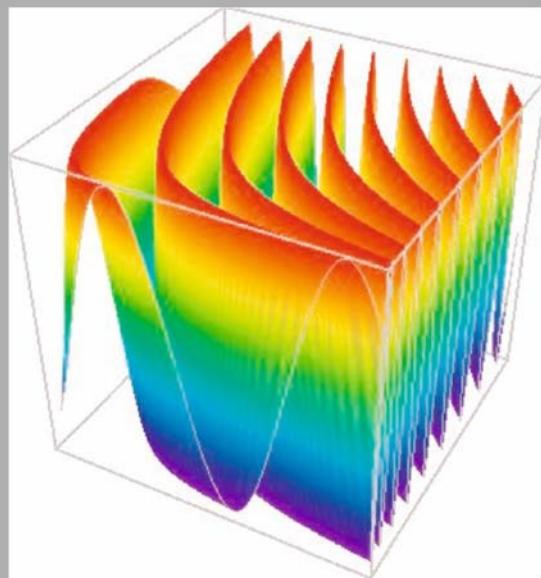


TEXTS IN COMPUTER SCIENCE

Explorations in Quantum Computing



Colin P. Williams

SECOND EDITION

 Springer

Texts in Computer Science

Editors

David Gries

Fred B. Schneider

For further volumes:

<http://www.springer.com/series/3191>

Colin P. Williams

Explorations in Quantum Computing

Second edition



Springer

Dr. Colin P. Williams
California Institute of Technology
NASA Jet Propulsion Laboratory
Oak Grove Drive 4800
Pasadena, CA 91109-8099
USA
Colin.P.Williams@jpl.nasa.gov

Series Editors

David Gries
Department of Computer Science
Upson Hall
Cornell University
Ithaca, NY 14853-7501, USA

Fred B. Schneider
Department of Computer Science
Upson Hall
Cornell University
Ithaca, NY 14853-7501, USA

ISSN 1868-0941
ISBN 978-1-84628-886-9
DOI 10.1007/978-1-84628-887-6
Springer London Dordrecht Heidelberg New York

e-ISSN 1868-095X
e-ISBN 978-1-84628-887-6

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2010936191

1st edition: © Springer-Verlag New York, Inc. 1998
2nd edition: © Springer-Verlag London Limited 2011
© Springer-Verlag London Limited 2011

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licenses issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The use of registered names, trademarks, etc., in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Cover design: SPI, Puducherry, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my wife and children

Preface

In the decade since the publication of the first edition of “Explorations in Quantum Computing” the field has blossomed into a rich and diverse body of knowledge, and tremendous progress has been made on building functional quantum computer hardware. Yet I find that a discussion of *applications* of quantum computers still remains largely confined to Shor’s algorithm for factoring composite integers and Grover’s algorithm for quantum search. As more and more books have been written on quantum computing this standard presentation has been reinforced, thereby overlooking less well known, but arguably more interesting, applications.

In this edition I have tried to survey the field of quantum computing from a fresh perspective, showing how it can be applied to solve problems in a wide range of technical areas including physics, computer science, mathematics, chemistry, simulation, and finance. For sure, many of the newer quantum algorithms have their roots in Shor’s algorithm or Grover’s algorithm, but I think it is important to appreciate how the daughter algorithms have diverged from their parents. Moreover, there are now several quantum transforms known, such as the quantum wavelet and quantum cosine transforms, which show promising complexity properties and yet await exploitation in practical quantum algorithms. The classical versions of these transforms are of widespread utility in classical computing, especially signal and image processing, and I am optimistic that some fresh attention might stimulate others to find good uses for them.

The second edition is organized around four main parts. Part I addresses the question “What is Quantum Computing?” It provides the mathematical framework and physics concepts needed to understand quantum computing, and introduces the first quantum trick—quantum parallelism—and its use within the Deutsch-Jozsa algorithm. I assume the quantum circuit model but discuss several non-standard 2-qubit gates, such as SWAP^a, iSWAP, and Berkeley B, that lend themselves more easily to implementation than does CNOT. In addition, I describe how to quantify the entangling power of quantum gates, and several techniques for constructing quantum circuits that achieve arbitrary n -qubit unitary, and non-unitary, operators including numerical, algebraic, and re-use methods, as well as specialized tricks for constructing optimal circuits for 2-qubit unitary operators.

Part II addresses the question “What Can you Do With a Quantum Computer?” I begin with Grover’s algorithm for quantum search, and applications thereof to speeding up randomized algorithms and synthesizing arbitrary superpositions. I then review Shor’s algorithm for factoring composite integers and computing discrete logarithms, and show how to apply these to breaking the RSA and elliptic curve public key cryptosystems. This is followed with a look at phase transition phenomena in computation and how to apply the insights gleaned from these studies to characterize the complexity of a nested quantum search I developed with Nicolas Cerf and Lov Grover for solving **NP-Complete** problems. This is followed by chapters on applications of quantum algorithms to quantum simulation, quantum chemistry and mathematics. These three areas have the greatest potential for finding new and important quantum algorithms for solving practical problems.

The second edition also includes a greatly expanded discussion of quantum information theory. In particular, in Part III “What Can you Do with Quantum Information”, I look at the notion of pure versus mixed states, density operators, entanglement, how to quantify it, the partial transpose (for signalling the presence of entanglement), the partial trace (for characterizing part of a larger quantum system), and Schmidt decompositions. I have gone beyond the standard presentations on quantum teleportation and superdense coding, to include less well known but potentially useful protocols such as quantum data compression, universal quantum cloning and universal negation—all with complete quantum circuit descriptions. I again emphasize applications of these protocols. In particular, I describe how quantum teleportation has inspired an entirely new, and very promising, model of quantum computation, and how approximate clones and approximate negated states can be used to determine the exact expectation values of observables of ideal clones and ideal negated states. I then describe the most mature of the quantum technologies—quantum cryptography—and discuss the challenges in integrating quantum cryptography with the commercial secure communications infrastructure. I survey the three main quantum key distribution protocols—Bennett and Brassard’s BB84, Bennett’s B92, and Ekert’s E91 protocols, and how they have been implemented in fiber and free-space systems, and look at the prospects for extending the range of quantum cryptography using quantum repeaters and Earth-to-Space channels.

Finally, the book concludes with Part IV “Towards Practical Quantum Computers” by examining some of the practical issues in designing scalable quantum computers. However, I have elected to focus not on hardware *per se*, for which many excellent texts already exist, but more on reliability and architectural issues. In particular, I describe several techniques for quantum error correction including error reduction by symmetrization, quantum error correcting codes, the optimal 5-qubit code, stabilizer codes, bounds on quantum codes, fault-tolerance and concatenated quantum codes. I end the book by discussing the amazing array of alternative models of quantum computing beyond the quantum circuit model, showing how they are inter-related, and how certain schemes lend themselves naturally to implementation in particular types of quantum computer hardware.

The new edition also includes numerous end-of-chapter exercises. Many of these were field tested on students I taught at Stanford University while teaching my “Introduction to Quantum Computing and Quantum Information Theory” course for

several years. In so doing, I learned first hand which concepts students found most difficult. Moreover, in teaching these classes and elsewhere I have learned that quantum physics appeals to many people who might not otherwise have much interest in science. For example, Playboy Playmate Carmen Elektra has been quoted as saying “*I’m really into quantum physics. Some of my friends are into it, some of them aren’t, so I’m trying to get them excited about discovering all these interesting things about thoughts and the power of thoughts. It gives me chills thinking about it. It’s fun.*” [169]. Although some of my colleagues have mocked her for saying this, I say bravo Carmen! Quantum physics is indeed an amazing branch of science, which challenges our most foundational assumptions about the nature of reality. It’s a wonderful thing when a scientific field can so electrify someone that they are compelled to seek a deeper understanding. Certainly, experience in teaching to a very diverse student body has encouraged me to explain things as simply as possible in a self-contained volume. And I hope the reader benefits from my more inclusive style. I can certainly say that Carmen Elektra’s interest in matters quantum has at least given me a more arresting answer to the question “Who did you have in mind when you wrote your book?” than is typical of most scholarly texts!

Finally, I would like to thank the people who have helped me make this second edition a reality. First my family for putting up with the countless evenings and weekends I was away from them. And to Wayne Wheeler and Simon Rees of Springer-Verlag for their encouragement, and eternal patience, in seeing the manuscript through to completion. They deserve a very big thank you! In addition, I am indebted to the physicists and computer scientists who have developed the field of quantum computing to what it is today. Many of these people are known to me personally, but some only via their research papers. I hope I have done justice to their research contributions in writing about them. Known personally to me or not, they have all greatly enriched my life via their discoveries and insights.

Colin P. Williams

Contents

Part I What is Quantum Computing?

1	Introduction	3
1.1	Trends in Computer Miniaturization	4
1.2	Implicit Assumptions in the Theory of Computation	7
1.3	Quantization: From Bits to Qubits	8
1.3.1	Ket Vector Representation of a Qubit	9
1.3.2	Superposition States of a Single Qubit	9
1.3.3	Bloch Sphere Picture of a Qubit	11
1.3.4	Reading the Bit Value of a Qubit	15
1.4	Multi-qubit Quantum Memory Registers	17
1.4.1	The Computational Basis	17
1.4.2	Direct Product for Forming Multi-qubit States	19
1.4.3	Interference Effects	20
1.4.4	Entanglement	21
1.5	Evolving a Quantum Memory Register: Schrödinger's Equation	23
1.5.1	Schrödinger's Equation	24
1.5.2	Hamiltonians	24
1.5.3	Solution as a Unitary Evolution of the Initial State	25
1.5.4	Computational Interpretation	26
1.6	Extracting Answers from Quantum Computers	26
1.6.1	Observables in Quantum Mechanics	26
1.6.2	Observing in the Computational Basis	29
1.6.3	Alternative Bases	30
1.6.4	Change of Basis	32
1.6.5	Observing in an Arbitrary Basis	34
1.7	Quantum Parallelism and the Deutsch-Jozsa Algorithm	35
1.7.1	The Problem: Is $f(x)$ Constant or Balanced?	36
1.7.2	Embedding $f(x)$ in a Quantum Black-Box Function	37
1.7.3	Moving Function Values Between Kets and Phase Factors	38
1.7.4	Interference Reveals the Decision	39
1.7.5	Generalized Deutsch-Jozsa Problem	40

1.8	Summary	44
1.9	Exercises	45
2	Quantum Gates	51
2.1	Classical Logic Gates	52
2.1.1	Boolean Functions and Combinational Logic	52
2.1.2	Irreversible Gates: AND and OR	53
2.1.3	Universal Gates: NAND and NOR	55
2.1.4	Reversible Gates: NOT, SWAP, and CNOT	57
2.1.5	Universal Reversible Gates: FREDKIN and TOFFOLI . .	60
2.1.6	Reversible Gates Expressed as Permutation Matrices . .	61
2.1.7	Will Future Classical Computers Be Reversible?	63
2.1.8	Cost of Simulating Irreversible Computations Reversibly	64
2.1.9	Ancillae in Reversible Computing	66
2.2	Universal Reversible Basis	67
2.2.1	Can All Boolean Circuits Be Simulated Reversibly? . .	68
2.3	Quantum Logic Gates	69
2.3.1	From Quantum Dynamics to Quantum Gates	70
2.3.2	Properties of Quantum Gates Arising from Unitarity . .	71
2.4	1-Qubit Gates	71
2.4.1	Special 1-Qubit Gates	71
2.4.2	Rotations About the x -, y -, and z -Axes	76
2.4.3	Arbitrary 1-Qubit Gates: The Pauli Decomposition . . .	81
2.4.4	Decomposition of R_x Gate	83
2.5	Controlled Quantum Gates	83
2.5.1	Meaning of a “Controlled” Gate in the Quantum Context	85
2.5.2	Semi-Classical Controlled Gates	86
2.5.3	Multiply-Controlled Gates	87
2.5.4	Circuit for Controlled- U	87
2.5.5	Flipping the Control and Target Qubits	90
2.5.6	Control-on- $ 0\rangle$ Quantum Gates	90
2.5.7	Circuit for Controlled-Controlled- U	91
2.6	Universal Quantum Gates	92
2.7	Special 2-Qubit Gates	94
2.7.1	CSIGN, $SWAP^\alpha$, iSWAP, Berkeley B	95
2.7.2	Interrelationships Between Types of 2-Qubit Gates . .	97
2.8	Entangling Power of Quantum Gates	100
2.8.1	“Tangle” as a Measure of the Entanglement Within a State	101
2.8.2	“Entangling Power” as the Mean Tangle Generated by a Gate	103
2.8.3	CNOT from any Maximally Entangling Gate	106
2.8.4	The Magic Basis and Its Effect on Entangling Power . .	106
2.9	Arbitrary 2-Qubit Gates: The Krauss-Cirac Decomposition .	107
2.9.1	Entangling Power of an Arbitrary 2-Qubit Gate	109

2.9.2	Circuit for an Arbitrary Real 2-Qubit Gate	110
2.9.3	Circuit for an Arbitrary Complex 2-Qubit Gate	111
2.9.4	Circuit for an Arbitrary 2-Qubit Gate Using $SWAP^\alpha$	111
2.10	Summary	112
2.11	Exercises	113
3	Quantum Circuits	123
3.1	Quantum Circuit Diagrams	123
3.2	Computing the Unitary Matrix for a Given Quantum Circuit	124
3.2.1	Composing Quantum Gates in Series: The Dot Product	126
3.2.2	Composing Quantum Gates in Parallel: The Direct Product	127
3.2.3	Composing Quantum Gates Conditionally: The Direct Sum	128
3.2.4	Measures of Quantum Circuit Complexity	130
3.3	Quantum Permutations	131
3.3.1	Qubit Reversal Permutation: P_{2^n}	131
3.3.2	Qubit Cyclic Left Shift Permutation: Π_{2^n}	135
3.3.3	Amplitude Downshift Permutation: Q_{2^n}	137
3.3.4	Quantum Permutations for Classical Microprocessor Design?	139
3.4	Quantum Fourier Transform: QFT	140
3.4.1	Continuous Signals as Sums of Sines and Cosines	141
3.4.2	Discrete Signals as Samples of Continuous Signals	142
3.4.3	Discrete Signals as Superpositions	144
3.4.4	QFT of a Computational Basis State	145
3.4.5	QFT of a Superposition	147
3.4.6	QFT Matrix	148
3.4.7	QFT Circuit	150
3.5	Quantum Wavelet Transform: QWT	151
3.5.1	Continuous Versus Discrete Wavelet Transforms	152
3.5.2	Determining the Values of the Wavelet Filter Coefficients	154
3.5.3	Factorization of Daubechies $D_{2^n}^{(4)}$ Wavelet Kernel	157
3.5.4	Quantum Circuit for $D_{2^n}^{(4)}$ Wavelet Kernel	158
3.5.5	Quantum Circuit for the Wavelet Packet Algorithm	158
3.5.6	Quantum Circuit Wavelet Pyramidal Algorithm	160
3.6	Quantum Cosine Transform: QCT	162
3.6.1	Signals as Sums of Cosines Only	163
3.6.2	Discrete Cosine Transform DCT-II and Its Relation to DFT	163
3.6.3	QCT_N^{II} Transformation	165
3.6.4	QCT_N^{II} Matrix	165
3.6.5	QCT_N^{II} Circuit	166
3.7	Circuits for a Arbitrary Unitary Matrices	172
3.7.1	Uses of Quantum Circuit Decompositions	173

3.7.2	Choice of Which Gate Set to Use	173
3.7.3	Circuit Complexity to Implement Arbitrary Unitary Matrices	173
3.7.4	Algebraic Method	174
3.7.5	Simplification via Rewrite Rules	178
3.7.6	Numerical Method	180
3.7.7	Re-use Method	184
3.8	Probabilistic Non-unitary Quantum Circuits	190
3.8.1	Hamiltonian Built from Non-unitary Operator	191
3.8.2	Unitary Embedding of the Non-unitary Operator	191
3.8.3	Non-unitarily Transformed Density Matrix	191
3.8.4	Success Probability	193
3.8.5	Fidelity when Successful	193
3.9	Summary	194
3.10	Exercises	195
4	Quantum Universality, Computability, & Complexity	201
4.1	Models of Computation	202
4.1.1	The Inspiration Behind Turing’s Model of Computation: The <i>Entscheidungsproblem</i>	202
4.1.2	Deterministic Turing Machines	204
4.1.3	Probabilistic Turing Machines	205
4.1.4	The Alternative Gödel, Church, and Post Models	207
4.1.5	Equivalence of the Models of Computation	208
4.2	Universality	208
4.2.1	The Strong Church-Turing Thesis	208
4.2.2	Quantum Challenge to the Strong Church-Turing Thesis .	209
4.2.3	Quantum Turing Machines	210
4.3	Computability	213
4.3.1	Does Quantum Computability Offer Anything New? . . .	214
4.3.2	Decidability: Resolution of the <i>Entscheidungsproblem</i> .	215
4.3.3	Proof Versus Truth: Gödel’s Incompleteness Theorem .	217
4.3.4	Proving Versus Providing Proof	218
4.4	Complexity	221
4.4.1	Polynomial Versus Exponential Growth	223
4.4.2	Big \mathcal{O} , Θ and Ω Notation	225
4.4.3	Classical Complexity Zoo	225
4.4.4	Quantum Complexity Zoo	229
4.5	What Are Possible “Killer-Aps” for Quantum Computers? . . .	233
4.6	Summary	234
4.7	Exercises	235

Part II What Can You Do with a Quantum Computer?

5	Performing Search with a Quantum Computer	241
5.1	The Unstructured Search Problem	242

5.1.1	Meaning of the Oracle	243
5.2	Classical Solution: Generate-and-Test	244
5.3	Quantum Solution: Grover's Algorithm	245
5.4	How Does Grover's Algorithm Work?	247
5.4.1	How Much Amplitude Amplification Is Needed to Ensure Success?	248
5.4.2	An Exact Analysis of Amplitude Amplification	249
5.4.3	The Oracle in Amplitude Amplification	250
5.5	Quantum Search with Multiple Solutions	251
5.5.1	Amplitude Amplification in the Case of Multiple Solutions	252
5.6	Can Grover's Algorithm Be Beaten?	254
5.7	Some Applications of Quantum Search	255
5.7.1	Speeding Up Randomized Algorithms	255
5.7.2	Synthesizing Arbitrary Superpositions	256
5.8	Quantum Searching of Real Databases	260
5.9	Summary	261
5.10	Exercises	262
6	Code Breaking with a Quantum Computer	263
6.1	Code-Making and Code-Breaking	264
6.1.1	Code-Breaking: The Enigma Code and Alan Turing . .	265
6.2	Public Key Cryptosystems	267
6.2.1	The RSA Public-Key Cryptosystem	267
6.2.2	Example of the RSA Cryptosystem	271
6.3	Shor's Factoring Algorithm for Breaking RSA Quantumly . .	272
6.3.1	The Continued Fraction Trick at the End of Shor's Algorithm	276
6.3.2	Example Trace of Shor's Algorithm	280
6.4	Breaking Elliptic Curve Cryptosystems with a Quantum Computer	285
6.5	Breaking DES with a Quantum Computer	287
6.6	Summary	289
6.7	Exercises	290
7	Solving NP-Complete Problems with a Quantum Computer	293
7.1	Importance and Ubiquity of NP-Complete Problems	295
7.1.1	Worst Case Complexity of Solving NP-Complete Problems	296
7.2	Physics-Inspired View of Computational Complexity	297
7.2.1	Phase Transition Phenomena in Physics	297
7.2.2	Phase Transition Phenomena in Mathematics	299
7.2.3	Computational Phase Transitions	299
7.2.4	Where Are the <i>Really</i> Hard Problems?	302
7.3	Quantum Algorithms for NP-Complete Problems	302

7.3.1	Quantum Solution Using Grover's Algorithm	303
7.3.2	Structured Search Spaces: Trees and Lattices	304
7.4	Quantum Solution Using Nested Grover's Algorithm	308
7.4.1	The Core Quantum Algorithm	308
7.4.2	Analysis of Quantum Structured Search	309
7.4.3	Quantum Circuit for Quantum Structured Search	312
7.4.4	Quantum Average-Case Complexity	312
7.5	Summary	316
7.6	Exercises	316
8	Quantum Simulation with a Quantum Computer	319
8.1	Classical Computer Simulations of Quantum Physics	320
8.1.1	Exact Simulation and the Problem of Memory	321
8.1.2	Exact Simulation and the Problem of Entanglement	321
8.1.3	Approximate Simulation and the Problem of Fidelity	322
8.2	Quantum Computer Simulations of Quantum Physics	325
8.2.1	Feynman Conceives of a Universal Quantum Simulator	326
8.2.2	Quantum Systems with Local Interactions	326
8.2.3	Lloyd-Zalka-Wiesner Quantum Simulation Algorithm	327
8.3	Extracting Results from Quantum Simulations Efficiently	328
8.3.1	Single Ancilla-Assisted Readout	328
8.3.2	Multi-Ancilla-Assisted Readout	330
8.3.3	Tomography Versus Spectroscopy	332
8.3.4	Evaluating Correlation Functions	333
8.4	Fermionic Simulations on Quantum Computers	334
8.4.1	Indistinguishability and Implications for Particle Statistics	334
8.4.2	Symmetric Versus Anti-Symmetric State Vectors	335
8.4.3	Bosons and Fermions	336
8.4.4	Bose-Einstein Statistics	337
8.4.5	Pauli Exclusion Principle and Fermi-Dirac Statistics	337
8.4.6	Fermionic Simulations via the Jordan-Wigner Transformation	339
8.4.7	Fermionic Simulations via Transformation to Non-interacting Hamiltonians	341
8.5	Summary	344
8.6	Exercises	345
9	Quantum Chemistry with a Quantum Computer	349
9.1	Classical Computing Approach to Quantum Chemistry	349
9.1.1	Classical Eigenvalue Estimation via the Lanczos Algorithm	351
9.2	Quantum Eigenvalue Estimation via Phase Estimation	352
9.2.1	The "Phase" State	352
9.2.2	Binary Fraction Representation of the Phase Factor	353
9.3	Quantum Phase Estimation	354

9.4	Eigenvalue Kick-Back for Synthesizing the Phase State	357
9.5	Quantum Eigenvalue Estimation Algorithms	361
9.5.1	Abrams-Lloyd Eigenvalue Estimation Algorithm	361
9.5.2	Kitaev Eigenvalue Estimation Algorithm	361
9.6	Quantum Chemistry Beyond Eigenvalue Estimation	364
9.7	Summary	364
9.8	Exercises	365
10	Mathematics on a Quantum Computer	369
10.1	Quantum Functional Analysis	369
10.1.1	Quantum Mean Estimation	370
10.1.2	Quantum Counting	371
10.2	Quantum Algebraic Number Theory	375
10.2.1	The Cattle Problem of Archimedes and Pell’s Equation	375
10.2.2	Why Solving Pell’s Equation Is Hard	376
10.2.3	Solution by Finding the “Regulator”	377
10.2.4	The Regulator and Period Finding	378
10.2.5	Quantum Core of Hallgren’s Algorithm	378
10.2.6	Hallgren’s Quantum Algorithm for Solving Pell’s Equation	378
10.2.7	What Is the Significance of Pell’s Equation?	381
10.3	Quantum Signal, Image, and Data Processing	382
10.3.1	Classical-to-Quantum Encoding	382
10.3.2	Quantum Image Processing: 2D Quantum Transforms	384
10.3.3	Quantum-to-Classical Readout	385
10.4	Quantum Walks	385
10.4.1	One-Dimensional Quantum Walks	387
10.4.2	Example: Biased Initial Coin State & Hadamard Coin	389
10.4.3	Example: Symmetric Initial Coin State & Hadamard Coin	391
10.4.4	Example: Chiral Initial Coin State & Hadamard Coin	392
10.4.5	Example: Symmetric Initial Coin State & Non-Hadamard Coin	393
10.4.6	Quantum Walks Can Spread Faster than Classical Walks	395
10.5	Summary	397
10.6	Exercises	398

Part III What Can You Do with Quantum Information?

11	Quantum Information	403
11.1	What is Classical Information?	404
11.1.1	Classical Sources: The Shannon Entropy	405
11.1.2	Maximal Compression (Source Coding Theorem)	407
11.1.3	Reliable Transmission (Channel Coding Theorem)	408
11.1.4	Unstated Assumptions Regarding Classical Information .	410

11.2	What is Quantum Information?	411
11.2.1	Pure States cf. Mixed States	411
11.2.2	Mixed States from Partial Knowledge: The Density Operator	411
11.2.3	Mixed States from Partial Ignorance: The Partial Trace .	417
11.2.4	Mixed States as Parts of Larger Pure States: “Purifications”	419
11.2.5	Quantifying Mixedness	420
11.3	Entanglement	422
11.3.1	Separable States Versus Entangled States	422
11.3.2	Signalling Entanglement via Entanglement Witnesses .	423
11.3.3	Signalling Entanglement via the Peres-Horodecki Criterion	425
11.3.4	Quantifying Entanglement	429
11.3.5	Maximally Entangled Pure States	431
11.3.6	Maximally Entangled Mixed States	432
11.3.7	The Schmidt Decomposition of a Pure Entangled State .	433
11.3.8	Entanglement Distillation	436
11.3.9	Entanglement Swapping	441
11.3.10	Entanglement in “Warm” Bulk Matter	443
11.4	Compressing Quantum Information	444
11.4.1	Quantum Sources: The von Neumann Entropy	445
11.4.2	Schumacher-Jozsa Quantum Data Compression	445
11.4.3	“Discard-on-Fail” Quantum Data Compression Protocol .	447
11.4.4	“Augment-on-Fail” Quantum Data Compression Protocol	449
11.4.5	Quantum Circuit for Schumacher-Jozsa Compressor .	450
11.4.6	Is Exponential Compression Possible?	452
11.5	Superdense Coding	453
11.5.1	Bell States	454
11.5.2	Interconversion Between Bell States by Local Actions .	455
11.5.3	Superdense Coding Protocol	455
11.6	Cloning Quantum Information	457
11.6.1	Historical Roots and Importance of Quantum Cloning .	457
11.6.2	Impossibility of Exact Deterministic Quantum Cloning .	458
11.6.3	Universal Approximate Quantum Cloning	460
11.6.4	Circuit for Quantum Cloning	463
11.6.5	Usability of the Quantum Clones	464
11.6.6	Universal Probabilistic Quantum Cloning	468
11.6.7	Broadcasting Quantum Information	470
11.7	Negating Quantum Information	470
11.7.1	Universal Quantum Negation Circuit	471
11.7.2	Expectation Value of an Observable Based on the Negated State	472
11.8	Summary	472
11.9	Exercises	474

12	Quantum Teleportation	483
12.1	Uncertainty Principle and “Impossibility” of Teleportation	483
12.1.1	Heisenberg Uncertainty Principle	484
12.2	Principles of True Teleportation	486
12.2.1	Local Versus Non-local Interactions	486
12.2.2	Non-locality: Einstein’s “Spooky Action at a Distance”	488
12.2.3	Bell’s Inequality	489
12.3	Experimental Tests of Bell’s Inequality	492
12.3.1	Speed of Non-local Influences	494
12.4	Quantum Teleportation Protocol	496
12.4.1	Teleportation Does Not Imply Superluminal Communication	499
12.5	Working Prototypes	500
12.6	Teleporting Larger Objects	501
12.7	Summary	502
12.8	Exercises	503
13	Quantum Cryptography	507
13.1	Need for Stronger Cryptography	508
13.1.1	Satellite Communications Can Be Tapped	508
13.1.2	Fiber-Optic Communications Can Be Tapped	510
13.1.3	Growing Regulatory Pressures for Heightened Security	512
13.1.4	Archived Encrypted Messages Retroactively Vulnerable	512
13.2	An Unbreakable Cryptosystem: The One Time Pad	515
13.2.1	Security of OTP: Loopholes if Used Improperly	518
13.2.2	Practicality of OTP: Problem of Key Distribution	519
13.3	Quantum Key Distribution	520
13.3.1	Concept of QKD	520
13.3.2	Security Foundations of QKD	520
13.3.3	OTP Made Practical by QKD	521
13.3.4	Varieties of QKD	521
13.4	Physics Behind Quantum Key Distribution	522
13.4.1	Photon Polarization	522
13.4.2	Single Photon Sources	523
13.4.3	Entangled Photon Sources	524
13.4.4	Creating Truly Random Bits	525
13.4.5	Encoding Keys in Polarized Photons	526
13.4.6	Measuring Photon Polarization with a Birefringent Crystal	528
13.4.7	Measuring Photon Polarization with a Polarizing Filter	529
13.5	Bennett and Brassard’s BB84 QKD Scheme	529
13.5.1	The BB84 QKD Protocol	531
13.5.2	Example: BB84 QKD in the Absence of Eavesdropping	534
13.5.3	Example: BB84 QKD in the Presence of Eavesdropping	536
13.5.4	Spedalieri’s Orbital Angular Momentum Scheme for BB84	537

13.5.5	Generalization of BB84: Bruss' 6-State Protocol	538
13.6	Bennett's 2-State Protocol (B92)	539
13.6.1	The B92 QKD Protocol	539
13.6.2	Threat of “Discard-on-Fail” Unambiguous State Discrimination	540
13.7	Ekert’s Entanglement-Based Protocol	541
13.7.1	The E91 Protocol	541
13.8	Error Reconciliation and Privacy Amplification	542
13.8.1	Error Reconciliation	543
13.8.2	Privacy Amplification	544
13.9	Implementations of Quantum Cryptography	545
13.9.1	Fiber-Optic Implementations of Quantum Cryptography .	545
13.9.2	Extending the Range of QKD with Quantum Repeaters .	547
13.9.3	Earth-to-Space Quantum Cryptography	548
13.9.4	Hijacking Satellites	550
13.9.5	Commercial Quantum Cryptography Systems	554
13.10	Barriers to Widespread Adoption of Quantum Cryptography .	555
13.10.1	Will People Perceive a Need for Stronger Cryptography?	555
13.10.2	Will People Believe the Foundations of QKD Are Solid?	556
13.10.3	Will People Trust the Warranties of Certification Agencies?	556
13.10.4	Will Wide Area Quantum Cryptography Networks Be Practical?	557
13.10.5	Will Key Generation Rate Be High Enough to Support OTP?	558
13.10.6	Will Security Be the Dominant Concern?	558
13.11	Summary	558
13.12	Exercises	560

Part IV Towards Practical Quantum Computers

14	Quantum Error Correction	567
14.1	How Errors Arise in Quantum Computing	568
14.1.1	Dissipation-Induced Bit Flip Errors	568
14.1.2	Decoherence-Induced Phase Shift Errors	569
14.1.3	Natural Decoherence Times of Physical Systems	570
14.1.4	What Makes Quantum Error Correction so Hard?	571
14.2	Quantum Error Reduction by Symmetrization	573
14.2.1	The Symmetrization Trick	574
14.2.2	Quantum Circuit for Symmetrization	576
14.2.3	Example: Quantum Error Reduction via Symmetrization .	577
14.3	Principles of Quantum Error Correcting Codes (QECCs)	579
14.3.1	Classical Error Correcting Codes	579

14.3.2	Issues Unique to Quantum Error Correcting Codes	580
14.3.3	Modeling Errors in Terms of Error Operators	581
14.3.4	Protecting Quantum Information via Encoding	583
14.3.5	Digitizing and Diagnosing Errors by Measuring Error Syndromes	585
14.3.6	Reversing Errors via Inverse Error Operators	585
14.3.7	Abstract View of Quantum Error Correcting Codes	585
14.4	Optimal Quantum Error Correcting Code	588
14.4.1	Laflamme-Miquel-Paz-Zurek's 5-Qubit Code	588
14.4.2	Error Operators for the 5-Qubit Code	588
14.4.3	Encoding Scheme for the 5-Qubit Code	589
14.4.4	Error Syndromes & Corrective Actions for the 5-Qubit Code	591
14.4.5	Example: Correcting a Bit-Flip	592
14.5	Other Additive Quantum Error Correcting Codes	593
14.5.1	Shor's 9-Qubit Code	593
14.5.2	Steane's 7-Qubit Code	594
14.6	Stabilizer Formalism for Quantum Error Correcting Codes	594
14.6.1	Group Theory for Stabilizer Codes	595
14.6.2	The Stabilizer	595
14.6.3	Example: A Stabilizer for the 5-Qubit Code	596
14.6.4	Using a Stabilizer to Find the Codewords It Stabilizes	597
14.6.5	How the Stabilizer is Related to the Error Operators	599
14.6.6	Example: Stabilizers and Error Operators for the 5-Qubit Code	600
14.6.7	Stabilizer-Based Error Correction: The Encoding Step	603
14.6.8	Stabilizer-Based Error Correction: Introduction of the Error	603
14.6.9	Stabilizer-Based Error Correction: Error Diagnosis & Recovery	603
14.6.10	Stabilizers for Other Codes	604
14.7	Bounds on Quantum Error Correcting Codes	605
14.7.1	Quantum Hamming Bound	606
14.7.2	Quantum Singleton Bound	606
14.7.3	Quantum Gilbert-Varshamov Bound	607
14.7.4	Predicting Upper and Lower Bounds on Additive Codes	607
14.7.5	Tightest Proven Upper and Lower Bounds on Additive Codes	611
14.8	Non-additive (Non-stabilizer) Quantum Codes	611
14.9	Fault-Tolerant Quantum Error Correcting Codes	611
14.9.1	Concatenated Codes and the Threshold Theorem	617
14.10	Errors as Allies: Noise-Assisted Quantum Computing	620
14.11	Summary	621
14.12	Exercises	622

15 Alternative Models of Quantum Computation	627
15.1 Design Principles for a Quantum Computer	627
15.2 Distributed Quantum Computer	628
15.3 Quantum Cellular Automata Model	630
15.4 Measurement I: Teleportation-Based Quantum Computer	633
15.5 Measurement II: One-Way Quantum Computer	640
15.6 Topological Quantum Computer	641
15.6.1 Topological Quantum Effects	642
15.6.2 Beyond Fermions and Bosons—Anyons	643
15.6.3 Abelian Versus Non-Abelian Anyons	644
15.6.4 Quantum Gates by Braiding Non-Abelian Anyons	644
15.6.5 Do Non-Abelian Anyons Exist?	649
15.7 Adiabatic Quantum Computing	649
15.8 Encoded Universality Using Only Spin-Spin Exchange Interactions	653
15.8.1 The Exchange Interaction	653
15.8.2 SWAP $^\alpha$ via the Exchange Interaction	654
15.8.3 Problem: Although SWAP $^\alpha$ Is Easy 1-Qubits Gates Are Hard	655
15.8.4 Solution: Use an Encoded Basis	655
15.8.5 $U_{\mathcal{L}}^{1,2}$, $U_{\mathcal{L}}^{2,3}$, and $U_{\mathcal{L}}^{1,3}$	656
15.8.6 R_z Gates in Encoded Basis	657
15.8.7 R_x Gates in Encoded Basis	657
15.8.8 R_y Gates in Encoded Basis	658
15.8.9 CNOT in Encoded Basis	658
15.9 Equivalences Between Alternative Models of Quantum Computation	659
15.10 Summary	660
15.11 Exercises	660
References	663
Index	689

Part I

What is Quantum Computing?

Chapter 1

Introduction

“The theory of computation has traditionally been studied almost entirely in the abstract, as a topic in pure mathematics. This is to miss the point of it. Computers are physical objects, and computations are physical processes. What computers can or cannot compute is determined by the laws of physics alone, and not by pure mathematics”

– David Deutsch¹

Over the past 50 years there has been an astonishing miniaturization in computer technology. Whereas a microprocessor in 1971 contained roughly 2,300 transistors, a modern microprocessor of the same size contains in excess of one billion transistors. Throughout this evolution, even though there have been several changes in how computer hardware is implemented, the *same* underlying mathematical model of a computer has held sway. However, if current trends continue, by the year 2020 the basic components of a computer will be the size of individual atoms. At such scales, the mathematical theory underpinning modern computer science will cease to be valid. Instead, scientists are inventing a new theory, called “quantum computing”, which is built upon the recognition that a computing device is a *physical* system governed by *physical* laws, and at very small scales, the appropriate laws are those of quantum mechanics—the most accurate model of reality that is currently known.

There are two attitudes one could adopt regarding the necessity of incorporating quantum mechanical effects into computing machinery. One response is to strive to suppress the quantum effects and still preserve a semblance of classicality even though the computational elements are very small. The other approach is to embrace quantum effects and try to find clever ways to enhance and sustain them to achieve old computational goals in new ways. Quantum computing attempts to pursue the latter strategy by *harnessing* quintessentially quantum effects.

Remarkably, this new theory of quantum computer science predicts that quantum computers will be able to perform certain computational tasks in phenomenally

¹Source: Opening words of Chap. 5, “Virtual Reality” of “The Fabric of Reality,” by David Deutsch, the Penguin Press (1997), ISBN 0-7139-9061-9.

fewer steps than *any* conventional (“classical”) computer—including any supercomputer yet to be invented! This bold assertion is justified because the algorithms available to quantum computers can harness physical phenomena that are not available to classical computers no matter how sophisticated they may be. As a result, quantum computers can perform computations in fundamentally new ways that can, at best, only be mimicked inefficiently by classical computers. Thus, quantum computing represents a *qualitative* change in how computation is done, making it of a different character than all previous advances in computer science. In particular, quantum computers can perform truly unprecedented tasks such as teleporting information, breaking supposedly “unbreakable” codes, generating true random numbers, and communicating with messages that betray the presence of eavesdropping. Similar counterintuitive capabilities are being discovered, routinely, making quantum computing a very active and exciting field. While no one book can do justice to the myriad of discoveries that have been made so far, I hope to give you a fresh perspective on the capabilities of quantum computers, and to provide you with the tools necessary to make your own foray into this exciting field.

1.1 Trends in Computer Miniaturization

“I like small gadgets, look at this tiny digital camera . . . where is it?”

– Artur Ekert [17]

Computer technology has been driven to smaller and smaller scales because, ultimately, the limiting factor on the speed of microprocessors is the speed with which information can be moved around inside the device. By cramming the transistors closer together, and evolving to ever faster mechanisms for switching, one can speed up the rate of computation. But there is a price to pay. As transistors are packed closer together it becomes more challenging to remove the heat they dissipate. So at any given stage of technological development there has always been an optimal transistor density that trades off size for thermal management.

In 1965 Gordon Moore, a co-founder of Intel, noticed that the most economically favorable transistor densities in integrated circuits seemed to have been doubling roughly every 18 months. He predicted that this trend would continue well into the future. Indeed, as evidenced by Table 1.1, it has, and Moore’s anticipated scaling became known as the more official sounding “Moore’s Law”. However, it is not a Law in the proper scientific sense as Nature does not *enforce* it. Rather, Moore’s Law is merely an empirical observation of a scaling regularity in transistor size and power dissipation that industry had achieved, and Gordon Moore extrapolated into the future. However, there is uncertainty in the chip industry today regarding how much longer Moore’s Law can be sustained.

Nevertheless, in the 40 years since Moore’s Law was invented, successive generations of Intel chips have adhered to it surprisingly. This is all the more surprising when one realizes how just how much the underlying transistor technology has changed (see Fig. 1.1).

Table 1.1 Growth of the clock rate, and the number of transistors per chip in Intel processors from 1971 to 2007. Note that the transistor sizes reduced over the same time period, allowing the chips to remain about the same size. In the table $1 \mu = 10^{-6}$ meter and $1 \text{ nm} = 10^{-9}$ meter

Intel microprocessor	Year	Speed	# Transistors	Manufacturing scale
4004	1971	108 kHz	2,300	10μ
8008	1972	500–800 kHz	3,500	10μ
8080	1974	2 MHz	4,500	6μ
8086	1978	5 MHz	29,000	3μ
8088	1979	5 MHz	29,000	3μ
286	1982	6 MHz	134,000	1.5μ
386	1985	16 MHz	275,000	1.5μ
486	1989	25 MHz	1,200,000	1μ
Pentium	1993	66 MHz	3,100,000	0.8μ
Pentium Pro	1995	200 MHz	5,500,000	0.6μ
Pentium II	1997	300 MHz	7,500,000	0.25μ
Pentium II Xeon	1997	300 MHz	7,500,000	0.25μ
Pentium III	1999	500 MHz	9,500,000	0.18μ
Pentium III Xeon	1999	500 MHz	9,500,000	0.18μ
Pentium 4	2000	1.5 GHz	42,000,000	0.18μ
Xeon	2001	1.5 GHz	42,000,000	0.18μ
Pentium M	2002	1.7 GHz	55,000,000	90 nm
Itanium 2	2002	1 GHz	220,000,000	0.13μ
Pentium D	2005	3.2 GHz	291,000,000	65 nm
Core 2 Duo	2006	2.93 GHz	291,000,000	65 nm
Core 2 Extreme	2006	2.93 GHz	291,000,000	65 nm
Dual-Core Xeon	2006	2.93 GHz	291,000,000	65 nm
Dual-Core Itanium 2	2006	1.66 GHz	1,720,000,000	90 nm
Quad-Core Xeon	2006	2.66 GHz	582,000,000	65 nm
Quad-Core Core 2 Extreme	2006	2.66 GHz	582,000,000	65 nm
Core 2 Quad	2007	2.66 GHz	582,000,000	65 nm
Quad-Core Xeon	2007	>3 GHz	820,000,000	45 nm
Dual-Core Xeon	2007	>3 GHz	820,000,000	45 nm
Quad-Core Core 2 Extreme	2007	>3 GHz	820,000,000	45 nm

Today, many industry insiders see Moore's Law surviving for just two or three more generations of microprocessors at best. In a valiant effort to sustain Moore's Law chip manufacturers are migrating to multi-core microprocessor architectures, and exotic new semiconductor materials. Beyond these advances, a switch to nanotechnology may be necessary.

Whatever strategy industry adopts to maintain Moore's Law it is clear that as time goes on fewer and fewer atoms will be used to implement more and more bits.

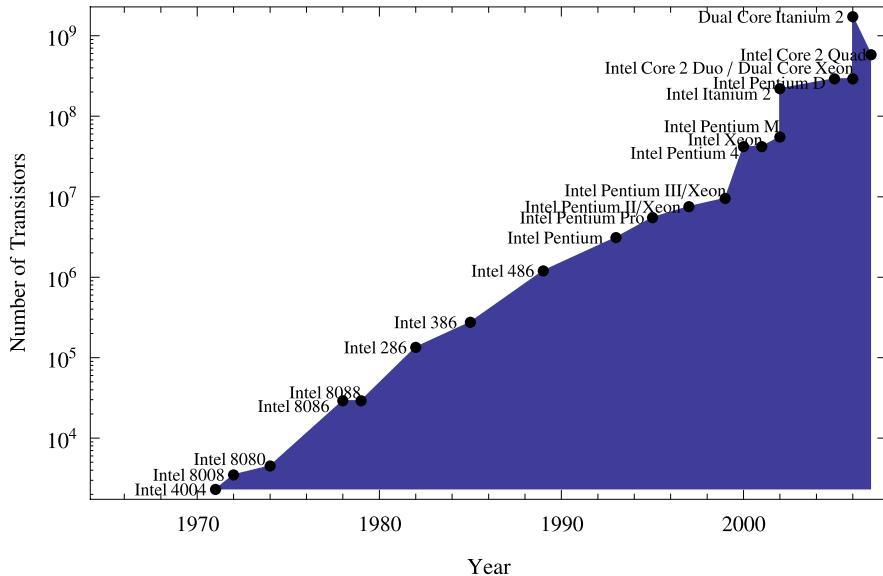


Fig. 1.1 Historical scaling in the numbers of transistors per chip in successive generations of Intel processors. The latest chips use multiple cores

Fig. 1.2 Historical scaling in the number of atoms needed to implement one bit

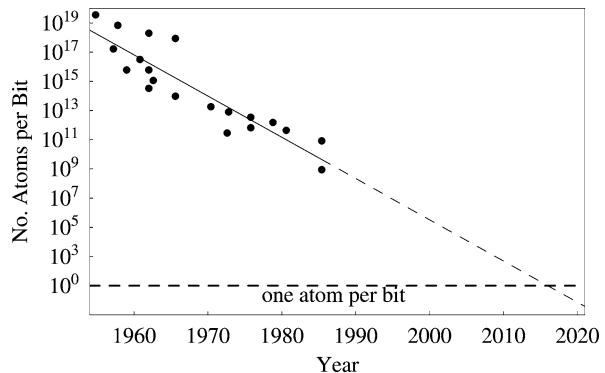


Figure 1.2 shows the scaling in the number of atoms needed to implement a bit as a function of time. Extrapolating this trend shows we will be at the one atom per bit level by about 2020. At the one-atom-per-bit level the appropriate physical model to describe what is going on is that of quantum physics rather than classical physics.

Quantum physics is considerably different from classical physics. Facts that we take as being “common sense” in our everyday (classical) world do not necessarily hold in the quantum realm. For example, in the classical world we are accustomed to thinking of particles (like grains of sand or dust) as having a definite location in space and time. But in the quantum world particles do have a definite location in space and time—in fact they can be in more than one place, or in more than one

state, at the same time! More bizarre still, supposed “particles” can interact with one another more in a manner that is more reminiscent of waves than solid objects. Ultimately, as bits must be encoded in the states of physical systems, whether those systems are quantum or classical can therefore affect their properties profoundly.

1.2 Implicit Assumptions in the Theory of Computation

“Nature isn’t classical damn it!”

– Richard Feynman

Bits, or “binary digits” lie at the heart of all modern digital equipment ranging from computers to iPODs to high-definition television (HDTV). Contemporary computers use voltage levels to encode bits. Old fashioned, mechanical, computers use the position of gear teeth. The only requirement is that the physical system must possess two clearly distinguishable configurations, or states, that are sufficiently stable so that they do not flip, spontaneously, from the state representing the bit 0 into the state representing the bit 1 or vice versa.

Once we have the ability to store 0s and 1s and to manipulate them in a controlled manner we have the basis for making all digital devices. By now, we are all so familiar with digital devices that, to the extent we even think about them at all, we take the properties of the bits within them for granted. For example, I am sure you will agree that the following operations on bits seem eminently reasonable: we can read a bit to learn the value it has; we can copy, erase or negate a bit regardless of whether it is a 0 or a 1; and we can read some of the bits inside a digital device without changing the other bits that we did not read. In fact such properties seem so obvious that we don’t even bother to question these assumptions.

However, in his 1959 address “There’s Plenty of Room at the Bottom” physicist Richard Feynman alluded to the tremendous opportunity available at the time for further miniaturization of technology [182]. He also anticipated that very small physical devices would be governed by quantum mechanics rather than classical mechanics and, as such, would not necessarily behave the same their larger counterparts. For example, a robot on the quantum scale might pick up and not pick up an object at the same time, and to carry it off left and right simultaneously. You would never know which was the case until you performed an observation as to what the robot had done. Once you did that, and made a permanent record of the result, its behavior would become definite. That sounds crazy, but that is what quantum mechanics tells us can happen.

Likewise, bits are going to be recorded, ultimately, in the state of some physical system. So as devices become miniaturized the sizes of the physical systems used to encode those bits will become smaller. At some point their behavior will need to be described by quantum physics rather than classical physics. At this point, our common sense assumptions about how bits ought to behave, e.g., that we can read, copy, erase, negate them without causing them to change in any way, cease to be

Table 1.2 Assumptions about the properties of bit that are no longer necessarily true at the quantum scale

Assumption	Classically	Quantum mechanically
A bit always has a definite value	True	False. A bit need not have a definite value until the moment after it is read
A bit can only be 0 or 1	True	False. A bit can be in a superposition of 0 and 1 simultaneously
A bit can be copied without affecting its value	True	False. A qubit in an unknown state cannot be copied without necessarily changing its quantum state
A bit can be read without affecting its value	True	False. Reading a qubit that is initially in a superposition will change the qubit
Reading one bit in the computer memory has no affect on any other (unread) bit in the memory	True	False. If the bit being read is entangled with another qubit, reading one qubit will affect the other
To compute the result of a computation, you must run the computer	True	False

valid. In fact, at the quantum scale you cannot necessarily read a bit without changing its value; you cannot necessarily copy, or negate it without perturbing it; you may be unable to erase it; and sometimes when you read one bit your actions can change the state of another bit with which you never interacted. Thus, bits encoded in quantum-scale objects cease to behave like normal bits ought. Some of the differences between normal (classical) and bits encoded at the quantum scale are shown in Table 1.2.

Thus, once computers becomes so small that we are then dealing with quantum bits as opposed to classical bits, we open up a new repertoire of physical effects that can be harnessed to achieve novel functionalities. As a result many new opportunities present themselves.

1.3 Quantization: From Bits to Qubits

Fortunately, quantum systems possess certain properties that lend themselves to encoding bits as physical states. When we measure the “spin” of an electron, for example, we always find it to have one of two possible values. One value, called “spin up” or $|\uparrow\rangle$, means that the spin was found to be parallel to the axis along which the measurement was taken. The other possibility, “spin-down” or $|\downarrow\rangle$, means that the spin was found to be anti-parallel to the axis along which the measurement was taken. This intrinsic discreteness, a manifestation of quantization, allows the spin of an electron to be considered as a natural binary digit or “bit”.

Such intrinsic “discreteness” is not unique to spin-systems. Any 2-state quantum system, such as the plane of polarization of a linearly polarized photon, the direction

of rotation of a circularly polarized photon, or the discrete energy levels in an excited atom, would work equally well. Whatever the exact physical embodiment chosen, if a quantum system is used to represent a bit, we call the resulting system a quantum bit, or just “qubit” for short.

1.3.1 Ket Vector Representation of a Qubit

As we are talking variously about (classical) bits and (their quantum counterparts) qubits, we’d better find a way of distinguishing them. To do so, we adopt a notation invented by British physicist extraordinaire Paul Dirac, which has since become known as “Dirac-notation”.

In Dirac notation, when we are talking about a qubit (a quantum bit) in a physical state that represents the bit value 0, we’ll write the qubit state using an angular-looking bracket, $|0\rangle$, which is called a “ket” vector. Likewise, a qubit in a physical state representing the bit value 1 will be written $|1\rangle$. What these notations mean *physically* will depend upon the nature of the system encoding them. For example, a $|0\rangle$ could refer to a polarized photon, or an excited state of an atom, or the direction of circulation of a superconducting current etc. The notation speaks only to the *computational* abstraction that we ascribe to a 2-state quantum system and doesn’t give us any direct information about the underlying physical embodiment of the system encoding that qubit.

Mathematically, kets are a shorthand notation for column vectors, with $|0\rangle$ and $|1\rangle$ corresponding to:

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (1.1)$$

You might ask “Why do we need to represent a *single* quantum bit as a *two-element* column vector?” “Isn’t one binary digit enough to specify it completely?” The answer lies in the fact that quantum bits are not constrained to be wholly 0 or wholly 1 at a given instant. In quantum physics if a quantum system can be found to be in one of a discrete set of states, which we’ll write as $|0\rangle$ or $|1\rangle$, then whenever it is not being observed it may also exist in a *superposition*, or blend of those states simultaneously, $|\psi\rangle = a|0\rangle + b|1\rangle$ such that $|a|^2 + |b|^2 = 1$.

1.3.2 Superposition States of a Single Qubit

Thus, whereas at any instant a classical bit can be either a 0 *or* a 1, a qubit can be a superposition of both a $|0\rangle$ *and* a $|1\rangle$ simultaneously, i.e., a state such as:

$$|\psi\rangle = a|0\rangle + b|1\rangle \equiv \begin{pmatrix} a \\ b \end{pmatrix} \quad (1.2)$$

where a , and b are complex numbers² having the property $|a|^2 + |b|^2 = 1$.

The coefficient “ a ” is called the *amplitude* of the $|0\rangle$ component and the coefficient “ b ” is called the *amplitude* of the $|1\rangle$ component. The requirement that $|a|^2 + |b|^2 = 1$ is to ensure the qubit is properly *normalized*. Proper normalization guarantees that when we do finally read a qubit, it will be found, with probability $|a|^2$ to be in state $|0\rangle$ or, with probability $|b|^2$ to be in state $|1\rangle$ and nothing else. Thus the sums of the probabilities of the possible outcomes add up to one.

Dirac notation makes it easy to write down compact descriptions of quantum states and operators. Some common examples are as follows:

Dirac Notation: Bras, Kets, Inner and Outer Products For every “ket” $|\psi\rangle$ (which can be thought of as a shorthand notation for a column vector) there is a corresponding “bra” $\langle\psi|$ (which can be thought of as shorthand for a row vector). The ket and the bra contain *equivalent information* about the quantum state in question. Mathematically, they are the dual of one another, i.e.:

$$\begin{aligned} |\psi\rangle &= a|0\rangle + b|1\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \\ \langle\psi| &= a^*\langle 0| + b^*\langle 1| = (a^* \ b^*) \end{aligned} \quad (1.3)$$

Note that the amplitudes in the bra space are the complex conjugates of the amplitudes in the ket space. That is, if $z = x + iy$ is a complex number with real part x and imaginary part y , then the complex conjugate of z is $z^* = x - iy$.

What is the purpose of introducing bra vectors into the discussion if they don’t contain any new information about the quantum state? It turns out that *products* of bras and kets give us insight into the similarities between two quantum states. Specifically, for a pair of qubits in states $|\psi\rangle = a|0\rangle + b|1\rangle$ and $|\phi\rangle = c|0\rangle + d|1\rangle$ we can define their *inner product*, $\langle\psi|\phi\rangle$ as:

$$\langle\psi|\phi\rangle = \underbrace{(\langle\psi|) \cdot (\langle\phi|)}_{\text{bra (c) ket}} = (a^* \ b^*) \cdot \begin{pmatrix} c \\ d \end{pmatrix} = a^*c + b^*d \quad (1.4)$$

The inner product $\langle\psi|\phi\rangle$ is also called the overlap between (normalized) states $|\psi\rangle$ and $|\phi\rangle$ because it varies from zero for orthogonal states to one for identical normalized states. We can verify this with a direct calculation: $\langle\psi|\psi\rangle = (a^* \ b^*) \cdot \begin{pmatrix} a \\ b \end{pmatrix} = a^*a + b^*b = |a|^2 + |b|^2 = 1$.

A second product we can define on states $|\psi\rangle = a|0\rangle + b|1\rangle$ and $|\phi\rangle = c|0\rangle + d|1\rangle$ is their outer product $|\psi\rangle\langle\phi|$:

$$|\psi\rangle\langle\phi| = (|\psi\rangle) \cdot (\langle\phi|) = \begin{pmatrix} a \\ b \end{pmatrix} \cdot (c^* \ d^*) = \begin{pmatrix} ac^* & ad^* \\ bc^* & bd^* \end{pmatrix} \quad (1.5)$$

²A complex number $z = x + iy$ is a composite number consisting of two real numbers x and y , and a constant $i = \sqrt{-1}$. $x = \text{Re}(z)$ is called the “real” part of z , and $y = \text{Im}(z)$ is called the “imaginary” part of z . $z^* = x - iy$ denotes the complex *conjugate* of z , and $|z| = \sqrt{x^2 + y^2}$ denotes the *modulus* of z .

which is a matrix. The outer product provides a very nice way of describing the structure of unitary operators, which as we will see later, correspond to quantum logic gates. For example, a NOT gate has a corresponding unitary matrix $\text{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. In terms of outer products this can also be written as $\text{NOT} = |0\rangle\langle 1| + |1\rangle\langle 0|$. The outer product factorization of the NOT gate shows the transformation it performs explicitly. Indeed, all quantum gates can be best understood as a sum of such outer products.

1.3.3 Bloch Sphere Picture of a Qubit

An intuitive, albeit approximate, way to visualize the quantum state of a single qubit is to picture it as a unit vector inside a bounding sphere, called the Bloch sphere (see Fig. 1.3). The parameters defining the quantum state are related to the azimuth and

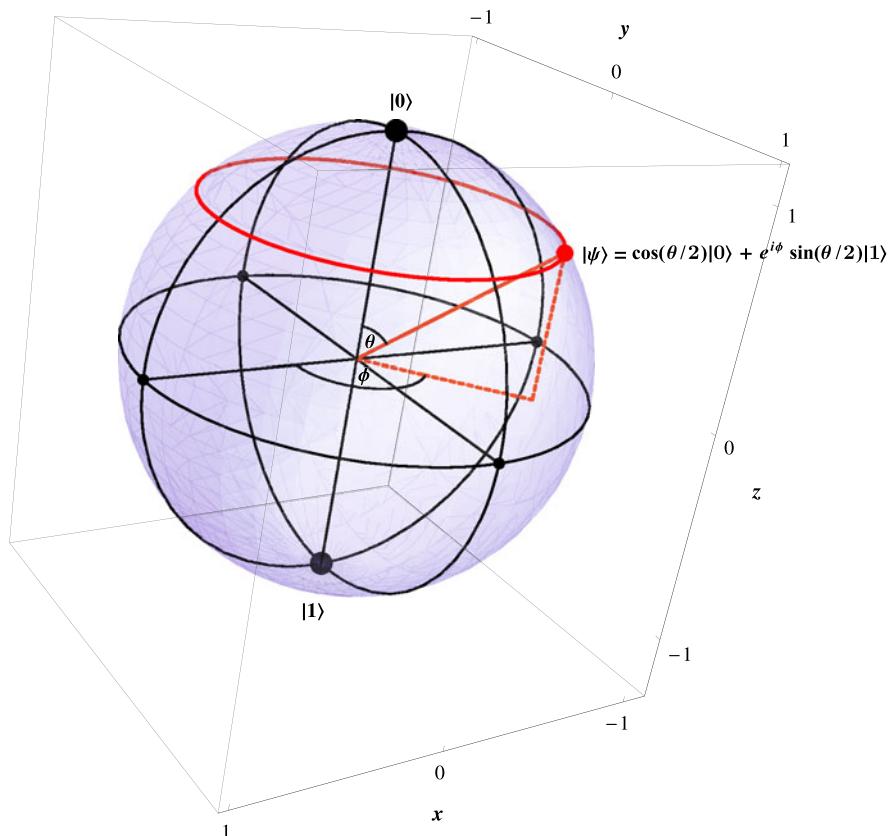


Fig. 1.3 Bloch sphere showing the computational basis states $|0\rangle$ and $|1\rangle$, and a general qubit state $|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi} \sin(\theta/2)|1\rangle$

elevation angles that determine where the tip of this vector touches the surface of the Bloch sphere. In this picture, the North pole corresponds to the pure state $|0\rangle$ and the South pole corresponds to the (orthogonal) pure state $|1\rangle$. All other points on the surface of the Bloch sphere correspond to the superposition states of the form $a|0\rangle + b|1\rangle$ for all possible values of the complex numbers a and b such that $|a|^2 + |b|^2 = 1$.

In particular, an arbitrary pure state of a single qubit $|\psi\rangle = a|0\rangle + b|1\rangle$ such that $|a|^2 + |b|^2 = 1$ can be written in terms of these azimuth and elevation angles as:

$$|\psi\rangle = e^{i\gamma} \left(\cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \right) \quad (1.6)$$

where γ , θ , and ϕ are all real numbers. A pair of elevation and azimuth angles (θ, ϕ) in the range $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$ pick out a point on the Bloch sphere. Qubit states corresponding to different values of γ are indistinguishable and are all represented by the same point on the Bloch sphere. γ is said to be an overall phase factor that is unobservable.

Students are often confused about the Bloch sphere for three main reasons: first how come the azimuth and elevation angles are expressed in half-angles? Second, how come orthogonal states are not at right angles on the Bloch sphere? Instead they are 180° apart. Third how can it be that the γ parameter has no observable effect?

How might we draw a picture that captures in an intuitive way the complete character of a qubit in a superposition state such as $a|0\rangle + b|1\rangle$? The Bloch sphere provides a way of *visualizing* the quantum mechanical state of a single qubit. “Wait a minute!” you say. “Aren’t orthogonal states supposed to be at right angles? How can the $|0\rangle$ state be the North pole and the $|1\rangle$ be the South Pole? They’re 180° apart!”

Students are often confused by the Bloch sphere representation of a quantum state because orthogonal states are not found to be at right angles on the Bloch sphere. So it is worth a little detour to explain how the Bloch sphere is constructed.

Consider the general quantum state $a|0\rangle + b|1\rangle$. Since a and b are complex numbers they can be written in either Cartesian or Polar coordinates as: $a = x_a + iy_a = r_a e^{i\phi_a}$ and $a = x_b + iy_b = r_b e^{i\phi_b}$ with $i = \sqrt{-1}$ and the x ’s, y ’s, r ’s, and ϕ ’s are all real numbers. So, naively, it looks like we need to depict four real numbers x_a, x_b, y_a, y_b or r_a, r_b, ϕ_a, ϕ_b depending on whether we use the Cartesian or polar representation of the complex numbers a and b . Not so!

Write the general state of a qubit $a|0\rangle + b|1\rangle$ as $r_a e^{i\phi_a} |0\rangle + r_b e^{i\phi_b} |1\rangle$. Since an overall phase factor has no observable consequence (you’ll prove this as an exercise later), we can multiply by any global phase we please to obtain an equivalent state. In particular, we could multiply by the phase factor $e^{-i\phi_a}$ to obtain $r_a |0\rangle + r_b e^{i(\phi_b - \phi_a)} |1\rangle$. This allows us to represent the state of the qubit using *three* real numbers r_a, r_b and $\phi = (\phi_b - \phi_a)$. Switching back to Cartesian coordinates for the amplitude of the $|1\rangle$ component we can write this state as $r_a |0\rangle + (x + iy) |1\rangle$. Applying normalization we have $|r_a|^2 + |x + iy|^2 = 1$ or equivalently $r_a^2 + x^2 + y^2 = 1$ which is the equation of a *sphere* in coordinates r_a , x , and y . We can rename $r_a = z$ for aesthetic reasons and it doesn’t change anything but now we have the equation

of a sphere in coordinates x , y , and z . Ok so let's switch from these Cartesian coordinates to spherical coordinates. We have,

$$x = r \sin(\theta) \cos(\phi) \quad (1.7)$$

$$y = r \sin(\theta) \sin(\phi) \quad (1.8)$$

$$z = r \cos(\theta) \quad (1.9)$$

But given the constraint $x^2 + y^2 + z^2 = r^2 = 1$, we see $r = 1$. So now the position on the surface of the sphere is specified using only *two* parameters, θ and ϕ . And the general qubit state can be written as $z|0\rangle + (x + iy)|1\rangle$ or equivalently, since $r = 1$, $\cos(\theta)|0\rangle + (\sin(\theta) \cos(\phi) + i \sin(\theta) \sin(\phi))|1\rangle$, or equivalently $\cos(\theta)|0\rangle + e^{i\phi} \sin(\theta)|1\rangle$ since $\cos(\phi) + i \sin(\phi) = e^{i\phi}$. Given that a qubit must lie between the extremes of being wholly $|0\rangle$ (which occurs when $\theta = 0$ and wholly $|1\rangle$) (which occurs when $\theta = 90^\circ$ it appears all the qubit states are mapped out over just a hemispherical region of the sphere defined by $x^2 + y^2 + z^2 = 1$. If we want all the possible qubit states to correspond to the points on the surface of a whole sphere, we can simply map this hemisphere or points onto a sphere of points by introducing a new angle $\theta' = 2\theta$. Thus the general qubit state can now be written as $\cos(\frac{\theta'}{2})|0\rangle + e^{i\phi} \sin(\frac{\theta'}{2})|1\rangle$. Thus the complete set of qubit states is now mapped out as θ' runs from 0° to 180° . This final sphere is the Bloch sphere.

An immediate consequence of how the Bloch sphere is constructed is that orthogonal quantum states, i.e., states $|\psi\rangle$ and $|\chi\rangle$ for which $\langle\psi|\chi\rangle = 0$, are represented by antipodal points on the Bloch sphere (rather than being drawn at right angles which is how we usually expect to see orthogonal vectors drawn in 3D space). This is the reason why $|0\rangle$ lies at the North Pole and $|1\rangle$ lies at the South Pole of the Bloch sphere. For a general pure state, represented as a point on the surface of the Bloch sphere, the antipodal state is the one diametrically opposite it on the other side of the Bloch sphere such that a straight line drawn between the original state and its antipodal state would pass through the center of the Bloch sphere. The operation that maps an unknown state to its antipodal state cannot be expressed as a rotation on the Bloch sphere. Rather it is the sum of a rotation (in longitude through 180 degrees) and a reflection (in latitude with respect to the equatorial plane of the Bloch sphere). This inability to express the operation purely as a rotation will turn out to impact our ability to achieve it in a sequence of unitary quantum gates.

Figure 1.4 shows the Bloch sphere labeled with pure 1-qubit states at the extremes of the x -, y -, and z -axes. These are, respectively, $|\nearrow\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|\nwarrow\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, $|R\rangle = |\circlearrowleft\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$, $|L\rangle = |\circlearrowright\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$, $|0\rangle$, and $|1\rangle$. Notice that orthogonal states are indeed located at antipodal points on the surface of the Bloch sphere.

1.3.3.1 Other Rotations Having Period 4π

When first encountering the Bloch sphere, students often find it hard to grasp why a rotation of 2π radians (i.e., 360°) would not restore an object back to its original

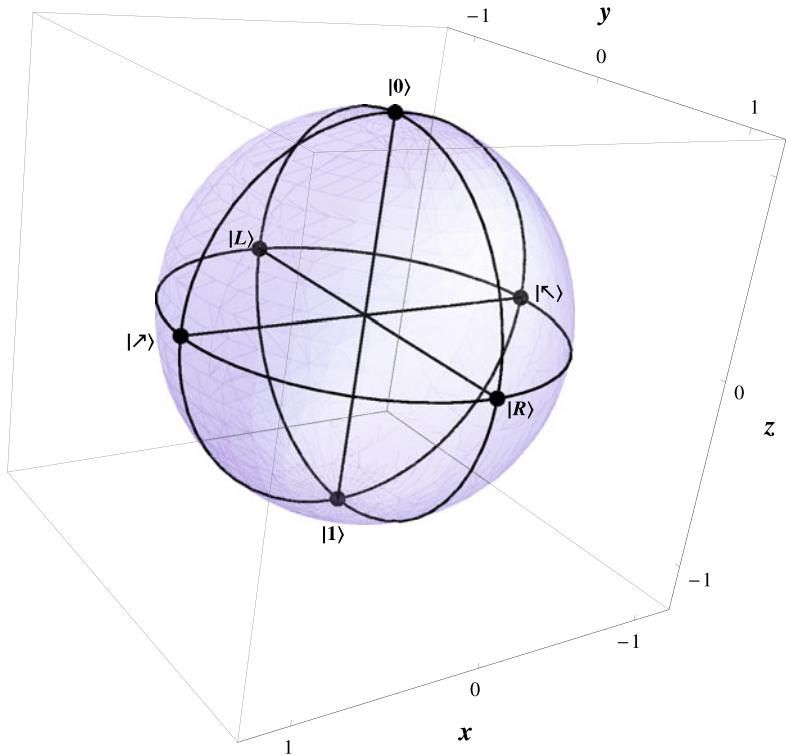


Fig. 1.4 Bloch sphere representation of the states $|0\rangle$, $|1\rangle$, $|\swarrow\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|\nwarrow\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, $|R\rangle = |\circlearrowleft\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$, and $|L\rangle = |\circlearrowright\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$. Orthogonal pure states are at antipodal points on the surface of the Bloch sphere

configuration. However, such a phenomenon can also be seen in the motions of certain classical physical systems.

For example, extend your right hand straight out so your palm is face up. *Keeping your palm face up at all times*, rotate your hand clockwise around a vertical axis passing through the center of your palm until your hand returns to its original configuration. The basic contortions you need to do are as follows: starting with your right hand extended straight out palm up, pull your arm inwards (keeping your palm flat), twisting your wrist to the right and pushing your elbow to the left, continue twisting your palm clockwise so your fingertips are pointing towards your right shoulder, and swing your elbow around to the right and upwards, and push your arm out again. Congratulations! Your palm has now been rotated through 2π radians (360°) and it is indeed still face up, *but* your hand is not in its original configuration because your elbow is now on top! To return your hand to its original configuration you need to apply another full rotation of 360° to your palm. To do so, continue turning your wrist to the right (still keeping your palm face up) so that your fingertips point towards your tight armpit, swing your elbow around and downwards in a clockwise

rotating arc, whilst twisting your wrist to the right. This will take your arm back to its starting configuration. Thus, your hand requires a total rotation of 4π radians (720°) to return it to its starting configuration. Resting a plate on your palm as you do this ensures you keep your palm face up for fear of dropping the plate. This is sometimes known as “Feynman’s Plate Trick”.

A more surprising demonstration of the same symmetry property occurs in the rotations of a flat belt that is fixed at one end and rotated at the other. This version is called “Dirac’s Belt Trick” and it is always a hit at parties. Take off a long flat belt strap. Have a friend hold the buckle end of the belt and hold the other end yourself. Pull the belt taut so it is flat with the outer face of the belt (as it is normally worn) pointing upwards. Tell your friend to keep hold their end of the belt tightly in a fixed position. Ok now twist (i.e., rotate) your end of the belt through 2π radians (i.e., 360°). Can you remove the kink you have imparted to the belt by passing the belt under and over itself *while keeping the orientation of the ends of the belt fixed* (i.e., flat with the outer face of the belt pointing upwards)? After a little experimentation you will conclude you cannot.

Let us make the problem even harder by applying an *additional* twist to your end of the belt through another 2π radians (i.e., another 360°). Can you remove the *double* kink by passing the belt under and over itself while keeping both ends flat and pointed upwards? Surely if you could not remove *one* kink in this manner, you would expect it would be even harder to remove two! Yet, remarkably, you can! After a rotation of 4π radians (720°) applied to the end of the belt, the belt can be restored to its original configuration by passing it under and over itself while keeping the orientations of the two ends fixed in space! This seems to be more surprising to most people than the plate trick. Yet both are examples of physical systems in which rotations of 2π radians do not restore an object to its original state whereas rotations of 4π radians do! Such examples show that the 4π periodicity of the Bloch sphere has parallels in the classical world around us.

1.3.4 Reading the Bit Value of a Qubit

In the everyday classical world when we read, or measure, or observe, something we don’t usually perturb it in the process. For example, when we read a newspaper we don’t change the words on the page merely by reading them. Moreover, if ten people read ten different copies of the same edition of the same paper they would all see the same words. However, in the quantum world this is not what happens.

The states $|0\rangle$ and $|1\rangle$ correspond to the North and South poles of the Bloch sphere respectively, and the axis passing through these points is the z -axis (see Fig. 1.5). Thus the act of reading the bit value of a qubit amounts to determining the alignment of its spin with respect to this z -axis. If the particle is aligned “spin-up” it is in the state $|0\rangle$. If it is aligned “spin-down” it is in the state $|1\rangle$.

When a single qubit in state $a|0\rangle + b|1\rangle$ is read (or “measured” or “observed”), with respect to some axis through the center of the Bloch sphere, the probability of

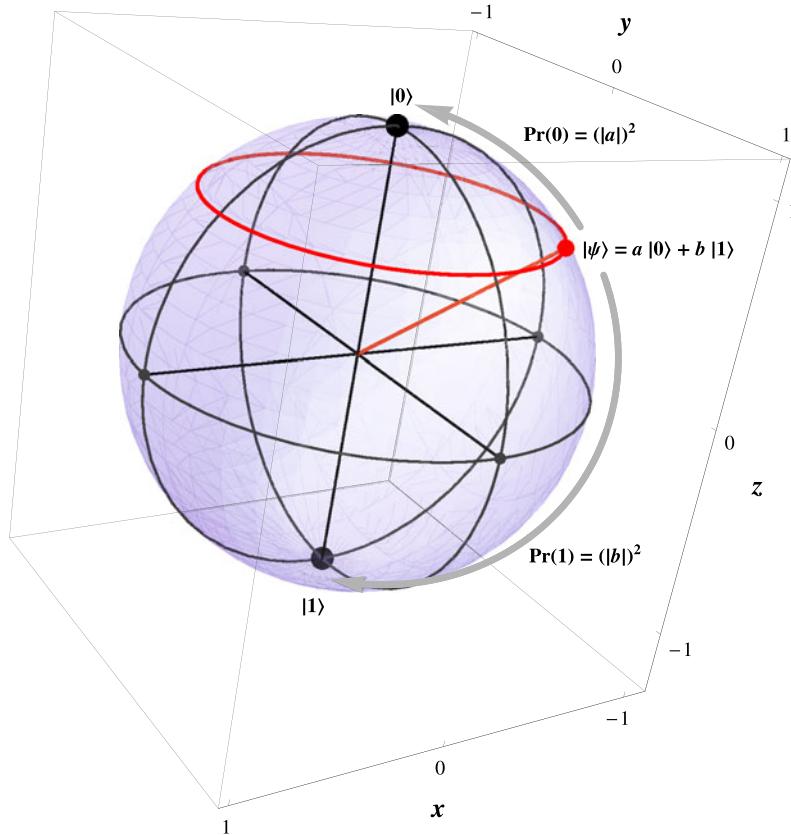


Fig. 1.5 Measuring the bit value of a qubit initially in state $a|0\rangle + b|1\rangle$ yields the answer 0 with probability $|a|^2$ or 1 with probability $|b|^2$, and projects the qubit into either state $|0\rangle$ or state $|1\rangle$ respectively

finding it in state $|0\rangle$ or state $|1\rangle$ depends upon the values of a and b , and on the orientation of this axis. The most commonly used axis is that passing through the North and South poles corresponding to the states $|0\rangle$ and $|1\rangle$. A measurement of a qubit with respect to this axis is called a measurement “in the computational basis” because the answer we get will be one of the bit values $|0\rangle$ or $|1\rangle$. The outcome we obtain is, in general, not certain but depends on the amplitudes a and b . Specifically, measuring the bit value of $a|0\rangle + b|1\rangle$ in the computational basis will yield the answer $|0\rangle$ with probability $|a|^2$ and the answer $|1\rangle$ with probability $|b|^2$. These two probabilities sum to 1, i.e., $|a|^2 + |b|^2 = 1$.

$$\text{Read}(a|0\rangle + b|1\rangle) = \begin{cases} 0 & \text{with probability } |a|^2 \\ 1 & \text{with probability } |b|^2 \end{cases} \quad (1.10)$$

Thus, a single qubit quantum memory register exhibits the interesting property that even though its contents may be definite, i.e., it may be precisely in the state

$|\psi\rangle = a|0\rangle + b|1\rangle$, the outcome we obtain from reading it is non-deterministic. Sometimes we will find it in state $|0\rangle$ and sometimes we will find it in state $|1\rangle$. However, the instant *after* the measurement is made, the state is known with certainty to be $|0\rangle$ or $|1\rangle$ consistent with result we obtained. Moreover, if we rapidly and repeatedly keep measuring the same state we can suppress its evolution and effectively freeze it in a fixed quantum state $|\psi\rangle \xrightarrow{\text{read}} |0\rangle \xrightarrow{\text{read}} |0\rangle \xrightarrow{\text{read}} |0\rangle \dots$ or $|\psi\rangle \xrightarrow{\text{read}} |1\rangle \xrightarrow{\text{read}} |1\rangle \xrightarrow{\text{read}} |1\rangle \dots$. This is a variant of the so-called Quantum Zeno Effect.³ But if we allow time to elapse between measurements the state will, in general, evolve, or “drift off”, in accordance with Schrödinger’s equation.

1.4 Multi-qubit Quantum Memory Registers

So far we have only been dealing with single qubits, but a useful quantum computational device will need to have a multi-qubit quantum memory register. In general, this is assumed to consist of a collection of n -qubits, which are assumed to be ordered, indexed and addressable so that selective operations can be applied to any single qubit or any pair of qubits at will. If two qubits selected for an operation are not physically adjacent, there is usually an operational sequence that achieves the interaction between them as if they were. This detail is typically omitted from the abstract model of the quantum memory as it is more an implementation issue than anything fundamental to the computational model.

Just as a single qubit can be found in a superposition of the possible bit values it may assume, i.e., $|0\rangle$ and $|1\rangle$, so too can a n -qubit register be found in a superposition of all the 2^n possible bit strings $|00\dots 0\rangle, |00\dots 1\rangle, \dots, |11\dots 1\rangle$ it may assume. However, the most interesting superposition states typically involve non-uniform contributions of eigenstates.

1.4.1 The Computational Basis

When we describe the state of a multi-qubit quantum memory register as a superposition of its possible bit-string configurations, we say the state is represented in the *computational basis*. This is arguably the most natural basis for quantum computing. For example, the most general form for a pure state of a 2-qubit quantum memory register can be written as:

$$|\psi\rangle = c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle \quad (1.11)$$

³The Quantum Zeno Effect says that if you repeatedly measure (or observe) a quantum system, you can suppress its quantum mechanical evolution. It is named after Zeno of Elea who devised a paradox that aimed to prove if you continually observe an arrow in flight at any instant it would appear motionless and hence it cannot be moving: “If everything when it occupies an equal space is at rest, and if that which is in locomotion is always occupying such a space at any moment, the flying arrow is therefore motionless.”—Aristotle, Physics VI:9, 239b5.

where $|c_0|^2 + |c_1|^2 + |c_2|^2 + |c_3|^2 = 1$. This implies we can think of the register as containing many different bit string configurations at once, each with their own amplitude. Similarly, the general state of a 3-qubit register can be written as:

$$\begin{aligned} |\psi\rangle = & c_0|000\rangle + c_1|001\rangle + c_2|010\rangle + c_3|011\rangle + c_4|100\rangle + c_5|101\rangle \\ & + c_6|110\rangle + c_7|111\rangle \end{aligned} \quad (1.12)$$

where $|c_0|^2 + |c_1|^2 + |c_2|^2 + |c_3|^2 + |c_4|^2 + |c_5|^2 + |c_6|^2 + |c_7|^2 = 1$. Continuing in this fashion, we see that the most general form for a pure state of an n -qubit quantum memory register is:

$$|\psi\rangle = c_0|00\dots0\rangle + c_1|00\dots1\rangle + \dots + c_{2^n-1}|11\dots1\rangle = \sum_{i=0}^{2^n-1} c_i|i\rangle$$

where $\sum_{i=0}^{2^n-1} |c_i|^2 = 1$ and $|i\rangle$ represents the “computational basis eigenstate” whose bit values match those of the decimal number i expressed in base-2 notation, padded on the left (if necessary) with “0” bits in order to make a full complement of n bits. For example, the 5-qubit computational basis eigenstate corresponding to $|6\rangle$ is $|00110\rangle$. This is because 6 in base-2 is “110” and then we pad on the left with two “0” bits to make a total of 5 bits.

As for the case of single qubits, such ket vectors can always be regarded as a short hand notation for a column vector. The size of these column vectors grow exponentially with the number of qubits, making it computationally intractable to simulate arbitrary quantum computations on classical computers. For example, a 100-qubit quantum memory register requires 2^{100} complex amplitudes to specify it completely! In very few qubits, we run out of particle in the known Universe with which to make a classical memory large enough to represent a quantum state.

In a multi-qubit quantum state it is not necessary (and for often not desirable) for every amplitude to be non-zero. For example, if the quantum memory register contains the output from some quantum computation, typically, many of the eigenstates (corresponding) to non-solutions will be absent. For example, a particular 3-qubit quantum state, $|\psi\rangle = a|001\rangle + b|010\rangle + c|100\rangle$ does not contain any contributions from the eigenstates $|000\rangle, |011\rangle, |101\rangle, |110\rangle, |111\rangle$. The amplitude of these omitted components is zero by implication. Hence, as a column vector, the aforementioned 3-qubit state would actually be:

$$|\psi\rangle = a|001\rangle + b|010\rangle + c|100\rangle \equiv \begin{pmatrix} 0 \\ a \\ b \\ 0 \\ c \\ 0 \\ 0 \\ 0 \end{pmatrix} \equiv \begin{pmatrix} \text{amplitude of } |000\rangle \text{ component} \\ " |001\rangle " \\ " |010\rangle " \\ " |011\rangle " \\ " |100\rangle " \\ " |101\rangle " \\ " |110\rangle " \\ " |111\rangle " \end{pmatrix} \quad (1.13)$$

1.4.2 Direct Product for Forming Multi-qubit States

Suppose we create a quantum memory register from a set of n independent qubits. How the state of the n -qubit register is related to the states of the individual qubits? The answer is provided by way of the *direct product* of the n individual quantum states.

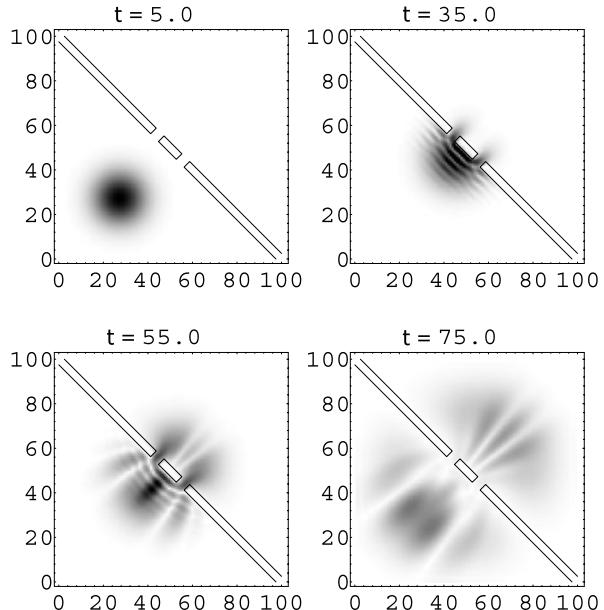
Definition *Direct Product of Quantum States* of qubit states. Let $|\phi\rangle = \sum_{j=0}^{2^m-1} a_j |j\rangle$ be an m -qubit pure state, and $|\psi\rangle = \sum_{k=0}^{2^n-1} b_k |k\rangle$ be an n -qubit pure state. The quantum state of a memory register formed by considering $|\phi\rangle$ and $|\psi\rangle$ together is computed by taking their direct product, $|\phi\rangle \otimes |\psi\rangle$ (sometimes called “tensor” or “Kronecker” product too):

$$\begin{aligned}
 |\phi\rangle \otimes |\psi\rangle &= \sum_{j=0}^{2^m-1} a_j |j\rangle \otimes \sum_{k=0}^{2^n-1} b_k |k\rangle = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{2^m-1} \end{pmatrix} \otimes \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{2^n-1} \end{pmatrix} \\
 &= \begin{pmatrix} a_0 \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{2^n-1} \end{pmatrix} \\ a_1 \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{2^n-1} \end{pmatrix} \\ \vdots \\ a_{2^m-1} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{2^n-1} \end{pmatrix} \end{pmatrix} = \begin{pmatrix} a_0 b_0 \\ a_0 b_1 \\ \vdots \\ a_0 b_{2^n-1} \\ a_1 b_0 \\ a_1 b_1 \\ \vdots \\ a_1 b_{2^n-1} \\ \vdots \\ a_{2^m-1} b_0 \\ a_{2^m-1} b_1 \\ \vdots \\ a_{2^m-1} b_{2^n-1} \end{pmatrix} \quad (1.14)
 \end{aligned}$$

For example, let $|\phi\rangle = a|0\rangle + b|1\rangle$ and $|\psi\rangle = c|0\rangle + d|1\rangle$. Then the direct product

$$\begin{aligned}
 |\phi\rangle \otimes |\psi\rangle &= \begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} a \begin{pmatrix} c \\ d \end{pmatrix} \\ b \begin{pmatrix} c \\ d \end{pmatrix} \end{pmatrix} \\
 &= \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix} = ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle \quad (1.15)
 \end{aligned}$$

Fig. 1.6 A particle impinging on a double slit seen at four different times $t = 5.0, 35.0, 55.0$ and 75.0 . Notice the interference pattern beyond the double slit (upper right quadrant of lower right frame). This, and several other stunning animations of quantum mechanical interference effects, can be found in [420]



1.4.3 Interference Effects

One of the most striking differences between quantum memory registers and classical memory registers is the possibility of encountering “quantum interference” effects in the quantum case that are absent in the classical case. In general terms, quantum interference can occur whenever there is more than one way to obtain a particular computational result. The different pathways can interfere constructively to increase the net probability of that result, or they can interfere destructively to reduce the probability of that result. For example, if a quantum mechanical particle impinges on a double slit it will, as shown in Fig 1.6, pass through both slits and self-interfere beyond the slit, resulting in an oscillatory pattern of probability amplitude for where the particle will be found. To understand this quantitatively, let’s consider the probability of obtaining a particular computational result first by pretending that our quantum register behaves like a classical probabilistic register and then by treating it (correctly) as a true quantum memory register.

Let $|j\rangle$ and $|k\rangle$ be two eigenstates of an n -qubit quantum memory register that hold two different bit strings corresponding to integers j and k respectively. These states are orthogonal ($\langle j|k \rangle = 0$) and normalized ($\langle j|j \rangle = \langle k|k \rangle = 1$). So long as it is not being observed, it is possible for the quantum memory register to exist in a superposition of any of its allowed eigenstates such as a superposition of $|j\rangle$ and $|k\rangle$, i.e., $|\psi\rangle = c_j|j\rangle + c_k|k\rangle$. If we observed this state in the computational basis we would find it in state $|j\rangle$ with probability $|c_j|^2$ and in state $|k\rangle$ with probability $|c_k|^2 = 1 - |c_j|^2$ (since these are the only two possibilities).

Thus, on the face of it, one might think that the quantum memory register holding the state $|\psi\rangle = c_j|j\rangle + c_k|k\rangle$ behaves just the same as if it were a classical proba-

bilistic memory register that outputs state $|j\rangle$ with probability p_j ($= |c_j|^2$) and state $|k\rangle$ with probability p_k ($= |c_k|^2$). But as we now show, this is not the case.

Specifically, let A be some observable that can act on an n -qubit register. Suppose one of the eigenvalues of this observable is “ a ” when the corresponding state of the memory register is $|\psi_a\rangle$. In other words we have $A|\psi_a\rangle = a|\psi_a\rangle$.

The question is, with what probability would be obtain the value “ a ” when we measure the observable A when the quantum memory register is in state $|\psi\rangle = c_j|j\rangle + c_k|k\rangle$?

Well in the (erroneous) “classical” view, the register *really* holds either state $|j\rangle$ or state $|k\rangle$ but we are ignorant about which is the case. The probability of getting “ a ” if the register is in state $|j\rangle$ is $P_j(a) = |\langle\psi_a|j\rangle|^2$. Similarly, the probability of getting “ a ” if the register is in state $|k\rangle$ is $P_k(a) = |\langle\psi_a|k\rangle|^2$. As we are ignorant about whether the register really holds state $|j\rangle$ or state $|k\rangle$ the probability with which we expect to see “ a ” is:

$$\begin{aligned} P^{\text{CLASSICAL}}(a) &= P_j(a)p_j + P_k(a)p_k = |c_j|^2 P_j(a) + |c_k|^2 P_k(a) \\ &= |c_j|^2 |\langle\psi_a|j\rangle|^2 + |c_k|^2 |\langle\psi_a|k\rangle|^2 \end{aligned} \quad (1.16)$$

So this is our prediction for the probability with which we see result “ a ” if our memory register behaves “classically”.

In the case of the “quantum” interpretation of the register, however, we’re not ignorant of anything! The register *truly* exists in the superposition state $|\psi\rangle = c_j|j\rangle + c_k|k\rangle$, and the probability of getting “ a ” is therefore:

$$\begin{aligned} P^{\text{QUANTUM}}(a) &= |\langle\psi_a|\psi\rangle|^2 = |c_j\langle\psi_a|j\rangle + c_k\langle\psi_a|k\rangle|^2 \\ &= |c_j|^2 |\langle\psi_a|j\rangle|^2 + |c_k|^2 |\langle\psi_a|k\rangle|^2 + 2 \operatorname{Re}(c_j c_k^* \langle\psi_a|j\rangle \langle\psi_a|k\rangle^*) \end{aligned} \quad (1.17)$$

Thus, in the quantum case there is an addition term contributing to the probability of obtaining result “ a ”. This is the result of quantum interference between the different computational pathways by which result “ a ” can be obtained.

1.4.4 Entanglement

“I would not call [entanglement] one but rather the characteristic trait of quantum mechanics, the one that enforces its entire departure from classical lines of thought.”

– Erwin Schrödinger

Another way in which quantum memory registers can differ from classical memory registers is in their ability to exist in *entangled* states. This is a state of a composite quantum system that involves unusually strong correlations between parts of the system. There is considerable debate at present about the nature of entanglement, especially in systems involving more than two particles, and whether entanglement is

strictly necessary to obtain a complexity advantage over a classical computer. However, at this time it appears that entanglement is *crucial* to obtaining the exponential speedups seen in some quantum algorithms.

So what is an entangled state exactly? In its simplest terms we can define an entangled state as follows:

Definition: Entangled Pure State A multi-qubit pure state is *entangled* if and only if it cannot be factored into the direct product of a definite state for each qubit individually. Thus, a pair of qubits, A and B , are entangled if and only if their joint state $|\psi\rangle_{AB}$ cannot be written as the product of a state for qubit A and a state for qubit B , i.e., if and only if $|\psi\rangle_{AB} \neq |\psi\rangle_A \otimes |\psi\rangle_B$ for any choice of states $|\psi\rangle_A$ and $|\psi\rangle_B$.

In a multi-qubit memory register if qubits are entangled then actions performed on one subset of qubits can have an impact on another, “untouched”, subset of qubits. For example, consider a 2-qubit memory register comprised of qubits A and B , in state $\frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B)$. If qubit A is measured in the computational basis and found to be in state $|1\rangle$ then even though qubit B has not yet been touched, its quantum state is now determined to be $|1\rangle$ too. Thus a measurement of qubit A has had a side effect on the value of qubit B !

For notational compactness entangled state are more commonly written by dropping the particle label (A , B , etc.) because this is implied by position, and by dropping the \otimes product as this is implied by simply abutting ket vectors. So the aforementioned entangled state could also be written as $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

Entanglement is a pervasive phenomenon in multi-qubit quantum memory registers. It is also the cornerstone of many quantum algorithms. For example, we can prepare two entangled quantum registers, A and B say, such that register A contains a set of indices running from 0 to $2^n - 1$ and register B contains a set of values of a function who behavior depends upon the value of the index in register A . So the joint state (ignoring the normalization factor) can be something like $\sum_{i=0}^{2^n-1} |i\rangle_A |f(i)\rangle_B$. By measuring the value of the function (in register B) to be value “ c ” say, we can project out the set of indices (in register A) consistent with the observed function value, giving us a superposition of the form $\sum_{\{i': f(i')=c\}} |i'\rangle_A |c\rangle_B$. That’s a neat trick because in one shot we get all the index values (in register A) that give the same value for the function (in register B).

1.4.4.1 Entanglement and Quantum States in Different Number Bases

One of the most interesting aspects of entanglement is how it is tied to our choice of representation of numbers. Traditionally, we think of quantum computing using the base-2 number system. Showing the number base as a subscript we have $|0_{10}\rangle = |0_2\rangle$, $|1_{10}\rangle = |1_2\rangle$, $|2_{10}\rangle = |10_2\rangle$, $|3_{10}\rangle = |11_2\rangle$,

If the quantum gate, represented by the unitary matrix U , is to act on n qubits, U will have dimensions of that are a power of two, specifically, $2^n \times 2^n$. Likewise,

the unitary matrix corresponding to a quantum gate that acts on qutrits (i.e., base 3 quantum computation), will have dimensions that are a power of three, i.e., $3^n \times 3^n$. Typically, most researchers use a base 2 (qubit) model of quantum computation. This is partly out of habit, and partly because quantum gates that manipulate qubits (and which therefore require 2-body interactions) are assumed to be simpler to build than those that manipulate qutrits (and which therefore require 3-body interactions). But in principle, one could use whatever base one wants.

Does the choice of base matter? Well, not from a computability perspective. Any computation that can be done using qubits can also be done using qutrits. However, it does raise some interesting issues when we consider the degree to which entanglement is critical to quantum computation. For example, suppose you wanted to create a superposition of two numbers “1” and “2” in some quantum memory register. Using qubits, such a superposition could be coded as $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ (which is entangled). However, using qutrits, the equivalent state could be encoded as $\frac{1}{\sqrt{2}}(|1\rangle + |2\rangle)$ (a plain, unentangled, superposition). So the choice of base affects the degree to which entanglement is needed.

Some researchers misinterpreted the implications of this by proposing that quantum computation can be implemented without entanglement. For example, suppose we consider using a single atom (or perhaps artificial quantum dot) that has a huge spectrum of energy levels available to it. We could imagine associating each energy level with a different computational state: the ground state of the atom could be “ $|0\rangle$ ”, the first excited state “ $|1\rangle$ ”, the second excited state “ $|2\rangle$ ” etc. We could then regard a quantum computation as a sequence of operations that maps some initial state of this atom (represented as an unentangled superposition of states) into a final state (represented as an unentangled superposition of states). And it would seem as though entanglement is unnecessary.

The problem with this approach is that it neglects a hidden exponential cost. To do universal (i.e., arbitrary) quantum computation we need to be able to access exponentially many different energy levels. However, as the total energy of the atom is finite, this means we will need to “fit” exponentially many energy levels into a finite energy interval. Hence, we will require exponentially increasing precision in order to address a specific energy level. Hence, although in principle one could perform quantum computation in higher bases, and perhaps lower the degree to which entanglement is needed, in practice it is very hard to imagine doing away with entanglement entirely.

1.5 Evolving a Quantum Memory Register: Schrödinger's Equation

So far we have been discussing the properties of individual quantum bits (such as superposition), and those of multi-qubit quantum memory registers (such as superposition, entanglement and interference). Our working assumption has been that the instantaneous state of a quantum memory register, $|\psi(t)\rangle$, holds the instantaneous

state of the quantum computation. But how does this state evolve with time, and how can we control this evolution to enact a purposeful quantum computation?

1.5.1 Schrödinger's Equation

Remarkably in 1929, long before anyone had ever thought of quantum computers, physicist Erwin Schrödinger discovered an equation that describes how *any* isolated quantum system evolves in time. Since a quantum memory register is nothing more than an isolated quantum system, it too must be described by Schrödinger's equation.

Schrödinger's equation is a linear first order deterministic partial differential equation that involves the instantaneous state of the quantum memory register $|\psi(t)\rangle$, a time independent Hermitian matrix \mathcal{H} , called the Hamiltonian (the observable for the total energy of the system), and a constant \hbar equal to Planck's constant divided by 2π . The fact that Schrödinger's equation is "linear" means that sums of solution to the equation are also solutions to the equation, which is the fundamental origin of the superposition principle. The fact that the Schrödinger equation is deterministic means that if you know its instantaneous state at any moment you can predict its future and past states with certainty (provided the system is not observed).

Regardless of the precise details of the physical system, Schrödinger's equation always takes the form:

$$i\hbar \frac{\partial |\psi(t)\rangle}{\partial t} = \mathcal{H}|\psi(t)\rangle \quad (1.18)$$

As \hbar is a constant, and $|\psi(t)\rangle$ describes the instantaneous state of the quantum memory register, the form of this equation implies that all of the details pertaining to the particular physical system in question must be bundled into the operator \mathcal{H} —the Hamiltonian. So what does this Hamiltonian mean exactly?

1.5.2 Hamiltonians

In quantum mechanics observables are described by operators, which in turn can be represented as Hermitian matrices. The allowed values for an observable are the eigenvalues of its associated Hermitian matrix. The Hamiltonian, \mathcal{H} is the observable corresponding to the total energy of the system, and its eigenvalues are the possible values one can obtain when one measures (or "observes") the total energy of the system. Depending on the physical situation such a Hamiltonian may be time dependent or time independent.

The Hamiltonian \mathcal{H} for a particular quantum physical system is built up from knowledge of the elementary interactions available in the system, and it can be writ-

ten in terms of operator products like those we encountered in Sect. 1.3.2. For example, the Hydra superconducting quantum processor [423] has the Hamiltonian:

$$\mathcal{H}(t) = \sum_{i=1}^N h_i Z_i + \sum_{i < j=2}^N J_{ij} Z_i Z_j + \sum_{i=1}^N \Delta_i(t) X_i \quad (1.19)$$

where $Z_i = \sigma_z^i$ and $X_i = \sigma_x^i$ are the Pauli-Z and Pauli-X matrices for qubit i , h_i is the bias applied to qubit i , $\Delta_i(t)$ is the tunneling matrix element for qubit i , and J_{ij} is the coupling between qubits i and j .

The fact that \mathcal{H} is the observable for the total energy of the n -qubit system means that \mathcal{H} is a $2^n \times 2^n$ dimensional Hermitian matrix such that there exist energy eigenstates $|\psi_i\rangle$, and energy eigenvalues E_i such that $\mathcal{H}|\psi_i\rangle = E_i|\psi_i\rangle$. The eigenvalues E_i are the only allowed values for the total energy of the system. Thus there is always some basis (the energy eigenbasis $\{|\psi_i\rangle\}$) in which \mathcal{H} is a diagonal matrix, $\mathcal{H} = \sum_i E_i |\psi_i\rangle \langle \psi_i|$.

$$\mathcal{H} = \begin{pmatrix} E_0 & 0 & 0 & 0 \\ 0 & E_1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & E_{2^n-1} \end{pmatrix} \quad (1.20)$$

However, the Hamiltonian is often stated with respect to some other basis, e.g., the computational basis, $\{|00\dots0\rangle, |00\dots1\rangle, \dots, |1\dots1\rangle\}$. Hence, it is sometimes necessary to change the basis used to describe states and operators in quantum computing. We will come back to this issue and discuss it in detail in Sect. 1.6.4.

1.5.3 Solution as a Unitary Evolution of the Initial State

Once the Hamiltonian is known the Schrödinger equation can be solved. The simplest case is that of a time-independent Hamiltonian. In this case the solution to the Schrödinger equation is:

$$U(t) = \exp(-i\mathcal{H}t/\hbar) \quad (1.21)$$

This says that if you know the initial state of the system, $|\psi(0)\rangle$, you can determine its state at a later time, t , by acting on the initial state with the operator $\exp(-i\mathcal{H}t/\hbar)$. Or, in other words, the system is described by some Hamiltonian \mathcal{H} and you let it “run” for a length of time t , then the result you get is $|\psi(t)\rangle = U(t)|\psi(0)\rangle = \exp(-i\mathcal{H}t/\hbar)|\psi(0)\rangle$.

The matrix $U(t)$ is therefore the matrix exponential of $-i\mathcal{H}t/\hbar$. If A is any matrix, its matrix exponential is:

$$e^A = \mathbb{1} + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \frac{A^4}{4!} + \frac{A^5}{5!} + \dots \quad (1.22)$$

As \mathcal{H} is an Hermitian matrix, its matrix exponential $\exp(-i\mathcal{H}t/\hbar)$ is a *unitary* matrix. A unitary matrix has the property that its inverse is equal to its conjugate transpose, i.e. $U^{-1} = U^\dagger$. Therefore, a unitary matrix is always invertible which means that the evolution it describes is reversible, i.e., there is no loss of information. Hence, the closest classical analog to quantum computing is classical reversible computing, as it too preserves information about the computational history.

1.5.4 Computational Interpretation

A classical computer follows essentially a LOAD-RUN-READ cycle wherein one *loads* data into the machine, *runs* a program using this data as input, and then *reads* out the result. This becomes an analogous PREPARE-EVOLVE-MEASURE cycle for a quantum computer. That is, one *prepares* a quantum state, *evolves* it on the quantum computer, and *measures* the result.

Each aspect of the quantum computer's operation offers new opportunities unavailable in the analogous phase of a classical computer's operation. For example, whereas in a classical computer you can only *load* one input at a time, in a quantum computer you can *prepare* exponentially many inputs in the same amount of time. The whereas a classical computer can only run a computation on one input, a quantum computer can evolve a superposition of computations on all inputs in the same time. Finally, whereas a classical computer can only read one output, we can perform more sophisticated measurements of the output state from a quantum computer to compute certain *joint properties* of all the answers to a particular computational problem in the time it takes a classical computer to find just one of the answers. This gives quantum computers the potential to be much faster than any classical computer, even a state-of-the supercomputer.

1.6 Extracting Answers from Quantum Computers

The process of extracting answers from quantum computers can be more tricky than one might imagine. In order to learn the result of a quantum computation we must *read* the quantum memory register that contains it. Such an act is more properly thought of as performing a *measurement* on a certain quantum state (i.e., the result of the quantum computation) in a certain basis (typically, but not necessarily, the *computational basis*).

1.6.1 Observables in Quantum Mechanics

A measurement of a quantum memory register couples the quantum computer to the measuring device, temporarily, causing information from the quantum memory

register to be transferred to the measuring apparatus, whereupon it is converted to classical information and amplified to a scale detectable by human senses. At this point we say the observable has been “read” or “measured”. Therefore, the act of reading a quantum memory register is more properly thought of as an experimental determination of the value of some *observable* of the system.

In quantum mechanics, an *observable* for some property of an n -qubit system is represented by a $2^n \times 2^n$ dimensional Hermitian matrix, \mathcal{O} say. The Hermitian property means that $\mathcal{O} = \mathcal{O}^\dagger$ and so the eigenvalues of \mathcal{O} are guaranteed to be real. The significance of this is that quantum mechanics says that when the property associated with observable \mathcal{O} is measured that the answer we obtain has to be one of the *eigenvalues* of \mathcal{O} , and the state immediately after the measurement is the eigenvector that pairs with this eigenvalue. Thus, if $\{|\psi_i\rangle\}$ are the family of eigenvectors of \mathcal{O} and $\{\lambda_i\}$ are the corresponding family of eigenvalues, such that:

$$\mathcal{O}|\psi_i\rangle = \lambda_i |\psi_i\rangle \quad (1.23)$$

then the only possible values we can ever obtain for the property associated with observable \mathcal{O} are one of the λ_i ’s and, having obtained such a result, the state immediately after this measurement will be $|\psi_i\rangle$. Moreover, if we repeatedly prepared and measured several preparations of the state $|\psi\rangle$ then the *average* value we would obtain would be:

$$\langle \mathcal{O} \rangle = \langle \psi | \mathcal{O} | \psi \rangle \quad (1.24)$$

where $|\psi\rangle$ and \mathcal{O} should be described with respect to the same basis.

Many students find this measurement formalism perplexing. Why should acts of measurement be associated with matrices? And why should the values obtained from acts of measurements be associated with eigenvalues? What motivates this formalism?

The answer lies in our desire to have a mathematical way of describing acts of measurement that reflects, faithfully, the phenomena experimentalists encounter when they perform real measurements on quantum systems. As we shall see in the next section, by associating observables with Hermitian matrices, and the allowed values of observables with eigenvalues of those operators, we can conjure up a relatively simple and concise mathematical model of the measurement process that naturally has all the requisite properties.

1.6.1.1 Observables as Hermitian Operators

Let us start by summarizing the phenomena scientists encounter when they try to make observations on quantum systems, as this will make the subsequent mathematical account of observation that is used in quantum mechanics far more intuitive.

The first idea is that when we measure some property of a system we obtain a real number for the answer. So *measurement results need to be real numbers*.

Secondly, for quantum-scale objects, the act of observing the system can change its state. For example, to find the position of an electron you need to bounce light

off it. The shorter the wavelength of the light used the more precisely you can determine position. But the shorter the wavelength of the light the greater the momentum kick the light imparts to the electron as it scatters off it. Hence, a very precise measurement of position necessarily induces a large uncertainty in momentum and vice versa. So the second idea is that *acts of observation can change the state*.

Third, the measured values one obtains do not usually span a continuous range of possibilities but instead may take on only certain discrete values. For example, if we measure the spin of an electron it is always found to be aligned or anti-aligned with the measurement axis. Even if you try to “cheat” by setting up an experiment with the electron spinning at some known angle to the axis of measurement, when you make the measurement the spin jumps into alignment or anti-alignment with the measurement axis. These are the only two values allowed. So the third idea is that *measured values are discrete rather than continuous*.

Fourth, the order in which we make a sequence of observations can affect the outcome we obtain. So an experiment that measures property *A* first and then property *B* does not always yield the same results, even statistically, as if we measured property *B* first and then property *A*. So the fourth idea is that *the order in which we perform measurements can affect the outcome we obtain*.

Fifth, when we measure certain pairs of observables, the more accurately we can pin down one, the less accurately we can pin down the other. That is there is a fundamental quantifiable limit to how accurately we can measure certain pairs of observables. In particular, defining:

$$\begin{aligned}\Delta \mathcal{O}_A &= \mathcal{O}_A - \langle \mathcal{O}_A \rangle \\ \Delta \mathcal{O}_B &= \mathcal{O}_B - \langle \mathcal{O}_B \rangle\end{aligned}\tag{1.25}$$

it can be shown that

$$\Delta \mathcal{O}_A \Delta \mathcal{O}_B \geq \text{constant}\tag{1.26}$$

where the inequality is strict if the order in which observations are made makes a difference. The mathematical machinery used in quantum mechanics to describe acts of observation has to reflect the phenomena scientists encounter when they do actual measurements.

It turns out that all these properties fall out naturally if we associate observables with Hermitian operators. If an observable *A* is associated with an Hermitian operator \mathcal{O}_A , then:

1. Quantized values: the only allowed outcomes for the measurement are the eigenvalues of \mathcal{O}_A .
2. Real values: as \mathcal{O}_A is Hermitian its eigenvalues must be real.
3. Observation changes the state: if the system is in a superposition state just prior to a measurement then upon obtaining the result λ_i the system will be projected into the state $|\psi_i\rangle$. This is the eigenstate of \mathcal{O}_A such that $\mathcal{O}_A |\psi_i\rangle = \lambda_i |\psi_i\rangle$.
4. Non-commuting Measurements: if we are interested in two observables *A* and *B* represented by Hermitian matrices \mathcal{O}_A and \mathcal{O}_B then the order in which measurements are made will make a difference whenever $\mathcal{O}_A \cdot \mathcal{O}_B \neq \mathcal{O}_B \cdot \mathcal{O}_A$.

5. Uncertainty principle: as we show in Chap. 12, for any pair of observables \mathcal{O}_A and \mathcal{O}_B there is a minimum uncertainty with which the \mathcal{A} and \mathcal{B} properties can be measured simultaneously given by $\Delta\mathcal{O}_A\Delta\mathcal{O}_B \geq \frac{1}{4}|\langle[\mathcal{O}_A, \mathcal{O}_B]\rangle|$.

Hence, although the quantum mechanical account of observables appears quite alien to most people when they first encounter it, remember that the reason it is set up this way is simply to capture the empirically determined properties of measurements and observations on quantum scale objects.

1.6.2 Observing in the Computational Basis

The most common kind of measurement that is made in quantum computing is to measure a set of qubits “in the computational basis”. By this we mean that the spin orientation of each qubit in the quantum memory register is measured along an axis parallel to the z -axis of the Bloch sphere, which is the axis passing through its North and South poles. When such a measurement is made, each qubit will be found to be aligned or anti-aligned with the z -axis corresponding to being “spin-up” (i.e., in state $|0\rangle$) or “spin-down” (i.e., in state $|1\rangle$) respectively. When such a measurement is applied to each qubit in an n -qubit quantum memory register one will obtain one of the 2^n possible bit string configurations that the register can take on. The probability of obtaining the different outcomes depends upon the amplitude with which each bit string configuration appears in the superposition state of the register just prior to it being measured.

Consider, for example, an n -qubit quantum memory register in the (normalized) state $\sum_{i=0}^{2^n-1} c_i |i\rangle$. Here we use the shorthand notation that $|i\rangle$ really stands for a bit string, $|i\rangle \equiv |i_{n-1}i_{n-2}\dots i_2i_1i_0\rangle$, such that $i = 2^0i_0 + 2^1i_1 + \dots + 2^{n-1}i_{n-1}$. The outcome we obtain will depend on the amplitudes c_i and on whether we measure some or all of the qubits.

1.6.2.1 Complete Readout

If *all* the qubits are measured in the computational basis one will obtain the result $|i\rangle$ with probability $|c_i|^2$. Consequently, if one of the amplitudes is zero, i.e., there exists an index value i' such that $c_{i'} = 0$, there is no chance whatsoever of obtaining the answer $|\psi_{i'}\rangle$ from the measurement. Conversely, if one of the amplitudes is unity, i.e., there exists an index value i'' such that $c_{i''} = 1$, then if the state is properly normalized, the result of the measurement is guaranteed to be the corresponding eigenstate, $|\psi_{i''}\rangle$.

Consider a 3-qubit quantum memory register that initially is in the state

$$\begin{aligned} |\psi\rangle = & c_0|000\rangle + c_1|001\rangle + c_2|010\rangle + c_3|011\rangle + c_4|100\rangle + c_5|101\rangle \\ & + c_6|110\rangle + c_7|111\rangle \end{aligned} \tag{1.27}$$

Table 1.3 Probabilities of obtaining the eight distinct triples of values when three qubits are read in the computational basis

Qubit A	Qubit B	Qubit C	Probability
$ 0\rangle$	$ 0\rangle$	$ 0\rangle$	$ c_0 ^2$
$ 0\rangle$	$ 0\rangle$	$ 1\rangle$	$ c_1 ^2$
$ 0\rangle$	$ 1\rangle$	$ 0\rangle$	$ c_2 ^2$
$ 0\rangle$	$ 1\rangle$	$ 1\rangle$	$ c_3 ^2$
$ 1\rangle$	$ 0\rangle$	$ 0\rangle$	$ c_4 ^2$
$ 1\rangle$	$ 0\rangle$	$ 1\rangle$	$ c_5 ^2$
$ 1\rangle$	$ 1\rangle$	$ 0\rangle$	$ c_6 ^2$
$ 1\rangle$	$ 1\rangle$	$ 1\rangle$	$ c_7 ^2$

where $\sum_{i=0}^7 |c_i|^2 = 1$. For convenience imagine labeling the leftmost qubit A , the middle qubit B , and the rightmost qubit C . When we do a complete measurement of all the qubits in this memory register, we expect to find the result $|i\rangle$ with probability $|c_i|^2$. That is we obtain the results shown in Table 1.3.

1.6.2.2 Partial Readout

Alternatively, suppose we measure only the middle qubit, B , and find it to be in state “ $|1\rangle$ ”. Such a measurement projects the qubits into a form that constrains the middle qubit to be $|1\rangle$, but leaves the other qubits indeterminate (since neither qubits A nor C were measured). Moreover, the resulting state must still be properly normalized. Hence, after the measurement, the state of the 3-qubit memory register is $\frac{c_2|010\rangle + c_3|011\rangle + c_6|110\rangle + c_7|111\rangle}{\sqrt{|c_2|^2 + |c_3|^2 + |c_6|^2 + |c_7|^2}}$.

1.6.3 Alternative Bases

We do not have to view the contents of a quantum memory register as being in the computational basis however. A basis for an n -qubit quantum memory register is any complete orthonormal set of eigenstates such any n -qubit state can be written as a superposition of states taken from only this set. The computational basis states for a single qubit memory register are $|0\rangle$ and $|1\rangle$, and for an n -qubit quantum memory register the tensor product of all combinations of these, i.e. $\{|0\rangle, |1\rangle\}^{\otimes n} \equiv \{|00\dots 0\rangle, |00\dots 1\rangle, \dots, |11\dots 1\rangle\}$. However, many other bases are possible, including those related to rotations of the single qubit computational basis states and tensor products thereof, and entirely unusual choices such as entangled multi-qubit states, e.g., the Bell basis. Table 1.4 shows some possible bases for a rudimentary (2-qubit) quantum memory register. The first three bases are related to rotations of the single computational basis states, but the fourth basis is a basis consisting of purely 2-qubit states, which is nevertheless a proper basis for 2-qubit states.

Table 1.4 Some examples of different bases for 2-qubit quantum memory register. Note that the Bell basis is defined over entangled 2-qubit states. The other bases shown are unitary transformations of the computational basis states $|0\rangle$ and $|1\rangle$

Basis	Eigenstates
θ° Rotated	$ \bar{0}\rangle = \cos\theta 0\rangle + \sin\theta 1\rangle$ $ \bar{1}\rangle = \cos\theta 0\rangle - \sin\theta 1\rangle$
Diagonal	$ \nwarrow\rangle = \frac{1}{\sqrt{2}}(0\rangle + 1\rangle)$ $ \nearrow\rangle = \frac{1}{\sqrt{2}}(0\rangle - 1\rangle)$
Chiral	$ \circlearrowleft\rangle = \frac{1}{\sqrt{2}}(0\rangle + i 1\rangle)$ $ \circlearrowright\rangle = \frac{1}{\sqrt{2}}(0\rangle - i 1\rangle)$
Bell	$ \beta_{00}\rangle = \frac{1}{\sqrt{2}}(00\rangle + 11\rangle)$ $ \beta_{01}\rangle = \frac{1}{\sqrt{2}}(01\rangle + 10\rangle)$ $ \beta_{10}\rangle = \frac{1}{\sqrt{2}}(00\rangle - 11\rangle)$ $ \beta_{11}\rangle = \frac{1}{\sqrt{2}}(01\rangle - 10\rangle)$

However, the proper way to think of this is that there is an *observable*, \mathcal{A} say, whose eigenvectors correspond to the possible n -bit computational eigenstates, $|00\dots 0\rangle$, $|00\dots 1\rangle$, ..., $|11\dots 1\rangle$. To remind ourselves that these are eigenvectors of observable \mathcal{A} we'll rename these eigenvectors $|a_i\rangle$ and call them the “ a ”-basis.

However, we do not *have* to use the computational basis to represent a state. Any complete orthonormal set of eigenvectors for an n -qubit state will do. In some circumstances, it is convenient to re-represent a given state in a new basis that simplifies some subsequent calculation. For example, suppose we are interested in calculating the expected outcomes of an observable property of an n -qubit state other than its bit values. Let us call the observable operator in which we are interested \mathcal{B} having eigenvectors $|b_j\rangle$. Measuring observable \mathcal{B} amounts to measuring the state $|\psi\rangle$ in the “ b ”-basis. The question is given a representation of a particular state $|\psi\rangle$ in the “ a ”-basis, how would this same state be represented in the “ b ”-basis? Knowing this we can then calculate the expected outcome from measuring observable \mathcal{B} of $|\psi\rangle$.

First we need to know how the eigenvectors in the two bases are related. In particular, imagine creating the operator, U , defined as follows:

$$U = \sum_k |b_k\rangle \langle a_k| \quad (1.28)$$

An operator, U , of this form is unitary and induces the following mapping between the “ a ”-basis and the “ b ”-basis:

$$\begin{aligned} |b_1\rangle &= U|a_1\rangle \\ |b_2\rangle &= U|a_2\rangle \\ &\vdots \\ |b_{2^n}\rangle &= U|a_{2^n}\rangle \end{aligned} \quad (1.29)$$

Hence for each eigenvector in the “*a*”-basis there is a corresponding eigenvector in the “*b*”-basis.

1.6.4 Change of Basis

A given quantum state is not wedded to any particular basis. The same state can be interpreted as different superposition states of eigenstates from a completely different basis. Once this is understood, it makes it easier to appreciate why we might choose to observe a given state in a basis other than the computational basis.

Typically, in this book, we represent the states of a quantum memory register in the *computational* basis. That is, we write an n -qubit pure state in the form:

$$|\psi\rangle = \sum_i c_i |i\rangle \quad (1.30)$$

where $|i\rangle$ is the binary representation of integer i padded on the left with zeroes, if necessary, to make a full complement of n bits, and c_i is the complex amplitude with which eigenstate $|i\rangle$ contributes to the superposition, such that $\sum_i |c_i|^2 = 1$.

In the computational basis representation it is easy to calculate the probability of measuring the quantum memory register to be in a certain bit-string configuration, since configuration $|i\rangle$ will be found with probability $|c_i|^2$.

1.6.4.1 Change of Basis for a State

Thus a given state $|\psi\rangle$ can be written in either the “*a*”-basis or the “*b*”-basis. Specifically, we have:

$$|\psi\rangle = \sum_i \alpha_i |a_i\rangle = \sum_j \beta_j |b_j\rangle \quad (1.31)$$

where the amplitudes α_i and β_j are given by:

$$\alpha_i = \langle a_i | \psi \rangle \quad (1.32)$$

$$\beta_j = \langle b_j | \psi \rangle \quad (1.33)$$

Equation (1.29) tells us how to compute each “*b*”-basis vector $|b_k\rangle$ given its corresponding “*a*”-basis vector, $|a_k\rangle$, and U . So all we need to do now is to learn how to compute β_j . We can rewrite β_j as follows:

$$\begin{aligned} \beta_j &= \langle b_j | \psi \rangle = \langle b_j | \left(\sum_i |a_i\rangle \langle a_i| \right) | \psi \rangle = \sum_i \langle b_j | a_i \rangle \langle a_i | \psi \rangle \\ &= \sum_i \langle a_j | U^\dagger | a_i \rangle \langle a_i | \psi \rangle = \sum_i (U^\dagger)_{ji} \alpha_i \end{aligned} \quad (1.34)$$

where we have used the facts that $(\sum_i |a_i\rangle\langle a_i|) = \mathbb{1}$, the identity operator, and $|b_j\rangle = U|a_j\rangle$, which implies $\langle b_j| = \langle a_j|U^\dagger$. The last line of (1.34) is the usual form for the dot product of a matrix (i.e., U^\dagger) with a column vector (i.e., the column vector of amplitudes α_i). Hence, the column vector of amplitudes β_j representing the state $|\psi\rangle$ in the $\{|b_j\rangle\}$ (i.e., “new”) basis is related to column vector of amplitudes c_i in the $\{|a_i\rangle\}$ (i.e., “old”) basis via the matrix equation:

$$|\psi\rangle_{\text{``b''-basis}} = U^\dagger |\psi\rangle_{\text{``a''-basis}} \quad (1.35)$$

where U is the operator define by (1.28) and which induced the connection between the bases given in (1.29).

Example: Linear versus Diagonal Polarization Bases Imagine a qubit encoded in the linear polarization state of a photon. By this we mean if we think of light as an oscillating electromagnetic wave, the plane in which the electric field component of that wave is oscillating, i.e., the state of its linear polarization, encodes our qubit. If the plane is “vertical” (with respect to some arbitrary axis in physical space) we say the qubit is a logical 0 $|\downarrow\rangle$. Conversely, if the plane in which the electric field is oscillating is “horizontal” (with respect to the same axis in physical space) we say the qubit is a logical 1 $|\leftrightarrow\rangle$. Note, just to reinforce your understanding of the geometry on the Bloch sphere, on the Bloch sphere the states representing vertical and horizontal polarization ($|0\rangle \equiv |\downarrow\rangle$ and $|1\rangle \equiv |\leftrightarrow\rangle$) correspond to the North Pole and South Pole respectively (i.e., at 180° separation). But in physical space the planes representing vertical and horizontally polarized photons lie at 90° to one another.

Now let’s imagine switching to a polarization basis that it tilted at 45° with respect to the original basis. The new basis kets are $|\swarrow\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ (corresponding to a photon whose plane of polarization is tipped at $+45^\circ$ to the old plane of polarization) and $|\nwarrow\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ (corresponding to a photon whose plane of polarization is tipped at -45° to the old plane of polarization). Thus following the recipe given above, the unitary matrix that maps a state in the old basis to its equivalent in the new basis is

$$\begin{aligned} U &= |\swarrow\rangle\langle 0| + |\nwarrow\rangle\langle 1| = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\langle 0| + \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\langle 1| \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \end{aligned} \quad (1.36)$$

1.6.4.2 Change of Basis for an Operator

Just as we can view quantum states in different bases, so too can we view quantum operators in different bases. Consider some operator, \mathcal{O} say, given initially in the

“ a ”-basis. By inserting the identity operator twice we can write:

$$\begin{aligned}
 \langle b_k | \mathcal{O} | b_\ell \rangle &= \langle b_k | \left(\sum_m |a_m\rangle\langle a_m| \right) \cdot X \cdot \left(\sum_m |a_m\rangle\langle a_m| \right) |b_\ell \rangle \\
 &= \sum_m \sum_n \langle b_k | a_m \rangle \langle a_m | \mathcal{O} | a_n \rangle \langle a_n | b_\ell \rangle \\
 &= \sum_m \sum_n \langle a_k | U^\dagger | a_m \rangle \langle a_m | \mathcal{O} | a_n \rangle \langle a_n | U | a_\ell \rangle \\
 &= \sum_m \sum_n (U^\dagger)_{km} \mathcal{O}_{mn} U_{n\ell}
 \end{aligned} \tag{1.37}$$

This has the form of a “similarity transform”, which is encountered routinely in linear algebra. That is, in matrix form, we can write:

$$\mathcal{O}_{\text{``}b\text{''}-\text{basis}} = U^\dagger \cdot \mathcal{O}_{\text{``}a\text{''}-\text{basis}} \cdot U \tag{1.38}$$

Thus, given an operator in the “ a ”-basis equation (1.38) shows how to transform it into the “ b ”-basis.

1.6.5 Observing in an Arbitrary Basis

So far we have equated the act of observing a quantum memory register with the act of reading its bit values, or equivalently, measuring its qubits in the computational basis. However, a given quantum state does not have a unique interpretation: any state—even the state of a quantum memory register—can be pictured as different superposition states over different bases. Consequently, although most of the time in quantum computing it seems natural to read a quantum memory register in the computational basis, in some circumstances it might be more natural to read the quantum memory register with respect to some other basis.

Consider what this means in the case of a single qubit. Although a qubit might be defined initially with respect to the computational basis, i.e., as a state of the form $a|0\rangle + b|1\rangle$, where $|0\rangle$ is the North pole, and $|1\rangle$ the South pole, of the Bloch sphere, this same state can be re-represented in infinitely many other ways simply by changing which vectors we regard as the “basis” vectors.

Picture the state of a single qubit as a point on the surface of the Bloch sphere. Define a vector whose origin lies at the center of the Bloch sphere and whose tip touches this same point on the surface. Imagine keeping this vector in a fixed orientation but rotating the Bloch sphere surrounding it. Although the vector has not changed, the coordinates of its tip with respect to the x -, y -, and z -axes of the Bloch sphere have changed, and so the state at the tip of the vector appears to have changed. But however we rotate the axes around we can always pick an observation-axis that is on a line from the qubit state, $|\psi\rangle$ say (on the surface of the Bloch sphere),

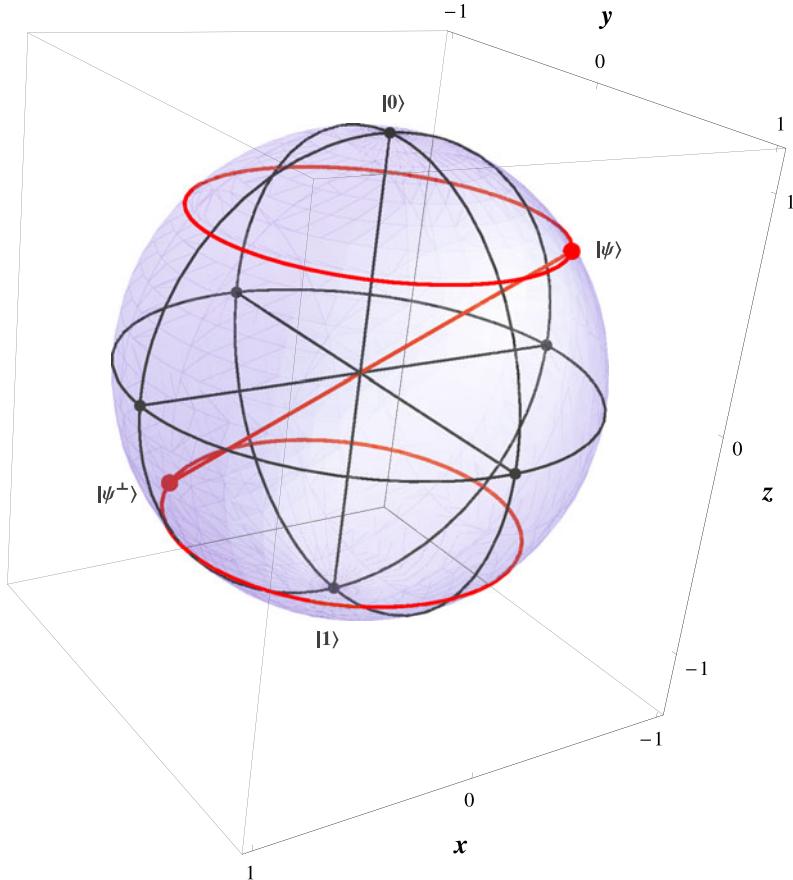


Fig. 1.7 Measuring the state of a qubit initially in state $a|0\rangle + b|1\rangle$ along an axis passing through states $|\psi\rangle$ and $|\psi^\perp\rangle$ corresponds to measuring the qubit in the $\{|\psi\rangle, |\psi^\perp\rangle\}$ basis

through the center of the Bloch sphere piercing the opposite side. The (antipodal) point where this line pierces the Bloch sphere corresponds to the quantum state $|\psi_\perp\rangle$, which is orthogonal to $|\psi\rangle$. Thus the basis made from states $\{|\psi\rangle, |\psi^\perp\rangle\}$ is equally as good as the computational basis, $\{|0\rangle, |1\rangle\}$, for describing single qubit states. Thus, it is possible to measure our qubit in this alternate $\{|\psi\rangle, |\psi^\perp\rangle\}$ basis too. Such a measurement is illustrated in Fig. 1.7.

1.7 Quantum Parallelism and the Deutsch-Jozsa Algorithm

Having introduced the main ideas of quantum computing we end this chapter by describing our first quantum computation—deciding whether a given function has a certain property using the Deutsch-Jozsa quantum algorithm. This computation

cannot be solved as efficiently using any classical computer. It is not an especially *useful* computation, mind you. In fact, it is rather contrived. Nevertheless it illustrates many of the key steps in a typical quantum computation.

1.7.1 The Problem: Is $f(x)$ Constant or Balanced?

The problem, originally formulated by Cleve, Ekert, Macchiavello, and Mosca [112] as a variant of one by Deutsch and Jozsa [138] is this: Let x be any n -bit binary number and let $f(x)$ be a function that returns a single binary output (i.e., 0 or 1) for each value of x . Furthermore, we are promised that $f(x)$ behaves in only one of two possible ways: either $f(x)$ returns the same value for all binary inputs (in which case $f(x)$ is said to be *constant*), else $f(x)$ returns one bit value for half its inputs and the other bit value for the other half of its inputs (in which case $f(x)$ is said to be *balanced*). Finally, we are not allowed to inspect the mathematical definition of $f(x)$. Instead, we imagine $f(x)$ is given to us as a “black-box” function that acts in such a way that, when given the input x , the black box responds with the correct value for $f(x)$. Our task is to decide, using the fewest calls to the black-box, whether $f(x)$ is constant or balanced. Note that the decision does require us to exhibit the values of $f(x)$. Rather it only concerns a property those values possess, namely, whether they are all the same, or whether half have one bit value and half the other.

Using our conventional (classical) thinking, the number of times we would seem to need to call the black box is clear. There are a total of 2^n possible bit string inputs that can be made from n bits. Thus, we will need to check at least one more than half of them, i.e., $(\frac{1}{2} \times 2^n) + 1 = 2^{n-1} + 1$, to be able to decide with certainty whether $f(x)$ is constant or balanced. Note that we don’t have to check all the 2^n input bit strings because we were promised that $f(x)$ is either constant or balanced. Thus, discovering $f(x)$ is non-constant is enough to conclude it must be the other possibility, namely, balanced. Even though we can avoid checking all inputs, classically, as larger and larger decision problems are considered the number of elementary calls to the black box would still seem to have to grow *exponentially* in the length of the input bit string n . In contrast, as we shall show, using a quantum computer, and a quantum implementation of the black-box that encodes $f(x)$, we can decide the question of whether $f(x)$ is constant or balanced in just *one* call to the black-box! This represents an *exponential speedup* in obtaining the decision—which is amazing!

Let’s begin by looking at the simplest instance of such a decision problem when the input bit string is just a single bit (i.e., when $n = 1$). In this case, the decision problem can be stated as:

$$\text{decision}(f) = \begin{cases} \text{constant} & \text{iff } f(0) = f(1) \\ \text{balanced} & \text{iff } f(0) \neq f(1) \end{cases} \quad (1.39)$$

Using a classical computer we could decide the matter by first computing $f(0)$ and then computing $f(1)$ and then comparing the results to determine whether $f(0) =$

$f(1)$ or $f(0) \neq f(1)$. This approach would require *two* calls to the black box to make the decision regarding whether $f(x)$ is constant or balanced.

A quantum computer can solve this problem differently using a technique called *quantum parallelism*. To understand how quantum parallelism works we must first figure out how to define the action of the black-box that encodes knowledge of the function $f(x)$ in a manner that is consistent with quantum mechanics.

1.7.2 Embedding $f(x)$ in a Quantum Black-Box Function

On the face of it you might think that the black-box could be defined as performing the mapping $|x\rangle \rightarrow |f(x)\rangle$ since, in the special (i.e., $n = 1$) case we are considering both x and $f(x)$ are single bits. However, this won't do because as we saw in Sect. 1.5 quantum mechanical evolutions are described by unitary, and hence logically reversible, operations. For an operation to be logically reversible, each distinct input ought to be mapped to a distinct output and vice versa. Unfortunately, describing the black-box as performing the mapping $|x\rangle \rightarrow |f(x)\rangle$ is not necessarily logically reversible. If $f(x)$ happens to be constant then both possible values for $|x\rangle$ would be mapped into the same value for $f(x)$. So if the operation performed by our black-box is to be described quantum mechanically, the specification $|x\rangle \rightarrow |f(x)\rangle$ won't do. Strike one!

Ok well how about introducing an extra register—one to hold the input and the other to hold the output? The starting configuration could be $|x\rangle|0\rangle$, with the second register initialized to $|0\rangle$, which we can think of as analogous to a blank piece of paper on which the correct answer for $f(x)$ is to be written. In this case, our black-box would perform the operation $|x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$. Since the input, $|x\rangle$, is now recorded explicitly in the output we can always invert this mapping unambiguously, whatever the value of $f(x)$. Unfortunately, we're still not done because for a mapping between bit strings to be unitary (as quantum mechanics requires) we need to a complete mapping, i.e., a specification how each possible binary input is mapped to a distinct output. Since, the specification of the black-box as performing the operation $|x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$ only accounts for inputs that end in $|0\rangle$ it is missing half the possible inputs that could be given to it. Hence, the specification is incomplete, and therefore, won't do either. Strike two!

Thus to ensure our description of the black-box is unitary we need to specify how input states ending in $|0\rangle$ and states ending in $|1\rangle$ are to be mapped to outputs. Thus the right way to define the black-box operation is as

$$|x\rangle|y\rangle \longrightarrow |x\rangle|y \oplus f(x)\rangle. \quad (1.40)$$

The $y \oplus f(x)$ operation is the exclusive-OR operation, and is computed as shown in Table 1.5. When $y = 0$, $y \oplus f(x) = f(x)$, so the definition $|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$ includes the case $|x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$. But by defining the operation with the second qubit allowed to be either $|0\rangle$ or $|1\rangle$ we ensure that our description of the action of the black-box is a unitary (reversible) operation, which specifies a complete mapping between all possible 2-qubit binary inputs and all possible 2-qubit

Table 1.5 Truth table of the exclusive-OR (\oplus) operation. This is different from the usual OR operation (\vee) in that $1 \vee 1 = 1$ whereas $1 \oplus 1 = 0$

x	$f(x)$	$y \oplus f(x)$
0	0	0
0	1	1
1	0	1
1	1	0

binary outputs, and hence is implementable quantum mechanically. The operation $|x\rangle|y\rangle \xrightarrow{f\text{-c-N}} |x\rangle|y \oplus f(x)\rangle$ is sometimes called an “ f -controlled-NOT” operation (f -c-N) since one way to think of it is that the value of $f(x)$ controls whether or not the value of y is negated.

1.7.3 Moving Function Values Between Kets and Phase Factors

Armed with our quantum black-box, which encapsulates the knowledge of $f(x)$, we are now ready to tackle the decision problem regarding whether $f(x)$ is constant or balanced.

If we restricted ourselves to inputting only quantum states corresponding to the “classical” binary inputs $|0\rangle|0\rangle$, $|0\rangle|1\rangle$, $|1\rangle|0\rangle$, and $|1\rangle|1\rangle$, to our quantum black box then our quantum method would confer no advantage over what we can do classically. The magic happens when we use quantum states corresponding to non-classical inputs. Specifically, consider what happens under the action of the f -controlled-NOT operation when the input is $|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. The transformation effected is shown in (1.41)

$$|x\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \xrightarrow{f\text{-c-N}} |x\rangle \frac{1}{\sqrt{2}}(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) \quad (1.41)$$

As we are only considering the simplest ($n = 1$) instance of the decision problem at this time, the argument of $f(x)$, i.e., x , can be only 0 or 1, and the value of $f(x)$ is also only 0 or 1. So we can write out a table showing how the values of x , $f(x)$, and the right hand side of (1.41) are related: Notice that the table also contains a fourth column corresponding to the value of the expression $(-1)^{f(x)}|x\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Remarkably, for all pairs of 2-bit binary inputs, the value returned by the expression $|x\rangle \frac{1}{\sqrt{2}}(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle)$ is, as shown in Table 1.6, identical to the value returned by the expression $(-1)^{f(x)}|x\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Hence—drum roll please—the two expressions are equally good mathematical descriptions of the output quantum state after the f -controlled-NOT operation has been applied. Thus, we could equally well describe the transformation the f -controlled-NOT operation has achieved as:

$$|x\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \xrightarrow{f\text{-c-N}} (-1)^{f(x)}|x\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (1.42)$$

Table 1.6 By considering the possible values of x , $f(x)$ and the right hand side of (1.41) recognize an equivalent way to write the equation

x	$f(x)$	$ x\rangle \frac{1}{\sqrt{2}}(0\rangle + f(x)\rangle) - 1\rangle + f(x))$	$(-1)^{f(x)} x\rangle \frac{1}{\sqrt{2}}(0\rangle - 1\rangle)$
0	0	$ 0\rangle(0\rangle - 1\rangle)$	$ 0\rangle(0\rangle - 1\rangle) = 0\rangle(0\rangle - 1\rangle)$
0	1	$ 0\rangle(1\rangle - 0\rangle)$	$- 0\rangle(0\rangle - 1\rangle) = 0\rangle(1\rangle - 0\rangle)$
1	0	$ 1\rangle(0\rangle - 1\rangle)$	$ 1\rangle(0\rangle - 1\rangle) = 1\rangle(0\rangle - 1\rangle)$
1	1	$ 1\rangle(1\rangle - 0\rangle)$	$- 1\rangle(0\rangle - 1\rangle) = 1\rangle(1\rangle - 0\rangle)$

Thus, with no physical action whatsoever taking place, we can simply re-interpret what mathematical transformation we have achieved. This re-interpretation of the output state allows us to regard the value of the function $f(x)$ as being moved from inside the ket (in (1.41)) to being in the phase factor (in (1.42)). This is very important because we saw in Sect. 1.4.3 that quantum mechanical interference effects can change the relative probabilities of various outcomes. What we will do next is engineer these interference effects to enhance or suppress various possible outcomes depending on whether $f(x)$ is constant or balanced!

1.7.4 Interference Reveals the Decision

To achieve our desired interference effect we take the interpretation of the $f =$ controlled-NOT transformation defined in (1.42) and we specialize the input $|x\rangle$ to be $|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. We can create this state by applying a Walsh-Hadamard gate to just the first qubit prepared initially in the state $|0\rangle$, i.e., $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. With this specialization, the transformation we perform is therefore:

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \xrightarrow{f\text{-c-N}} \frac{1}{\sqrt{2}}\left((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\right) \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (1.43)$$

Next we apply a Walsh-Hadamard gate to just the first qubit again. This results in the transformation:

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \xrightarrow{f\text{-c-N}} \frac{1}{\sqrt{2}}\left((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\right) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (1.44)$$

Summarizing all the steps in the Deutsch-Jozsa algorithm:

$$\begin{aligned} |0\rangle|1\rangle &\xrightarrow{H\otimes H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ &\xrightarrow{f\text{-c-N}} \frac{1}{\sqrt{2}}\left((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\right) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned}$$

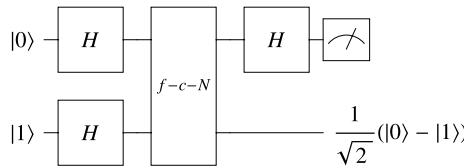


Fig. 1.8 Quantum circuit implementing the Deutsch-Jozsa algorithm. The black-box function $f(x)$ accepts a single bit x and returns 0 or 1. If the returned values are the same $f(x)$ is “constant”. Otherwise $f(x)$ is “balanced”. The function $f(x)$ is implemented by way of the Deutsch-Jozsa oracle f -controlled-NOT ($f\text{-}c\text{-}N$). This implements the transformation $|x\rangle|y\rangle \xrightarrow{f\text{-}c\text{-}N} |x\rangle|y \oplus f(x)\rangle \equiv (-1)^{f(x)}|x\rangle|y\rangle$ when $|y\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Using the Deutsch-Jozsa algorithm we can decide whether $f(x)$ is constant or balanced using a single call to the oracle

$$\xrightarrow{H^{\otimes 2}} \left[\left(\frac{1}{2}(-1)^{f(0)} + \frac{1}{2}(-1)^{f(1)} \right) |0\rangle + \left(\frac{1}{2}(-1)^{f(0)} - \frac{1}{2}(-1)^{f(1)} \right) |1\rangle \right] \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (1.45)$$

Inspection of the amplitudes of the $|0\rangle$ and $|1\rangle$ components of the first qubit suggest that if this qubit is read (in the computational basis) then if $f(x)$ is constant, i.e., if $f(0) = f(1)$, then we will find the first qubit in state $|0\rangle$. Else if $f(x)$ is balanced, i.e., $f(0) \neq f(1)$, then we will find the first qubit in state $|1\rangle$. This means we can determine whether $f(x)$ is constant or balanced in just one call to the black-box (when using quantum inputs) versus two calls to the black-box (if using classical bit value inputs). The quantum circuit implementing the Deutsch-Jozsa algorithm is shown in Fig. 1.8. This is interesting but not that dramatic. To determine what scaling we’re actually seeing we need to consider the relative costs of the quantum and classical methods as we scale up to larger problem instances.

1.7.5 Generalized Deutsch-Jozsa Problem

The aforementioned decision problem only pertains to a function $f(x)$ that has a single bit input and a single bit output. In this case we obtain a factor of two speedup over the naive classical algorithm for the solving the same problem. How does this speedup change if we allow f to accept an n -bit input instead of just a single bit input?

To formalize the question, let $\mathbf{x} = x_1 x_2 \dots x_n$ be an n -bit binary string with binary values x_1, x_2, \dots, x_n . Thus, \mathbf{x} represents the bit string corresponding to any integer in the range 0 to $2^n - 1$ inclusive. Let $f(\mathbf{x})$ be a function that accepts an n -bit input \mathbf{x} and returns a single bit output, i.e., 0 or 1. We are promised that $f(\mathbf{x})$ is one of only two kinds of function, namely, “constant” or “balanced”. An n -bit function $f(\mathbf{x})$ is “constant” if it returns the same bit value on all 2^n possible inputs, and

“balanced” if it returns 0 on exactly half its possible inputs, and 1 on the other half of inputs. Note that the promise is our guarantee that the only types of functions under consideration are constant functions and balanced functions and no others. With this promise in mind, our challenge is to decide whether $f(\mathbf{x})$ is constant or balanced using the fewest calls to the oracle.

Classically, in the worst case, we will have to call the oracle a total of $\frac{1}{2}2^n + 1 = 2^{n-1} + 1$ times. This is because we cannot know for sure that $f(\mathbf{x})$ is constant until we have evaluated $f(\mathbf{x})$ on one more than half its possible inputs. At that point if all the returned values are the same, the promise allows us to conclude that $f(\mathbf{x})$ is constant. However, if in the course of performing these $2^{n-1} + 1$ evaluations we find any two inputs that yield different values for the function, then the promise allows us to conclude the given $f(\mathbf{x})$ is balanced. So, given the promise, on average deciding that $f(\mathbf{x})$ is balanced is easier than deciding it is constant. But in the worst case (when we are unlucky enough that even though $f(\mathbf{x})$ is balanced the first 2^{n-1} inputs we tried happened to be those for which $f(\mathbf{x})$ returned the same value), we need to test one more than half the values to be sure. Since the classical algorithm needs $2^{n-1} + 1$ calls to the oracle, the classical complexity is *exponential* in the number of bits, n .

Can do better using a quantum algorithm? As you will see shortly, it turns out that there is a quantum algorithm, the “Generalized Deutsch-Jozsa Algorithm”, for solving this same decision problem that only needs to make a *single* call to the oracle, regardless of n . This amounts to an exponential speedup over what is possible classically! This is an astonishing difference in complexity between a quantum computer and a classical computer on the same problem. So even though the actual problem solved is rather arcane and esoteric, nevertheless, it illustrates the enormous potential of quantum computers to outperform classical computers on certain computational problems.

The best way to see how the Generalized Deutsch-Jozsa algorithm works is to start with the quantum circuit that implements it and to walk through the state transformations it enacts. This will allow us to compute the mathematical form of the final state that is synthesized by the circuit and hence determine how a measurement made upon this final state can reveal the decision regarding whether $f(\mathbf{x})$ is constant or balanced.

The quantum circuit for the generalized Deutsch-Jozsa algorithm is shown in Fig. 1.9 and the associated algorithm is as follows:

Generalized Deutsch-Jozsa Algorithm Given an oracle, or black-box quantum function, $f(\mathbf{x})$ that accepts an n -bit binary string input, $\mathbf{x} = x_1x_2 \cdots x_n$, and the promise that $f(\mathbf{x})$ is either constant or balanced, decide which is the case using the fewest calls to the oracle.

1. Create an $(n + 1)$ -qubit quantum register having n control qubits, each in state $|0\rangle$, and one ancilla qubit in state $|1\rangle$.

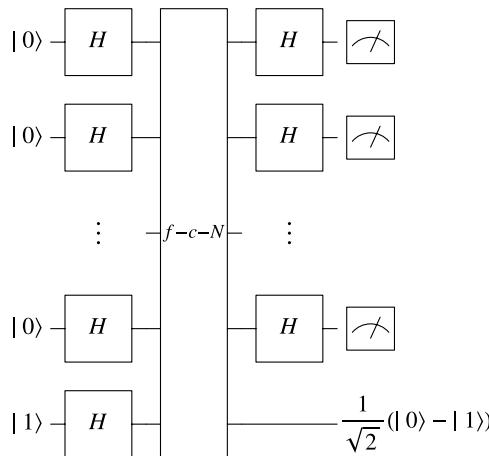


Fig. 1.9 Quantum circuit implementing the Generalized Deutsch-Jozsa algorithm. The black-box function $f(\mathbf{x})$ accepts an n -bit input \mathbf{x} and returns the single bit 0 or 1. We are promised that $f(\mathbf{x})$ is either “constant” (i.e., returns the same value on all its possible 2^n inputs) or “balanced” (i.e., returns 0 on half of its possible inputs and 1 on the other half of its possible inputs). The function $f(\mathbf{x})$ is implemented by way of the Generalized Deutsch-Jozsa oracle f -controlled-NOT ($f\text{-c-N}$). This implements the transformation $|\mathbf{x}\rangle|y\rangle \xrightarrow{f\text{-c-N}} |\mathbf{x}\rangle|y \oplus f(\mathbf{x})\rangle$. In turn, this is equivalent to $(-1)^{f(\mathbf{x})}|\mathbf{x}\rangle|y\rangle$ when $|y\rangle$ is specialized to be the state $|y\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Using the Generalized Deutsch-Jozsa algorithm we can decide whether $f(\mathbf{x})$ is constant or balanced using a *single* call to the oracle. A classical computer would need to use $\frac{1}{2}2^n + 1$ calls to the oracle to arrive at the same decision. Hence, in this case, a quantum computer running the Generalized Deutsch-Jozsa algorithm is exponentially more efficient than a classical computer. Hence, the Generalized Deutsch-Jozsa algorithm, although not particularly useful as a practical algorithm, illustrates the potential for enormous complexity advantages of quantum computers over classical computers on certain problems

2. Apply a Walsh-Hadamard gate to each qubit. That is, perform the operation:

$$|00\dots0\rangle|1\rangle \xrightarrow{H^{\otimes(n+1)}} \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}=0}^{2^n-1} |\mathbf{x}\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (1.46)$$

3. Then apply the Generalized Deutsch-Jozsa oracle.

$$\xrightarrow{f\text{-c-N}} \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}=0}^{2^n-1} (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (1.47)$$

4. Apply a Walsh-Hadamard gate to the top n qubits.

$$\xrightarrow{H^{\otimes n} \otimes \mathbb{I}} \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}=0}^{2^n-1} (-1)^{f(\mathbf{x})} \underbrace{(H \otimes H \otimes \dots \otimes H)}_n |\mathbf{x}\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (1.48)$$

$$\equiv \frac{1}{\sqrt{2^n}} \left(\sum_{x=0}^{2^n-1} (-1)^{f(x)} \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} (-1)^{\mathbf{x}\cdot\mathbf{z}} |z\rangle \right) \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \quad (1.49)$$

$$\equiv \frac{1}{2^n} \left(\sum_{x=0}^{2^n-1} \sum_{z=0}^{2^n-1} (-1)^{f(x)} (-1)^{\mathbf{x}\cdot\mathbf{z}} |z\rangle \right) \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \quad (1.50)$$

5. Measure the top n qubits in the computational basis. If the first n qubits are found to be in state $|0\rangle = |00\dots 0\rangle$, $f(\mathbf{x})$ is “constant”. If any other pattern of values is obtained for the first n qubits, then $f(\mathbf{x})$ is “balanced”.

The algorithm works as follows: in the first step we initialize n control qubits to be in state $|00\dots 0\rangle$ and we initialize a single ancilla qubit to be in state $|1\rangle$. Next we apply our oracle, i.e., the f -controlled-NOT gate. This acts on n control bits (which hold the value of the input “ \mathbf{x} ”) and one target qubit (which starts off in state $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$). The transformation the oracle performs is:

$$\underbrace{|x\rangle}_{n \text{ qubits}} \otimes \underbrace{|y\rangle}_{1 \text{ qubit}} \xrightarrow{f\text{-c-N}} |x\rangle \otimes |y \oplus f(x)\rangle \equiv (-1)^{f(x)} |x\rangle |y\rangle \quad (1.51)$$

Next we apply a Walsh-Hadamard gate to the top n qubits only. This is perhaps the hardest part of the algorithm to understand because it is not immediately obvious why $H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} (-1)^{\mathbf{x}\cdot\mathbf{z}} |z\rangle$. To see why this is true let's start by considering a simple 3-qubit instance of the problem.

$$\begin{aligned} (H \otimes H \otimes H)|x\rangle &= (H \otimes H \otimes H)|x_1 x_2 x_3\rangle \\ &= H|x_1\rangle \otimes H|x_2\rangle \otimes H|x_3\rangle \\ &= \frac{1}{\sqrt{2^3}} (|0\rangle + (-1)^{x_1}|1\rangle) \otimes (|0\rangle + (-1)^{x_2}|1\rangle) \\ &\quad \otimes (|0\rangle + (-1)^{x_3}|1\rangle) \\ &= \frac{1}{\sqrt{2^3}} \left(|000\rangle + (-1)^{x_3}|001\rangle + (-1)^{x_2}|010\rangle \right. \\ &\quad \left. + (-1)^{x_2+x_3}|011\rangle + (-1)^{x_1}|100\rangle \right. \\ &\quad \left. + (-1)^{x_1+x_3}|101\rangle + (-1)^{x_1+x_2}|110\rangle + (-1)^{x_1+x_2+x_3}|111\rangle \right) \\ &= \frac{1}{\sqrt{2^3}} \sum_{z_1=0}^1 \sum_{z_2=0}^1 \sum_{z_3=0}^1 (-1)^{x_1 z_1 + x_2 z_2 + x_3 z_3} |z_1 z_2 z_3\rangle \\ &= \frac{1}{\sqrt{2^3}} \sum_{z=0}^{2^3-1} (-1)^{\mathbf{x}\cdot\mathbf{z}} |z\rangle \end{aligned} \quad (1.52)$$

where $\mathbf{x} \cdot \mathbf{z} = x_1 z_1 + x_2 z_2 + x_3 z_3$. The generalization to the case of $H^{\otimes n} |\mathbf{x}\rangle$ (i.e., n qubits) is obvious.

The last step of the algorithm is to measure the first n qubits in the computational basis. We claim that if the n qubits are each found in state $|0\rangle$, then $f(\mathbf{x})$ is constant. Otherwise, $f(\mathbf{x})$ is balanced.

To justify this claim, consider the amplitude of the $|\mathbf{z}\rangle = |\mathbf{0}\rangle = |00\dots\rangle$ component of the final superposition created in step 4 of Generalized Deutsch Jozsa Algorithm:

$$\frac{1}{2^n} \left(\sum_{\mathbf{x}=0}^{2^n-1} \sum_{\mathbf{z}=0}^{2^n-1} (-1)^{f(\mathbf{x})} (-1)^{\mathbf{x} \cdot \mathbf{z}} |\mathbf{z}\rangle \right) \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \quad (1.53)$$

This amplitude is:

$$a_0 = \frac{1}{2^n} \sum_{\mathbf{x}=0}^{2^n-1} (-1)^{f(\mathbf{x})} \quad (1.54)$$

If $f(\mathbf{x})$ is “constant”, i.e., always returns 0 or always returns 1 regardless of the input \mathbf{x} , then the amplitude $a_0 = \pm 1$. Hence, the probability of finding the first n qubits to be in state $|\mathbf{0}\rangle = |00\dots 0\rangle$ is $|\pm 1|^2 = 1$, i.e., certainty! Conversely, if $f(\mathbf{x})$ is “balanced”, there will be exactly as many terms in the sum for a_0 that are -1 as there are that are $+1$. Hence a_0 , will be zero. Thus, if $f(x)$ is “balanced” there is no chance whatsoever of finding the first n qubits to be in state $|\mathbf{0}\rangle = |00\dots 0\rangle$. Thus, a final measurement of the first n qubits reveals the decision as to whether $f(x)$ is “constant” or “balanced”, with only a single call to the oracle! Contrast this with a classical computer that requires $2^{n-1} + 1$ calls to oracle. Hence, a quantum computer is *exponentially* faster than a classical computer at deciding whether $f(x)$ is “constant” or “balanced”.

1.8 Summary

Quantum computing is forcing us to re-think the foundations of computer science. Although theoretical computer science, being based on pure mathematics, is supposed to be free of any assumptions regarding how a computer is implemented, in fact it is not. It is flawed in assuming that the rules by which any computer operates must be those of classical physics. Until recently this assumption was reasonable. But as computer miniaturization is leading us inexorably to smaller and smaller scales, we are close to the point at which we can no longer ignore the fact that the laws of physics that most accurately describe what happens at those small scales is quantum mechanics. And the rules of quantum mechanics are quite unlike the rules that hold sway in the everyday world around us.

In this chapter we have introduced the basic mathematical formalism of quantum computing, and have described several quantum effects that can be harnessed to conceive of algorithms that cannot be run as efficiently on any classical computer. The

most important quantum effects are superposition, interference, non-determinism, and entanglement. Superposition allows a quantum computer to act upon an input state representing an exponential number of different classical inputs simultaneously. Interference can cause the relative proportions of a superposition to change making some outcomes more likely than others. Non-determinism means that we cannot predict with certainty what answer we will get when we read a quantum memory register that exists in a superposition state. However, we can calculate the probabilities with which we expect to see the various possible outcomes. Finally, entanglement is the most quintessentially quantum effect that allows strong correlations to exist between different subsets of qubits such that measurements made (say) on one subset of qubits can affect the likelihood of the outcomes of measurements made on other subsets of qubits, even though they were not “touched” in any direct way.

Whereas when a classical computer completes a computation we are restricted to merely reading its output in the computational basis, in a quantum computer we can choose the measurement basis so as to extract different types of information. Not surprisingly, therefore, we found that a given quantum state can be represented in different ways by using different bases, and we showed how to change from one basis to another. Such basis transformations can reveal insight into the structure of the superposition states or operators with which you are dealing.

We ended the chapter with an example of a simple quantum computation, a decision problem, that cannot be done as efficiently on a classical computer. The trick of moving information between the arguments of ket vectors and phase factors is used in many quantum algorithms, and is especially prevalent in algorithms that involve the quantum Fourier transform (QFT).

1.9 Exercises

1.1 A single qubit in state $|\psi\rangle = a|0\rangle + b|1\rangle$ is normalized iff $|a|^2 + |b|^2 = 1$. Which of the following states of a single qubit are normalized?

1. $\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$
2. $-\frac{i}{3}|0\rangle - \frac{\sqrt{3}}{2}|1\rangle$
3. $-\frac{1}{3}|0\rangle + \frac{2}{3}|1\rangle$
4. $\cos(\theta)|0\rangle - \sin(\theta)|1\rangle$
5. $\cosh(\theta)|0\rangle + i \sinh(\theta)|1\rangle$
6. $e^{i\alpha} \cos(\theta)|0\rangle - e^{i\beta} \sin(\theta)|1\rangle$

1.2 An un-normalized quantum state, i.e., $|\psi\rangle = a|0\rangle + b|1\rangle$ s.t. $|a|^2 + |b|^2 \neq 1$, can be normalized by re-scaling the amplitudes according to:

$$|\psi\rangle = a|0\rangle + b|1\rangle \longrightarrow \frac{a}{\sqrt{|a|^2 + |b|^2}}|0\rangle + \frac{b}{\sqrt{|a|^2 + |b|^2}}|1\rangle$$

Normalize the following un-normalized quantum states.

- (a) $|0\rangle + e^{i\theta}|1\rangle$
- (b) $\frac{1}{2}|0\rangle - \frac{2}{3}|1\rangle$
- (c) $i|00\rangle + |01\rangle + |00\rangle$
- (d) $(|0\rangle + i|1\rangle) \otimes (|0\rangle - i|1\rangle)$

1.3 Compute the probability with which each of the following qubits is found in the state $|0\rangle$ when measured in the computational basis. Be careful as the given state may or may not be properly normalized as given.

1. $\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$
2. $-\frac{i}{3}|0\rangle - \frac{\sqrt{3}}{2}|1\rangle$
3. $\frac{2\sqrt{2}}{3}|0\rangle + \frac{1}{3}|1\rangle$
4. $\frac{i}{2}|0\rangle - \frac{\sqrt{3}}{2}|1\rangle$

1.4 Let $|\bar{0}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|\bar{1}\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Prove that the states $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are orthogonal. Write the state $a|0\rangle + b|1\rangle$ in the $\{|\bar{0}\rangle, |\bar{1}\rangle\}$ basis.

1.5 The memory register of a 3-qubit quantum computer evolves into the state $\frac{1}{3}|001\rangle + \frac{\sqrt{5}}{3}|010\rangle + \frac{1}{\sqrt{3}}|100\rangle$. What is the probability of:

1. Finding the first qubit to be $|1\rangle$?
2. Finding the second qubit to be $|0\rangle$?
3. Finding the last two qubits to be $|00\rangle$?

1.6 Which of the following states are entangled, and which are unentangled?

- (a) $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$
- (b) $\frac{1}{3\sqrt{2}}|00\rangle + \frac{2}{3}|01\rangle + \frac{1}{3\sqrt{2}}|10\rangle + \frac{2}{3}|11\rangle$
- (c) $\frac{1}{6}|00\rangle - \frac{1}{2\sqrt{3}}|01\rangle + \frac{\sqrt{2}}{3}|10\rangle - \sqrt{\frac{2}{3}}|11\rangle$
- (d) $\frac{1}{\sqrt{3}}(|001\rangle + |010\rangle + |100\rangle)$
- (e) $\frac{1}{\sqrt{2}}(|00\rangle - i|10\rangle)$

1.7 Prove that the quantum state $|\psi\rangle$ defined by:

$$\begin{aligned} |\psi\rangle &= \frac{6}{\sqrt{181}}|000\rangle - \frac{4}{\sqrt{181}}|001\rangle + \frac{3}{\sqrt{181}}|010\rangle - 4\sqrt{\frac{3}{181}}|011\rangle \\ &\quad + \sqrt{\frac{2}{181}}|100\rangle - \sqrt{\frac{6}{181}}|101\rangle + \frac{4}{\sqrt{181}}|110\rangle - 4\sqrt{\frac{3}{181}}|111\rangle \end{aligned} \quad (1.55)$$

is properly normalized. Given the state $|\psi\rangle$, what is the probability, when you read $|\psi\rangle$ in the computational basis, of obtaining:

- (a) the result $|010\rangle$?

- (b) the result $|001\rangle$?
- (c) finding the first qubit to be in state $|1\rangle$?
- (d) finding the first and third qubits to both be in state $|0\rangle$?
- (e) finding the first and second qubits to be the same?

1.8 Consider a single qubit in state $|\psi\rangle = \cos(\frac{\theta}{2})|\psi\rangle + e^{i\phi} \sin(\frac{\theta}{2})|1\rangle$ such that $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$. Prove that the state $|\psi\rangle^\perp$ at the antipodal point of the Bloch sphere is orthogonal to $|\psi\rangle$. The antipodal point is found by projecting a straight line from the point on the surface of the Bloch sphere representing $|\psi\rangle$ through the origin to intersect the surface of the Bloch sphere on the opposite side.

1.9 Prove that the expectation value of any observable \mathcal{A} , $\langle\psi|\mathcal{A}|\psi\rangle$, for a quantum system in state $|\psi\rangle$ is no different from that obtained if the state where $e^{i\phi}|\psi\rangle$ instead. That is, prove the claim made in this chapter that overall phase factors have no observable consequence.

1.10 Let \mathcal{Q} be an observable operator for a single qubit described as:

$$\mathcal{Q} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Answer the following questions:

- (a) Which elements of \mathcal{Q} must be real numbers?
- (b) Which elements of \mathcal{Q} can be complex numbers?
- (c) Which two elements of \mathcal{Q} are related?
- (d) What are the eigenvalues of \mathcal{Q} ?
- (e) What is the expectation value $\langle\psi|\mathcal{Q}|\psi\rangle$ when $|\psi\rangle = \alpha|0\rangle + \sqrt{1 - |\alpha|^2}|1\rangle$?

1.11 A qubit in an arbitrary pure quantum state is described mathematically by the state vector $|\psi\rangle = e^{i\gamma}(\cos(\frac{\theta}{2})|0\rangle + e^{i\phi} \sin(\frac{\theta}{2})|1\rangle)$. Equivalently, as you will see in Sect. 11.2.2 the *same* state can be described by the density operator $\rho = |\psi\rangle\langle\psi|$. On how many free parameters does the state $|\psi\rangle$ depend? Compute the density operator ρ corresponding to the state $|\psi\rangle$. On how many free parameters does the density operator ρ depend? Explain what role the parameter γ plays in the density operator representation of the state.

1.12 Look up the definition of the quantum Fourier transform (QFT) matrix defined in Sect. 3.4.6. Prove that the 1-qubit Walsh-Hadamard gate, $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, can be thought of as a 1-qubit quantum Fourier transform.

1.13 Find the unitary matrix that changes a state represented in the $\{|+\rangle, |-\rangle\}$ basis to one represented in the $\{|R\rangle, |L\rangle\}$ basis. You may assume $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, $|R\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$, and $|L\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$.

1.14 Find the unitary matrix that changes an arbitrary 2-qubit gate,

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ u_{21} & u_{22} & u_{23} & u_{24} \\ u_{31} & u_{32} & u_{33} & u_{34} \\ u_{41} & u_{42} & u_{43} & u_{44} \end{pmatrix},$$

in the $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ basis to one represented in the $\{|RR\rangle, |RL\rangle, |LR\rangle, |LL\rangle\}$ basis. You may assume $|R\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$, and $|L\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$, and $|RL\rangle = |R\rangle \otimes |L\rangle$ etc.

1.15 Consider a 2-qubit Hamiltonian having a block diagonal structure when expressed in the computational basis:

$$\mathcal{H} = \begin{pmatrix} a & e & 0 & 0 \\ e^* & b & 0 & 0 \\ 0 & 0 & c & g \\ 0 & 0 & g^* & d \end{pmatrix} \quad (1.56)$$

What are the eigenvalues and *normalized* eigenvectors of \mathcal{H} ?

1.16 Suppose we are promised that we are given either a known state $|\psi\rangle$ or a known state $|\varphi\rangle$ and we have to decide, by making some measurement, which is the case. If $|\psi\rangle$ and $|\varphi\rangle$ are non-orthogonal quantum states there is no single measurement that can distinguish between them 100% of the time. However, given knowledge of the forms for $|\psi\rangle$ and $|\varphi\rangle$ we can choose a measurement basis in which to measure our mystery state that optimizes our chances of guessing correctly. For example, consider the pair of quantum states defined by:

$$\begin{aligned} |\psi\rangle &= |0\rangle \\ |\varphi\rangle &= \alpha(\theta)|0\rangle + \beta(\theta)|1\rangle \end{aligned} \quad (1.57)$$

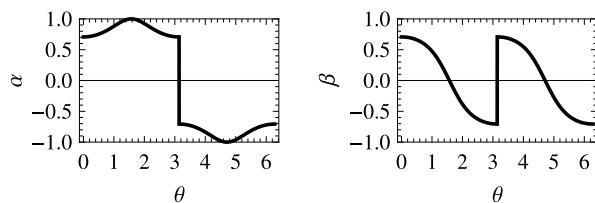
where

$$\begin{aligned} \alpha(\theta) &= \frac{\csc \theta}{\sqrt{|\csc \theta|^2 + |\cot \theta|^2}} \\ \beta(\theta) &= \frac{\cot \theta}{\sqrt{|\csc \theta|^2 + |\cot \theta|^2}} \end{aligned} \quad (1.58)$$

The amplitudes of the $|\varphi\rangle$ state are certainly peculiar, having the form over the interval $0 \leq \theta \leq 2\pi$ shown in Fig. 1.10.

- (a) Nevertheless, prove that $|\varphi\rangle$ is a properly normalized state.
- (b) With what probability can we guess correctly if we measure the mystery state in the computational, i.e., $\{|0\rangle, |1\rangle\}$, basis?
- (c) In what basis *ought* we to make the measurement to maximize our chances of guessing correctly whether we were given $|\psi\rangle$ and $|\varphi\rangle$?

Fig. 1.10 Amplitudes of the state $|\varphi\rangle = \alpha|0\rangle - \beta|1\rangle$ where $\alpha = \frac{\csc\theta}{\sqrt{|\csc\theta|^2 + |\cot\theta|^2}}$ and $\beta = \frac{\cot\theta}{\sqrt{|\csc\theta|^2 + |\cot\theta|^2}}$



- (d) What is the state $|\varphi\rangle$ at $\theta = \pi$? Is there any ambiguity?
- (e) What is the density operator corresponding to $|\varphi\rangle$, i.e., $\rho = |\varphi\rangle\langle\varphi|$?
- (f) What is the density operator ρ at $\theta = \pi$? Is there any ambiguity? How do you reconcile your answers to parts (d) and (f)?

Chapter 2

Quantum Gates

“When we get to the very, very small world—say circuits of seven atoms—we have a lot of new things that would happen that represent completely new opportunities for design. Atoms on a small scale behave like nothing on a large scale, for they satisfy the laws of quantum mechanics. So, as we go down and fiddle around with the atoms down there, we are working with different laws, and we can expect to do different things. We can manufacture in different ways. We can use, not just circuits, but some system involving the quantized energy levels, or the interactions of quantized spins.”

– Richard P. Feynman¹

Currently, the circuit model of a computer is the most useful abstraction of the computing process and is widely used in the computer industry in the design and construction of practical computing hardware. In the circuit model, computer scientists regard any computation as being equivalent to the action of a circuit built out of a handful of different types of Boolean logic gates acting on some binary (i.e., bit string) input. Each logic gate transforms its input bits into one or more output bits in some deterministic fashion according to the definition of the gate. By composing the gates in a graph such that the outputs from earlier gates feed into the inputs of later gates, computer scientists can prove that any feasible computation can be performed.

In this chapter we will look at the types of logic gates used within circuits and how the notions of logic gates need to be modified in the quantum context.

¹Source: Opening words of the “Atoms in a SmallWorld” section of Richard Feynman’s classic talk “There’s Plenty of Room at the Bottom,” given on 29th December 1959 at the annual meeting of the American Physical Society at the California Institute of Technology. The full transcript of the talk is available at <http://www.zyvex.com/nanotech/feynman.html>.

2.1 Classical Logic Gates

2.1.1 Boolean Functions and Combinational Logic

Logic is a sub-field of mathematics that is principally concerned with the validity of arguments, i.e., determining the truth or falsity of propositions by a process of reasoning from starting assumptions, called axioms, and by applying valid rules of inference to them. Logic is not concerned with determining what is actually true or false in the real world, since the real world is but one of infinitely many possible worlds we may choose to reason about. Rather logic provides the mathematical framework upon which we may draw valid conclusions from given starting assumptions.

The concept of a logic gate arose from efforts to formalize the laws of thought. George Boole (1815–1864) was a British mathematician who lived long before days of transistors and electronic digital computers. Like Babbage and von Leibnitz before him, Boole was interested in formalizing the process of mathematical reasoning. Before Boole, algebra had been thought about, primarily, as a vehicle for performing numerical calculations. However, Boole foresaw a wider opportunity: “[...] hitherto the expression of magnitude, or of operations upon magnitude, has been the express object for which the symbols of Analysis [algebra] have been invented, and for which their laws have been investigated, but this does not mean that the interpretations of algebra can only be quantitative”.

Boole went on to provide an interpretation of algebraic expressions as statements about classes of objects. The universe of all objects is a set, and symbols, such as A , B , C , stands for subsets of objects from this set. Then the usual operations on sets, such as intersection ($A \cap B$), union ($A \cup B$), and complement (A^c) can be interpreted as making statements about these subsets of objects as shown in Fig. 2.1.

For example, suppose we consider a universe of people with various pizza preferences. If A is the set people who like pepperoni, and B is the set of people who like anchovies, then $A \cap B$ is the set of people who like pepperoni *and* anchovies,

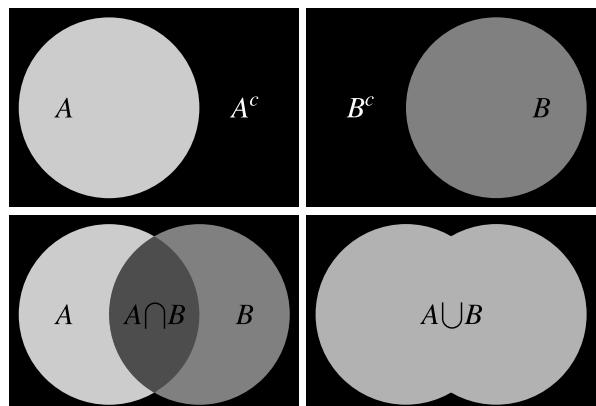


Fig. 2.1 Graphical illustration of the union, intersection and complement operations on sets

$A \cup B$ is the set of people who like pepperoni *or* anchovies or both, and A^c is the set of people who do *not* like pepperoni etc. Algebraic expressions interpreted in this way define what is called a Boolean *algebra*.

As you can see from the example, the *interpretation* of the sets that result from the intersection, union, and complement operations are described in terms of the logical connectives AND, OR, and NOT, indicating that there is a close parallel between set operations and logical operations. For example, if one assumes there are only two objects 1 = the set of all objects = TRUE and 0 = the empty set of objects = \emptyset = FALSE, we can write algebraic expressions that correctly capture alternate syntactic forms for logically equivalent statements. Hence, the *logical* assertion that a statement and its negation is necessarily contradictory expressed as the logical statement $a \wedge (\neg a) = 0 = \text{FALSE}$ (i.e., a AND (NOT a) is necessarily FALSE) mirrors the algebraic statement that the intersection of a set and its complement is necessarily empty, $A \cap A^c = \emptyset$. This restriction of the variables to just 0 and 1 makes the Boolean algebra into a Boolean *logic*.

Once one has the thought of interpreting algebraic statements as logical statements, one can easily define syntactically different forms having the same logical meaning. These are mathematical formulae in which the symbols, a, b, c, \dots stand for logical propositions that can be either true or false, and the connectives are logical functions. Table 2.1 lists the so-called “De Morgan’s Laws” which give syntactically equivalent versions of elementary logical propositions. By using these laws we can systematically eliminate from any logical expression all instances of \wedge or all instances of \vee . This means that we can reduce very complicated logical propositions to forms one of two standard forms, i.e., either a disjunction of conjuncts (i.e., Disjunctive Normal Form) or a conjunction of disjuncts (Conjunctive Normal Form).

Thus, if we can create hardware implementations of some very simple elementary gates, e.g., NOT, AND and OR, we can in principle combine those operations into very complex circuits

2.1.2 Irreversible Gates: AND and OR

The logical connectives AND (\wedge) and OR (\vee) capture, respectively, the notions of logical conjunction and disjunction. That is, for a compound proposition of the form $a \wedge b$ to be true *both* a and b must be true. Conversely, for a compound proposition of the form $a \vee b$ to be true it is sufficient for *either* a or b to be true individually.

Conventionally, a logic *gate* is thought of as a physical device that takes one or more Boolean values (i.e., FALSE or TRUE) as inputs and returns a single Boolean value as output. The Boolean values (FALSE and TRUE) are often used synonymously with the bit values 0 and 1 respectively. Logic gates are the key components of modern computers. Any classical computation can always be decomposed into a sequence of logic gates that act on only a few bits at a time. Hence logic gates lie at the heart of all modern computers.

Table 2.1 Logically equivalent propositions. Note by using De Morgan's laws any proposition can be expressed using NOT and AND alone or using NOT and OR alone

Logically equivalent forms

$a \wedge 0 = 0$	Zero of \wedge
$a \wedge 1 = a$	Identity of \wedge
$a \vee 0 = a$	Zero of \vee
$a \vee 1 = 1$	Identity of \vee
$a \wedge a = a$	Indempotence
$a \vee a = a$	Indempotence
$a \wedge \neg a = 0$	Law of Contradiction
$a \vee \neg a = 1$	Tautology
$\neg\neg a = a$	Double Negation
$a \wedge b = b \wedge a$	Commutativity of \wedge
$a \vee b = b \vee a$	Commutativity of \vee
$a \vee (b \vee c) = (a \vee b) \vee c$	Associativity
$a \wedge (b \wedge c) = (a \wedge b) \wedge c$	Associativity
$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$	Distributivity
$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$	Distributivity
$a \wedge (a \vee b) = a$	Absorption
$a \vee (a \wedge b) = a$	Absorption
$a \vee (\neg a \wedge b) = a \vee b$	Absorption
$a \wedge (\neg a \vee b) = a \wedge b$	Absorption
$\neg(a \wedge b) = (\neg a) \vee (\neg b)$	De Morgan's Law
$\neg(a \vee b) = (\neg a) \wedge (\neg b)$	De Morgan's Law
$(a \wedge b) \vee (a \wedge \neg b) = a$	
$a \implies b = \neg a \vee b$	
$a \implies b = \neg(a \wedge \neg b)$	

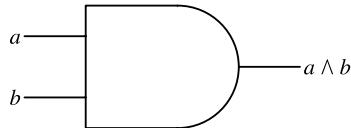
The best way to describe the action of a logic gate is in terms of its “truth table”. In a truth table we write down all the possible logical values of the inputs together with their corresponding outputs. For example, the truth table for the AND gate is given in Table 2.2. The corresponding icon for the AND gate as seen in circuit diagrams is shown in Fig. 2.2. The AND gate is logically irreversible, which means that you cannot determine unique inputs for all outputs. Specifically, if the output is 0 (i.e. FALSE), you cannot tell whether the input values were 00, 01, or 10. It “erases” some information when it acts whenever the output from the AND gate is 0.

Similarly, the truth table for the OR gate is shown in Table 2.3. The corresponding circuit icon for the OR gate is shown in Fig. 2.3. The OR gate is also logically irreversible because when its output is 1 (i.e., TRUE) it is impossible to say whether the inputs were 01, 10, or 11. Hence, again the OR gate erases some information when it acts whenever the output is a 1.

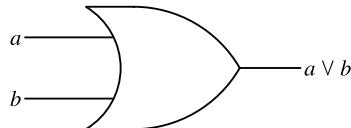
There is a variant of the OR gate, called exclusive-OR (often written “XOR” or “ \oplus ”) that turns out to be very useful. The XOR gate is like the OR gate except that

Table 2.2 Truth table of AND

	a	b	$a \wedge b$
AND:	0	0	0
	0	1	0
	1	0	0
	1	1	1

Fig. 2.2 Icon for the AND gate—a logically irreversible gate**Table 2.3** Truth table of OR

	a	b	$a \vee b$
OR:	0	0	0
	0	1	1
	1	0	1
	1	1	1

Fig. 2.3 Icon for the OR gate—a logically irreversible gate**Table 2.4** Truth table of XOR (exclusive-OR)

	a	b	$a \oplus b$
XOR:	0	0	0
	0	1	1
	1	0	1
	1	1	0

it returns 0 (i.e., FALSE) when both its inputs are 1 (i.e., TRUE). The truth table for XOR is shown in Table 2.4. The corresponding circuit icon for XOR is shown in Fig. 2.4.

2.1.3 Universal Gates: NAND and NOR

There is a special class of logic gates, called *universal* gates, any one of which is alone sufficient to express any desired computation. The possibility of such uni-

Fig. 2.4 Icon for the XOR gate—a logically irreversible gate

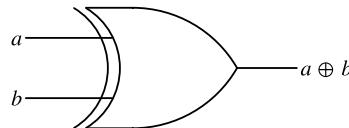
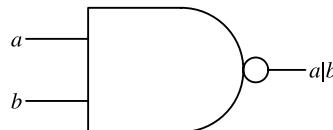


Table 2.5 Truth table of NAND

NAND:	a	b	a b
	0	0	1
	0	1	1
	1	0	1
	1	1	0

Fig. 2.5 Icon for the NAND gate—a universal gate for classical irreversible computing



versal gates accounts, in part, for the remarkable miniaturization of modern computers since computer designers need only focus on miniaturizing a single type of gate. Nowadays, the logic gates that manipulate these values are implemented using transistors, but in future computers even smaller, and faster, devices are being considered in an effort to maintain the pace of Moore’s Law.

You can see why such universal gates are possible from Table 2.1. The rules in the table show that any Boolean function can be reduced to an expression involving only \neg and \wedge or only \neg and \vee . Hence, any Boolean function can be computed by means of a circuit comprising NOT and AND gates, or NOT and OR gates. Nevertheless, the construction of large scale logic circuits would be greatly streamlined if manufacturers only had to use a *single* type of gate. Such a gate is said to be “universal” since from it circuits for any Boolean function can be derived. Restricting circuits to using a single type of universal gate does not necessarily lead to the smallest circuit for computing a desired Boolean function but it does allow chip manufacturers to perfect the design and manufacturing process for the universal gate, which, in practice, tends to make it easier to improve yield, reliability, and boost speed. Today, the microprocessor industry pursues this strategy by basing their circuits on the NAND (“NOT AND”) gates. Mathematically, $a \text{NAND } b \equiv \neg(a \wedge b)$, often written as $a|b$, and is universal for classical irreversible computing. The truth table for the NAND gate is shown in Table 2.5: The corresponding circuit icon for the NAND gate is shown in Fig. 2.5.

To convince you that the NAND gate is truly universal, given that we already know we can compute any Boolean function in a circuit comprising only NOT and AND gates, it is sufficient to show we can obtain NOT from NAND gates and AND from NAND gates. Table 2.6 shows how to obtain $\neg a$ from $a|a$: Likewise, Table 2.7 shows we can obtain $a \wedge b$ from two $a|b$ gates. Since we proved that any logical

Table 2.6 A NOT gate can be obtained using a NAND gate since $a|a$ has precisely the same truth values as $\neg a$

a	a	$a a$	$\neg a$
NOT in terms of NAND:			
0	0	1	1
1	1	0	0

Table 2.7 An AND gate can be obtained using only NAND gates since $a \wedge b$ has precisely the same truth values as $(a|b)|(a|b)$

a	b	$a b$	$(a b) (a b)$	$a \wedge b$
AND in terms of NAND:				
0	0	1	0	0
0	1	1	0	0
1	0	1	0	0
1	1	0	1	1

proposition can be written in terms of only \neg and \wedge , and that \neg and \wedge can, in turn, each be written in terms of $|$ (NAND) we have proved that any logical proposition can be written only in terms of $|$ (NAND) gates. This is good news for chip manufacturers because it means they need only perfect the implementation of just one type of gate, the NAND gate, to be sure that they can build a circuit that can perform any feasible computation.

There are other universal gates for classical irreversible computing including the NOR gate (“NOT OR”) and the NMAJORITY gate (“NOT MAJORITY”). The NMAJORITY gate is a relatively new universal gate. It is especially interesting because it is implementable in a new transistor design and leads to highly compact circuits.

Unfortunately, logical irreversibility comes at a price. Fundamental physics dictates that energy must be dissipated when information is erased, in the amount $kT \ln 2$ per bit erased, where k is Boltzman’s constant ($k = 1.3805 \times 10^{-23} \text{ JK}^{-1}$) and T is the absolute temperature (in degrees Kelvin). Thus, even if all other energy loss mechanisms were eliminated from any NAND based circuit, the circuit would still dissipate energy when it operated due to the unavoidable energy losses that occur when information is erased.

Today energy losses in NAND-based logic circuits due to logical irreversibility are dwarfed by other loss mechanisms. However, as these other loss mechanisms are tamed, someday the energy losses due solely to information erasure (in turn a consequence of using irreversible logic gates) will become the significant contribution. At this point if nothing is done, further miniaturization of computer technology will be impeded by the difficulty of removing this unwanted waste heat from deep within the irreversible circuitry.

2.1.4 Reversible Gates: NOT, SWAP, and CNOT

One way chip manufacturers can suppress the unwanted heat produced as a side effect of running irreversible logic gates is to modify their chip designs to use only

reversible logic gates. In a reversible logic gate there is always a unique input associated with a unique output and vice versa. So reversible gates never erase any information when they act, and consequently, a computation based on reversible logic can be run forward to obtain an answer, the answer copied, and then the whole computation undone to recover all the energy expended apart from the small amount used to copy the answer at the mid-way point.

The simplest example of a reversible logic gate is the NOT gate. NOT is a 1-input/1-output gate that simply inverts the bit value it is handed. The truth table for the NOT gate is shown in Table 2.8. The circuit icon for the NOT gate is shown in Fig. 2.6. If one knows the output bit value, one can infer the input bit value unambiguously and vice versa.

A slightly more complicated example, is the 2-input/2-output SWAP gate. SWAP simply exchanges the bit values it is handed. Its truth table is shown in Table 2.9. The circuit icon for the SWAP gate is shown in Fig. 2.7. In quantum computing a circuit may not have any physical wires connecting the gates together. Instead a circuit can be merely a visual specification of a sequence of gate operations with time increasing from left to right in the circuit diagram as successive gates are applied. Consequently, in quantum computing we sometimes use a different icon for a SWAP gate (showing in Fig. 2.8, that is more suggestive that some operation (other than crossing wires) needs to occur to achieve the effect of a SWAP operation.

A reversible gate of considerable importance in quantum computing is the 2-bit controlled-NOT gate (CNOT). The truth table for CNOT is shown in Table 2.10. The circuit icon for the CNOT gate is shown in Fig. 2.9. The effect of the “controlled”-NOT gate is to flip the bit value of the second bit if and only if the first bit is set to 1.

Table 2.8 Truth table of NOT

NOT:	a	$\neg a$
0	0	1
1	1	0

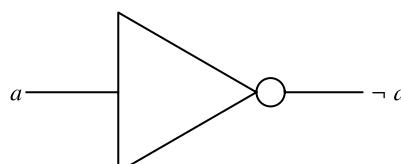


Fig. 2.6 Icon for the XOR gate—a 1-bit logically reversible gate

Table 2.9 Truth table of SWAP

SWAP:	a	b	a'	b'
0	0	0	0	0
0	0	1	1	0
1	0	0	0	1
1	0	1	1	1

Fig. 2.7 Icon for the SWAP gate—a 2-bit logically reversible gate. The icon conveys the idea that to swap two bits we simply cross the wires on which those bits reside

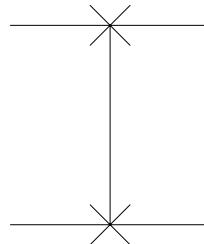
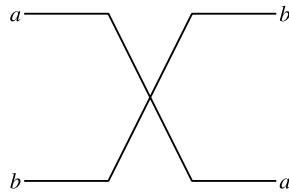
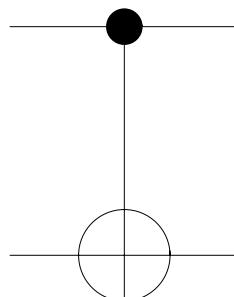


Fig. 2.8 Alternative icon for a SWAP gate that is more common in quantum circuit diagrams. The reason for having a different icon for SWAP in quantum circuits compared to classical circuits is that many implementations of quantum circuits do not have physical wires as such. Hence, it could be misleading to depict a SWAP operation as a crossing of wires. Instead, a SWAP operation can be achieved as the result of a sequence of applied fields

Table 2.10 Truth table of CNOT

	a	b	a'	b'
CNOT:	0	0	0	0
	0	1	0	1
	1	0	1	1
	1	1	1	0

Fig. 2.9 Icon for the CNOT gate—a 2-bit logically reversible gate



That is, the decision to negate or not negate the second bit is controlled by the value of the first bit. Hence, the name “controlled-NOT”. Note that, as shown in Fig. 2.10, the SWAP gate can be obtained from a sequence of three CNOT gates.

Fig. 2.10 A SWAP gate can be obtained from three CNOT gates

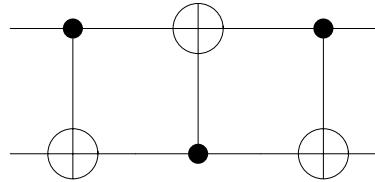


Table 2.11 Truth table of the TOFFOLI gate, which is universal for classical reversible computing

	a	b	c	a'	b'	c'
TOFFOLI:	0	0	0	0	0	0
	0	0	1	0	0	1
	0	1	0	0	1	0
	0	1	1	0	1	1
	1	0	0	1	0	0
	1	0	1	1	0	1
	1	1	0	1	1	1
	1	1	1	1	1	0

Table 2.12 Truth table of the FREDKIN gate, which is universal for classical reversible computing

	a	b	c	a'	b'	c'
FREDKIN:	0	0	0	0	0	0
	0	0	1	0	0	1
	0	1	0	0	1	0
	0	1	1	0	1	1
	1	0	0	1	0	0
	1	0	1	1	1	0
	1	1	0	1	0	1
	1	1	1	1	1	1

2.1.5 Universal Reversible Gates: FREDKIN and TOFFOLI

Just as there can be universal gates for classical irreversible computing, such as the NAND gate (which has two inputs and one output), so too can there be universal gates for classical *reversible* computing. However, the smallest gates that are both reversible *and* universal require *three* inputs and *three* outputs. Two well-known examples are the FREDKIN (controlled-SWAP) gate and the TOFFOLI (controlled-CNOT) gate, whose truth tables are shown in Tables 2.11 and 2.12 respectively.

Fig. 2.11 Icon for the TOFFOLI gate also called the controlled-controlled-NOT gate. TOFFOLI is reversible and universal

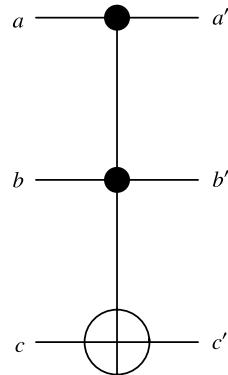
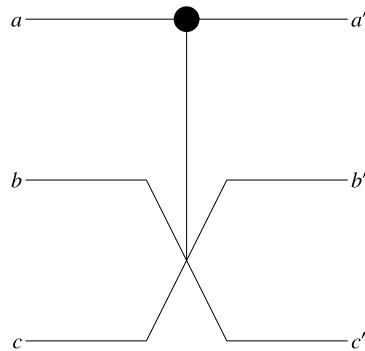


Fig. 2.12 Icon for the FREDKIN gate also called the controlled-SWAP gate. FREDKIN is reversible and universal



2.1.5.1 TOFFOLI (a.k.a. “Controlled-Controlled-NOT”)

The TOFFOLI gate is also called the controlled-controlled-NOT gate since it can be understood as flipping the third input bit if, and only if, the first two input bits are both 1. In other words, the values of the first two input bits control whether the third input bit is flipped. The icon for the TOFFOLI gate is shown in Fig. 2.11.

2.1.5.2 FREDKIN (a.k.a. “Controlled-SWAP”)

Another famous reversible gate is the FREDKIN (controlled-SWAP) gate. The truth table for the FREDKIN gate is: The icon for the FREDKIN gate is shown in Fig. 2.12. The FREDKIN gate can also be seen as a controlled-SWAP gate in that it swaps the values of the second and third bits, if, and only if, the first bit is set to 1.

2.1.6 Reversible Gates Expressed as Permutation Matrices

Any n -bit reversible gate must specify how to map each distinct bit string input into a distinct bit string output of the same length. Thus no two inputs are allowed to be

mapped to the same output and vice versa. This ensures the mapping is reversible. Consequently, one can think of a reversible gate as encoding a specification for how to permute the 2^n possible bit strings inputs expressible in n bits. In the case of the 2-bit SWAP gate, for example, the four possible input bit strings are 00, 01, 10, 11 and these are mapped, respectively, into $00 \rightarrow 00$, $01 \rightarrow 10$, $10 \rightarrow 01$, $11 \rightarrow 11$. In the case of CNOT gate, the inputs 00, 01, 10, and 11 are mapped into 00, 01, 11, and 10 respectively. Thus a natural way to represent an n -bit reversible gate is as an array whose rows and columns are indexed by the 2^n possible bit strings expressible in n bits. The (i, j) -th element of this array is defined to be 1 if, and only if, the input bit string corresponding to the i -th row is mapped to the output bit string corresponding to the j -th column. The resulting array will contain a single 1 in each row and column and zeroes everywhere else, and will therefore be a permutation matrix. As arrays, the NOT, SWAP and CNOT gates would be described as follows:

$$\text{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}; \quad \text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \quad (2.1)$$

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Likewise, the TOFFOLI gate could be represented as:

$$\text{TOFFOLI: } \begin{array}{ccccccccc} & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ \begin{matrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{matrix} & \left(\begin{array}{ccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right) \end{array} \quad (2.2)$$

Similarly, the action of the FREDKIN gate could be represented as:

$$\text{FREDKIN: } \begin{array}{ccccccccc} & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ \begin{matrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{matrix} & \left(\begin{array}{ccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array} \right) \end{array} \quad (2.3)$$

In fact, the matrices corresponding to classical reversible gates are always permutation matrices, i.e., 0/1 matrices having a single 1 in each row and column, and permutation matrices are also always unitary matrices.

To calculate the effect of a reversible gate, e.g., the FREDKIN or TOFFOLI gate, on an input bit string, we simply prepare the column vector corresponding to that bit string, and then perform the usual matrix vector product operation. For example, since the FREDKIN and TOFFOLI gates act on three bits, we can imagine a column vector consisting of $2^3 = 8$ slots, one of which (the i -th say) contains a single 1, and all the other elements are 0.

$$000 \equiv \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad 001 \equiv \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad 010 \equiv \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \dots \quad 111 \equiv \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (2.4)$$

etc. We can calculate the effect of, e.g., the TOFFOLI gate on such an input by vector-matrix multiplication.

$$\text{TOFFOLI}|110\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |111\rangle \quad (2.5)$$

2.1.7 Will Future Classical Computers Be Reversible?

The computer industry has done a truly remarkable job at squeezing more and more computation out of fewer and fewer physical resources. For example, the energy per logical operation has decreased pretty much exponentially since the inception of the microchip, in lock step with a similar reduction in the size of transistors. As a result a given *volume* of microprocessor has, over successive generations, been made to perform exponentially more computation.

However, chip designers are now finding it harder to increase performance without incurring the need to dissipate more energy per unit area of chip. You can sense this quite directly if you spend any time working with a notebook computer on your lap. After a while you will notice it becoming quite warm. This is because the microprocessor is dissipating heat as it runs. Indeed, modern chips can consume 100

Watts or more. Since it is impractical to allow them to dissipate more power than this, this problem could ultimately stall Moore’s Law.

Today, power losses arise from the non-zero electrical resistance of the conductors used inside microprocessors and some leakage of current through materials that are supposed to be insulators. This chip designers are working feverishly to lessen such losses by using fewer and fewer electrons and avoiding large voltage swings, which cuts down leakage. Once these stratagems have been played out to the maximum extent possible chip designers will have to consider various methods, such as charge recovery, to recapture energy, much like a flywheel recaptures energy in mechanical devices. Beyond this, what options remain to further reduce energy dissipation during computation?

The answer could lie in the use of classical reversible gates, such as FREDKIN and TOFFOLI gates that we discussed earlier. This is because, as Rolf Landauer showed, energy need only be dissipated when information is erased, and the minimum amount that Nature demands is $k_B T \ln 2$ per bit erased, where k_B is Boltzmann’s constant and T is the temperature in degrees Kelvin. At room temperature (300 Kelvin) this is about 3×10^{-21} Joules per bit erased. Therefore, if we were to use reversible computing, the only energy that must be dissipated is related to that required to initialize the computer, or to make a permanent record on an answer, because these operations must take a memory register in one state, and reset it, regardless of what that state was, in a fixed configuration. Hence this operation is necessarily irreversible. But apart from that, in principle, it takes no energy to compute!

2.1.8 Cost of Simulating Irreversible Computations Reversibly

Today, most computing hardware employs, at its lowest level, gates that are logically irreversible. Logical irreversibility means that certain outputs from a logic gate are consistent with more than one set of inputs, preventing one from inferring a unique input for each output. For example, the logic gate $\text{AND}(x, y) = z$ that maps two input bits, x and y , into a single bit, z , is logically irreversible because an output $z = 0$ (false) could be accounted for by any of the three input pairs $(x = 0, y = 0)$, $(x = 0, y = 1)$ and $(x = 1, y = 0)$. Hence, for this particular output, the input is ambiguous and the operation is therefore logically irreversible.

It has long been known that such logical irreversibility has a thermodynamic consequence, namely, that energy must be dissipated, in the amount $k_B T \log 2$ per bit erased, whenever a logically irreversible operation is performed [299]. However, the converse of this is also true. If we were to employ only *logically reversible* gates inside our chips, then no net energy need be dissipated in performing those gate operations. The only thermodynamic cost to computing would then be the cost of creating the initial input, reading the output, and re-setting the computer.

For a computation to be logically reversibility each “step” of the computation must be logically reversible. However, the exact meaning of a “step” changes de-

pending on the model of computation being used. For example, in the Turing machine model one step of computation is a transition of the finite control of the machine [44], which maps one “configuration” of the machine to another configuration. Likewise, in the circuit model, a step of computation is the execution of one gate of the circuit (see, e.g., [187, 494]). Thus, a *reversible Turing machine* is a machine mapping distinct input configurations to distinct output configurations, and a *reversible circuit* is a circuit comprised of gates each mapping distinct input bit patterns to distinct output bit patterns.

There are two important questions concerning reversible computing. The first is the practical question of how to find the optimal reversible circuit implementing a desired Boolean function [343, 451, 494]. This approach boils down to understanding how to implement permutations by reversible circuits, and is mainly concerned with generic functions.

The second question concerning reversible computing is to determine with what efficiency a reversible computer can simulate an irreversible computation [44, 45, 88, 119, 302, 311, 312]. Most previous studies of this question have addressed it in the context of the Turing machine model of computation. In this paper we present a similar analysis in the context of the circuit model. In order to aid comparison we first recap the insights gleaned from these Turning machine studies.

Initially it was believed that the only way to simulate an irreversible computation on a reversible Turing machine was to keep all the intermediate calculations. Consequently, the size of the memory (i.e., “space”) needed to perform the computation reversibly was proportional to the time (i.e., number of steps) of the corresponding irreversible computation. Bennett, however, [44] discovered that the history of a reversible computation could be cleared in a reversible fashion, leaving only the input and the output in memory, and recording the configuration of certain *checkpoints* of the irreversible computation. This reduced the space needed to simulate an irreversible computation reversibly but at the expense of increasing the time of the reversible computation. Specifically, in [45] Bennett proposed a method which uses time $S T^{\log 3}$ and space $S \log T$, when the irreversible computation uses T time and S space. In this case the space complexity of the simulation is S^2 in the worst case. Later it was shown that it is possible to have a reversible simulation in space $O(S)$ but at the cost of requiring the simulation to run in exponential time [302]. The best tradeoff for reversible simulation of an irreversible computation was provided by Li [312]. It uses time $\Theta(T^{1+\varepsilon}/S^\varepsilon)$ and space $\Theta(c(\varepsilon)S[1 + \log(T/S)])$, for any $\varepsilon > 0$, where $c(\varepsilon) \approx \varepsilon 2^{1/\varepsilon}$. Similarly, in [119] it is shown that any *nondeterministic* Turing machine running in space S can be simulated by a reversible machine using space $O(S^2)$.

The foregoing studies of the efficiency with which a reversible computer can simulate an irreversible computation were all based on the deterministic or non-deterministic Turing machine models. As best we can tell there has been no similar direct study in the literature based on the circuit model of computation. This is the main contribution of our paper.

Toffoli and Fredkin [187, 494] performed some of the first systematic studies of reversible circuits. Toffoli showed, for example, that the reversible basis consisting

of NOT, CNOT , and Toffoli gates (defined in Sect. 2.2) is *universal* for reversible computation. More precisely, he showed that every permutation on $\{0, 1\}^n$ can be realized by means of a reversible circuit over the NOT-CNOT-TOFFOLI basis using at most one ancilla bit.²

2.1.9 Ancillae in Reversible Computing

Ancillae are an essential ingredient in classical reversible computing. For example, every circuit with more than 3 inputs over the NOT-CNOT-TOFFOLI basis realizes an *even* permutation on the space of its inputs. Therefore, to realize an *odd* permutation on $\{0, 1\}^n$, we need at least one ancilla bit with fixed constant value in addition to the n variable inputs. Toffoli has shown that one ancilla bit is, in fact, always sufficient [451]. Another way to see ancillae are essential is to consider computing a Boolean function $f : \{0, 1\}^n \longrightarrow \{0, 1\}$ reversibly. Every reversible circuit on m inputs, computing f , has exactly m outputs with one of them considered the value of f . If $m = n$, i.e., there is no ancilla bit, then it is easy to see that every output function must be a *balanced* Boolean function.³ Therefore, if the function we want to simulate is not balanced, we require $m > n$ and there must therefore be at least one ancilla bit.

In general, we use the model described in Fig. 2.13 to define how a reversible circuit computes a function $f : \{0, 1\}^n \longrightarrow \{0, 1\}$. In this model, it is required that at the end of the computation all ancillae have their initial values, except one ancilla bit, designated as the “answer” bit, that carries the value of the function.

As in the case of reversible Turing machines, we can trade space for time in reversible *circuit* simulations of irreversible computations. But in the circuit picture “space” (i.e., the amount of auxiliary memory) is measured in terms of the *number of ancillae* required to perform the computation, and “time” is measured by the *size*, i.e. total gate count, of the circuit. In some cases allowing more ancillae results in a reversible circuit with smaller net size (i.e., fewer total gates).

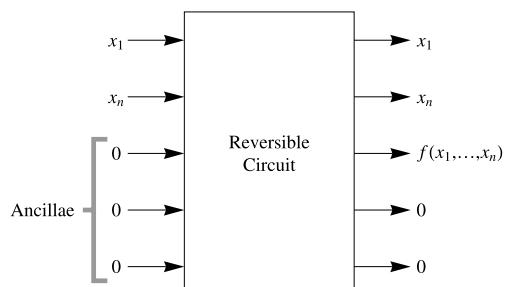


Fig. 2.13 Computing a Boolean function using a reversible circuit

²What we call an “ancilla bit” is also referred to as a “storage bit” or a “garbage bit” in the literature.

³A balanced function on $\{0, 1\}^n$ returns a value “1” for 2^{n-1} of its inputs and a value “0” for the other 2^{n-1} inputs.

To the best of my knowledge, only Cleve [110, 111] has addressed the space-time (ancillae-size) trade-off of simulation for the reversible circuits. He has shown that any polynomial size *formula* can be simulated by a polynomial size reversible circuit, which uses only 3 ancillae. If his method is applied to a *circuit*, then the result is an exponential size reversible circuit with 3 ancillae.

In contrast, we provide two new methods for simulating general Boolean circuits. In the first method, we show that any irreversible computation having t gates, depth d , and width w , can be implemented in a reversible circuit having $O(t^{2.58})$ gates, and at most $(w+1)\log d + O(1)$ ancillae. The second method deals with the simulation of branching programs. We prove that any branching program of depth d and width w can be simulated by a reversible circuit of size $\leq 4w2^d$ with $2w$ ancillae.

2.2 Universal Reversible Basis

We consider reversible circuits over the NOT-CNOT-TOFFOLI basis. Table 2.13 defines the action of these gates, and the Fig. 2.14 represents their standard icons. Note that the TOFFOLI gate *alone* is universal for reversible computing so, in principle, we do not need the NOT and CNOT gates. However, we allow them to simplify the constructions. Figure 2.15 shows how these reversible gates can simulate the classical (irreversible) standard gates, in some cases with ancillae.

Table 2.13 The action of reversible gates

NOT	CNOT	TOFFOLI
$a \mapsto 1 \oplus a$	$\begin{pmatrix} a \\ b \end{pmatrix} \mapsto \begin{pmatrix} a \\ a \oplus b \end{pmatrix}$	$\begin{pmatrix} a \\ b \\ c \end{pmatrix} \mapsto \begin{pmatrix} a \\ b \\ c \oplus (a \cdot b) \end{pmatrix}$

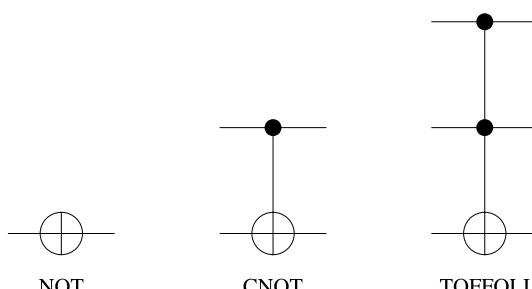


Fig. 2.14 The reversible basis

Fig. 2.15 Reversible simulation of classical gates

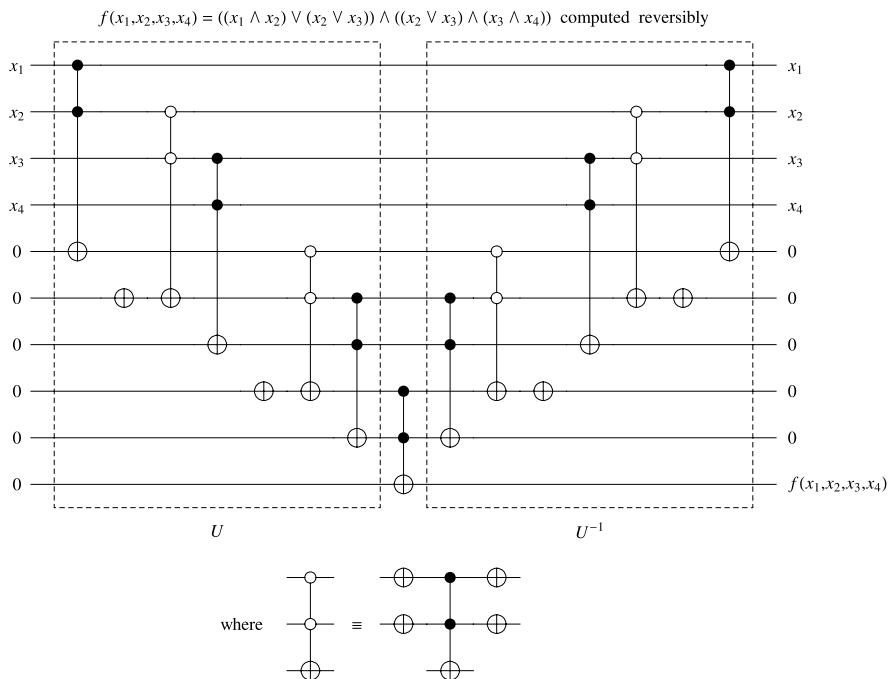
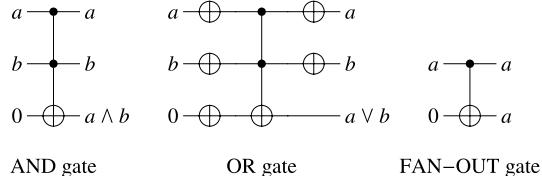


Fig. 2.16 Synthesis via reversible substitution

2.2.1 Can All Boolean Circuits Be Simulated Reversibly?

The constructions of Fig. 2.15 suggest a simple (naive) method for simulating any Boolean (irreversible) circuit: simply replace each irreversible gate in the circuit with its reversible counterpart. Figure 2.16 shows an example of this method.

However, this naive method is hardly efficient and we now present a better scheme. Before we begin, we define some useful terminology. A synchronous circuit is one in which all paths from the inputs to any gate have the same length. Synchronous circuits may have delay (identity) gates, and gates at level m get inputs from gates at level $m - 1$. Thus, without loss of generality, we can assume that our desired irreversible circuit is synchronous. For a Boolean circuit, the *size* is the total number of gates, the *depth* is the number of levels, and the *width* is the maximum number of gates in any level.

The following procedure shows how to create a reversible circuit that simulates and irreversible circuit while making substantial savings in the number of ancillae used.

- First simulate the gates in the first-half levels.
- Keep the results of the gates in the level $d/2$ separately.
- Clean up the ancillae bits.
- Use them to simulate the gates in the second-half levels.
- After computing the output, clean up the ancillae bits.
- Clean up the result of the level $d/2$.

Note This method needs roughly *half* the number of ancillae used by the previous (naive) method. Figure 2.16 shows the circuit of this procedure.

By applying the above procedure recursively, on a circuit of size t , depth d , and width w we obtain the following recursive relations for S , the size, and A , the number of the ancillae needed:

$$S(t) \leq 6S(t/2) + O(1),$$

$$A(d) \leq A(d/2) + w + 1.$$

Solving these recursion relations leads to the following result.

Efficiency of Reversible Simulation *Any irreversible computation (in the synchronous form) having t gates, depth d , and width w , can be simulated by a reversible circuit having $O(t^{2.58})$ gates, and at most $(w + 1) \log d + O(1)$ ancillae.*

Thus, most of the irreversible computations going on inside your notebook computer could, in principle, be implemented using reversible logic gates, which in turn need no *net* energy to run apart from any operations that require erasure of information, such as overwriting a memory register to make a copy of an answer! This is surprise to many people because their perception is that computers are making something new. But in reality, they don't. They just take the known information given as input and re-arrange it. The vast majority of the operations employed along the way can be done reversibly, and hence, don't generate any more information in their output than they had in their input. There is no truly creative act as such. As Pablo Picasso once said, “Computers are useless—they only give answers!”

2.3 Quantum Logic Gates

Now that we have looked at classical irreversible and classical reversible gates, we have a better context in which to appreciate the benefits of quantum gates.

Just as any classical computation can be broken down into a sequence of classical logic gates that act on only a few classical bits at a time, so too can any quantum computation can be broken down into a sequence of quantum logic gates that act on

only a few qubits at a time. The main difference is that whereas classical logic gates manipulate the classical bit values, 0 or 1, quantum gates can manipulate arbitrary multi-partite quantum states including arbitrary superpositions of the computational basis states, which are frequently also entangled. Thus the logic gates of quantum computation are considerably more varied than the logic gates of classical computation.

2.3.1 From Quantum Dynamics to Quantum Gates

The physical phenomena used to achieve the desired manipulation of a quantum state can be very varied. For example, if qubits are encoded in particles having quantum mechanical spin, the logic is effected by spin-manipulation brought about by varying an applied magnetic field at various orientations. Or if the qubit is encoded in an internal excitation state of an ion, the gate operation can be achieved by varying the time a laser beam is allowed to irradiate the ion or by varying the wavelength of that laser light.

As any quantum gate must be implemented physically as the quantum mechanical evolution of an isolated quantum system, the transformation it achieves is governed by Schrödinger's equation, $i\hbar\partial|\psi\rangle/\partial t = \mathcal{H}|\psi\rangle$, where \mathcal{H} is the Hamiltonian, specifying the physical fields and forces at work. Thus, the unitary matrices describing quantum gates are related to the physical processes by which they are achieved via the equation $U = \exp(-i\mathcal{H}t/\hbar)$. Here \mathcal{H} is the Hamiltonian which specifies the interactions that are present in the physical system.

As we saw in Chap. 1, the quantum mechanical evolution induced by this equation is unitary provided no measurements are made, and no unwanted stray interactions occur with the environment. In this case, starting from some initial state, $|\psi(0)\rangle$, the quantum system will evolve, in time t , into the state $|\psi(t)\rangle = \exp(-i\mathcal{H}t/\hbar)|\psi(0)\rangle = U|\psi(0)\rangle$ where U is some unitary matrix. Thus the evolution, in time t , of an isolated quantum system is described by a unitary transformation of an initial state $|\psi(0)\rangle$ to a final state $|\psi(t)\rangle = U|\psi(0)\rangle$. This means that a quantum logic gate acting on an isolated quantum computer, will transform that state unitarily up until the point at which an observation is made. Hence, quantum logic gates are described, mathematically, by unitary matrices, and their action is always logically reversible.

The parallels between classical reversible gates and quantum gate were not lost the early quantum computer pioneers Richard Feynman and David Deutsch. They recognized that since the matrices corresponding to reversible (classical) gates were permutation matrices, they were also unitary matrices and hence could be interpreted as operators that evolved some initial quantum state representing the input to a gate into some final quantum state representing its output in accordance with Schrödinger's equation. Thus, the closest classical analogs to quantum logic gates are the classical reversible gates such as the NOT, SWAP, CNOT, TOFFOLI and FREDKIN. However, whereas the repertoire of gates available in classical reversible

computing is limited to the unitary gates whose matrix representations correspond to permutation matrices, in deterministic quantum computing any gate is allowed whose matrix is unitary whether or not it is also a permutation matrix.

2.3.2 Properties of Quantum Gates Arising from Unitarity

The essential properties of quantum logic gates flow immediately from that fact that they are described by unitary matrices. A matrix, U , is unitary if and only if its inverse⁴ equals its conjugate transpose, i.e., if and only if $U^{-1} = U^\dagger$. If U is unitary the following facts hold:

- U^\dagger is unitary.
- U^{-1} is unitary.
- $U^{-1} = U^\dagger$ (which is the criterion for determining unitarity).
- $U^\dagger U = \mathbb{1}$
- $|\det(U)| = 1$.
- The columns (rows) of U form an orthonormal set of vectors.
- For a fixed column, $\sum_{i=1}^{2^n} |U_{ij}|^2 = 1$.
- For a fixed row, $\sum_{j=1}^{2^n} |U_{ij}|^2 = 1$.
- $U = \exp(i\mathcal{H})$ where \mathcal{H} is an hermitian matrix, i.e., $\mathcal{H} = \mathcal{H}^\dagger$.

The fact that, for any quantum gate U , $U^\dagger U = \mathbb{1}$ ensures that we can always undo a quantum gate, i.e., that a quantum gate is logically reversible. Moreover, that fact that for a fixed column $\sum_{i=1}^{2^n} |U_{ij}|^2 = 1$ and for a fixed row $\sum_{j=1}^{2^n} |U_{ij}|^2 = 1$ guarantee that if you start with a properly normalized quantum state and act upon it with a quantum gate, then you will end up with a properly normalized quantum state. Thus, there are no probability “leaks”. The fact that it is the magnitude $|\det(U)|$ that is constrained to be unity means that the constraint on the determinant can be satisfied with $\det(U) = \pm 1$ or $\pm i$. Thus the elements of a general unitary matrix are generically allowed to be complex numbers.

2.4 1-Qubit Gates

2.4.1 Special 1-Qubit Gates

2.4.1.1 Pauli Spin Matrices

For single qubits, the “Pauli matrices” ($\mathbb{1}, X, Y, Z$), which happen to be both hermitian and unitary, are of special interest since any 1-qubit Hamiltonian can always be

⁴If A and B are two matrices B is the inverse of A when $A \cdot B = \mathbb{1}$ where $\mathbb{1}$ is the identity matrix, i.e., a matrix having only ones down the main diagonal.

written as a weighted sum of the Pauli matrices:

$$\mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (2.6)$$

Some common forms for Hamiltonians that arise in practice are $\mathcal{H} = Z^{(1)}Z^{(2)}$ (the Ising interaction) and $\mathcal{H} = X^{(1)} \otimes X^{(2)} + Y^{(1)} \otimes Y^{(2)}$ (the XY interaction) and $\mathcal{H} = 2X^{(1)} \otimes X^{(2)} + Y^{(1)} \otimes Y^{(2)}$ where the parenthetical superscripts labels which of two qubits the operator acts upon.

2.4.1.2 NOT Gate

The Pauli X matrix is synonymous with the classical (reversible) NOT gate, i.e.,

$$X \equiv \text{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (2.7)$$

Thus, it is not surprising that X negates the computational basis states $|0\rangle$ and $|1\rangle$, correctly as these correspond to the classical bits, 0 and 1, respectively. Specifically, we have:

$$X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \quad (2.8)$$

$$X|1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \quad (2.9)$$

2.4.1.3 $\sqrt{\text{NOT}}$ Gate

One of the simplest 1-qubit non-classical gates one can imagine is a fractional power of the NOT gate, such as $\sqrt{\text{NOT}}$:

$$\sqrt{\text{NOT}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^{\frac{1}{2}} = \begin{pmatrix} \frac{1}{2} + \frac{i}{2} & \frac{1}{2} - \frac{i}{2} \\ \frac{1}{2} - \frac{i}{2} & \frac{1}{2} + \frac{i}{2} \end{pmatrix} \quad (2.10)$$

The $\sqrt{\text{NOT}}$ gate has the property that a repeated application of the gate, i.e., $\sqrt{\text{NOT}} \cdot \sqrt{\text{NOT}}$, is equivalent to the NOT operation, but a single application results in a quantum state that neither corresponds to the classical bit 0, or the classical bit 1. So $\sqrt{\text{NOT}}$ it is the first truly non-classical gate we have encountered.

$$|0\rangle \xrightarrow{\sqrt{\text{NOT}}} \left(\frac{1}{2} + \frac{i}{2}\right)|0\rangle + \left(\frac{1}{2} - \frac{i}{2}\right)|1\rangle \xrightarrow{\sqrt{\text{NOT}}} |1\rangle \quad (2.11)$$

$$|1\rangle \xrightarrow{\sqrt{\text{NOT}}} \left(\frac{1}{2} - \frac{i}{2}\right)|0\rangle + \left(\frac{1}{2} + \frac{i}{2}\right)|1\rangle \xrightarrow{\sqrt{\text{NOT}}} |0\rangle \quad (2.12)$$

2.4.1.4 Is Pauli X a NOT Gate for Qubits?

Although the Pauli X gate negates the computational basis states correctly, does it also behave like a true “NOT” gate when acting on a qubit in an *arbitrary* quantum state, i.e., a qubit state corresponding to a point on the Bloch sphere other than the North or South poles? To answer this, we must first specify what we *require* a quantum NOT gate to do, and then determine whether X acts in the appropriate manner.

Since the NOT gate has the effect of mapping a state at the North pole of the Bloch sphere into a state at the South pole and vice versa, it is natural to extend the definition of a NOT gate to be the operation that maps a qubit, $|\psi\rangle$, lying at any point on the surface of the Bloch sphere, into its antipodal state, $|\psi^\perp\rangle$, on the opposite side of the Bloch sphere as shown in Fig. 2.17. The antipodal point is that obtained by projecting a straight line from the original state through the origin to intersect the surface of the Bloch sphere on the opposite side. Mathematically, we can assume that our arbitrary starting state $|\psi\rangle$ is given by:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right)|1\rangle \quad (2.13)$$

where θ is the “latitude” and ϕ the “longitude” angles of $|\psi\rangle$ on the Bloch sphere. To obtain the antipodal point we move, just as we would on Earth, to the equivalent latitude in the opposite hemisphere and shift the longitude by 180° (i.e., π radians). Given the aforementioned definition of $|\psi\rangle$, the mathematical form of the antipodal state, $|\bar{\psi}\rangle$, must therefore be:

$$\begin{aligned} |\psi^\perp\rangle &= \cos\left(\frac{\pi - \theta}{2}\right)|0\rangle + e^{i(\phi+\pi)} \sin\left(\frac{\pi - \theta}{2}\right)|1\rangle \\ &= \cos\left(\frac{\pi - \theta}{2}\right)|0\rangle - e^{i(\phi)} \sin\left(\frac{\pi - \theta}{2}\right)|1\rangle \\ &= \sin\left(\frac{\theta}{2}\right)|0\rangle - e^{i\phi} \cos\left(\frac{\theta}{2}\right)|1\rangle \end{aligned} \quad (2.14)$$

where we have used the identities $\cos(\frac{\pi-\theta}{2}) = \sin(\frac{\theta}{2})$ and $\sin(\frac{\pi-\theta}{2}) = \cos(\frac{\theta}{2})$.

Having understood the relationship between the mathematical form of an arbitrary starting state, $|\psi\rangle$ to that of its true antipodal state, $|\psi^\perp\rangle$, we can now check whether $X|\psi\rangle = |\psi^\perp\rangle$, and hence, whether X qualifies as a true NOT gate for an arbitrary qubit. By direct evaluation we have:

$$\begin{aligned} X|\psi\rangle &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ e^{i\phi} \sin\left(\frac{\theta}{2}\right) \end{pmatrix} = \begin{pmatrix} e^{i\phi} \sin\left(\frac{\theta}{2}\right) \\ \cos\left(\frac{\theta}{2}\right) \end{pmatrix} \\ &= e^{i\phi} \sin\left(\frac{\theta}{2}\right)|0\rangle + \cos\left(\frac{\theta}{2}\right)|1\rangle \end{aligned} \quad (2.15)$$

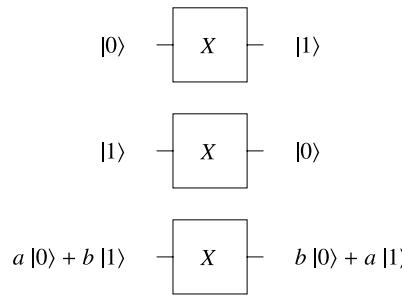


Fig. 2.17 The affect of the Pauli X gate operation on the computational basis states and an arbitrary pure state of a single qubit. The Pauli X gate “negates” the computational basis states correctly, but not an arbitrary superposition state! So the Pauli X gate is not a universal NOT gate for qubits. The universal NOT gate for qubits is discussed in Chap. 11

We are free to multiply by any overall phase factor we please since states that differ only in global phase are indistinguishable. As the amplitude of the $|0\rangle$ component of the true $|\psi^\perp\rangle$ state is $\sin(\theta/2)$, we multiply through (2.15) by $e^{-i\phi}$. Hence, the result of $X|\psi\rangle$ can be written as:

$$X|\psi\rangle = \sin\left(\frac{\theta}{2}\right)|0\rangle + e^{-i\phi} \cos\left(\frac{\theta}{2}\right)|1\rangle \neq |\psi^\perp\rangle \quad (2.16)$$

This is not $|\psi^\perp\rangle$. Hence, it is clear that $X|\psi\rangle$ does *not* negate an arbitrary single qubit state $|\psi\rangle$ since the result we get is not $|\psi^\perp\rangle$. Thus although, in classical computing, we can legitimately call the gate whose matrix is $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ the “NOT” gate, we really ought not to use this name in the context of quantum computing.

We shall see in Chap. 11 that there is, in fact, *no* universal quantum NOT gate! That is, there is no fixed quantum gate that correctly negates every qubit it is handed.

2.4.1.5 Hadamard Gate

One of the most useful single qubit gates, in fact perhaps *the* most useful one, is the Hadamard gate, H . The Hadamard gate is defined by the matrix:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (2.17)$$

It acts, as depicted in Fig. 2.18, so as to map computational basis states into superposition states and vice versa:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad (2.18)$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (2.19)$$

Fig. 2.18 The icon for the 1-qubit Walsh-Hadamard gate, H and its affect on computational basis states

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|1\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$\begin{aligned} |0\rangle &\xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |0\rangle &\xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) &= \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle \\ &\vdots \\ |0\rangle &\xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \end{aligned}$$

Fig. 2.19 By applying n H gates independently to n qubits, all prepared initially in state $|0\rangle$, we can create an n -qubit superposition whose component eigenstates are the binary representation of all the integers in the range 0 to $2^n - 1$. Thus, a superposition containing *exponentially* many terms can be prepared using only a *polynomial* number of operations. This trick is used in a great many quantum algorithms to load a quantum memory register efficiently with an equally weighted superposition of all the numbers it can contain

When the Hadamard gate H acts on a computational basis state $|x\rangle$ it transforms the input according to $H|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^x|1\rangle)$.

The Hadamard is one of the unsung heroes of quantum computing. It is a deceptively simple looking gate but it harbors a remarkable property that, if you think about it, turns out to be of vital importance to quantum computing. If you prepare n qubits each in the state $|0\rangle$ and you apply to each qubit, in parallel, its own Hadamard gate, then, as shown in Fig. 2.19, the state produced is an equal superposition of all the integers in the range 0 to $2^n - 1$.

$$H|0\rangle \otimes H|0\rangle \otimes \cdots \otimes H|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle \quad (2.20)$$

where $|j\rangle$ is the computational basis state indexed by the binary number that would correspond to the number j in base-10 notation. For example, in a 7-qubit register the state “ $|19\rangle$ ” corresponds to the computational basis state $|0010011\rangle$. The first two bits (00) are padding to make the binary number 7 bits in length, and 10011_2 (i.e., 10011 in base 2) corresponds to 19_{10} (i.e. 19 in base-10).

The utility of the Hadamard gate derives from that fact that by applying, in parallel, a separate Hadamard gate to each of n qubits, each initially in the state $|0\rangle$,

we can create an n -qubit superposition containing 2^n component eigenstates. These eigenstates represent all the possible bit strings one can write using n bits. The importance of this capability is often overlooked. But, in reality, it is one of the most important tricks of quantum computing as it gives us the ability to load exponentially many indices into a quantum computer using only polynomially many operations. Had Nature been unkind, and had we had to enter the different bit-strings individually, as we do in classical computing, then quantum computing would have had far less potential for breakthroughs in computational complexity.

2.4.2 Rotations About the x -, y -, and z -Axes

Having seen a couple of examples of special quantum logic gates (i.e., $\sqrt{\text{NOT}}$ and H) we next turn to the question of what is the most general kind of quantum gate for a single qubit. To address this, we must first introduce the family of quantum gates that perform rotations about the three mutually perpendicular axes of the Bloch sphere.

A single qubit pure state is represented by a point on the surface of the Bloch sphere. The effect of a single qubit gate that acts in this state is to map it to some other point on the Bloch sphere. The gates that rotate states around the x -, y -, and z -axes are of special significance since we will be able to decompose an arbitrary 1-qubit quantum gate into sequences of such rotation gates.

First, let's fix our reference frame with respect to which arbitrary single qubit pure states is defined. We choose three mutually perpendicular axes, x -, y -, and z -, or equivalently, three polar coordinates, a radius r (which is unity for all points on the surface of the Bloch sphere) and two angles θ (the latitude, measured monotonically from the North pole to the South pole over the interval $0 \leq \theta \leq \pi$) and ϕ the longitude (measured monotonically as we rotate around the z -axis in a clockwise fashion. So any point on the surface of the Bloch sphere can be specified using its (x, y, z) coordinates or, equivalently, its (r, θ, ϕ) coordinates. Right? Well actually not quite right since a general qubit state also must specify an overall phase factor. But let's ignore this for now. These two coordinate systems are related via the equations:

$$x = r \sin(\theta) \cos(\phi) \quad (2.21)$$

$$y = r \sin(\theta) \sin(\phi) \quad (2.22)$$

$$z = r \cos(\theta) \quad (2.23)$$

So what are the quantum gates that rotate this state about the x -, y -, or z -axes? We claim that these gates, illustrated in Figs. 2.20, 2.21, and 2.22, can be built from the Pauli X , Y , Z , matrices, and the fourth Pauli matrix, $\mathbb{1}$, can be used to achieve a global overall phase shift. Specifically, let's define the following unitary matrices, $R_x(\theta)$, $R_y(\theta)$, $R_z(\theta)$, and Ph from Hamiltonians chosen to be, respectively, the four Pauli matrices, X , Y , Z , and I (the identity matrix). That is, we have:

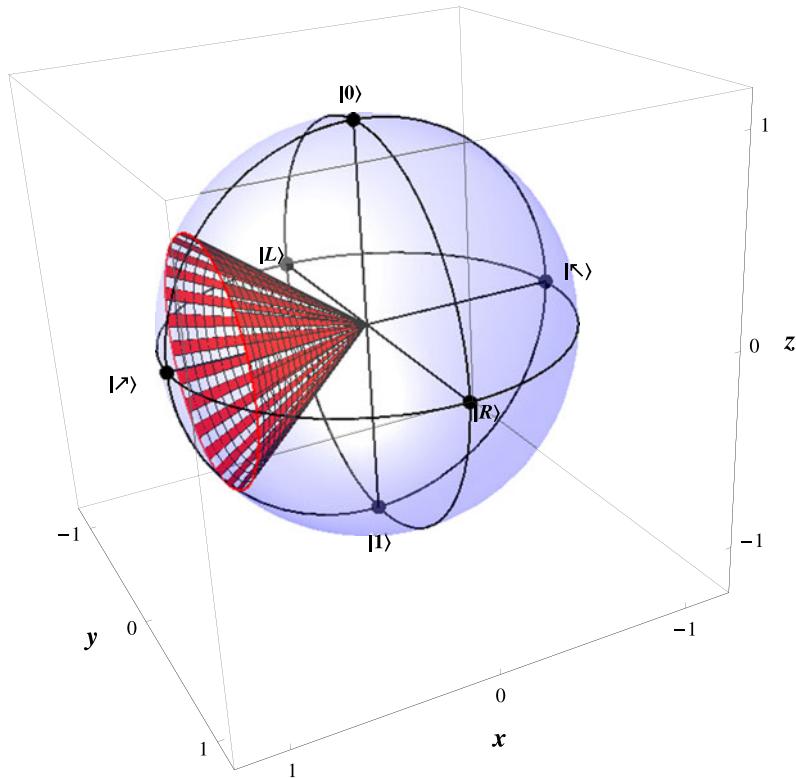


Fig. 2.20 An $R_x(\theta)$ gate maps a state $|\psi\rangle$ on the surface of the Bloch sphere to a new state, $R_x(\theta)|\psi\rangle$, represented by the point obtained by rotating a radius vector from the center of the Bloch sphere to $|\psi\rangle$ through an angle $\frac{\theta}{2}$ around the x -axis. Note that a rotation of 4π is needed to return to the original state

$$R_x(\alpha) = \exp(-i\alpha X/2) = \begin{pmatrix} \cos(\frac{\alpha}{2}) & -i \sin(\frac{\alpha}{2}) \\ -i \sin(\frac{\alpha}{2}) & \cos(\frac{\alpha}{2}) \end{pmatrix} \quad (2.24)$$

$$R_y(\alpha) = \exp(-i\alpha Y/2) = \begin{pmatrix} \cos(\frac{\alpha}{2}) & -\sin(\frac{\alpha}{2}) \\ \sin(\frac{\alpha}{2}) & \cos(\frac{\alpha}{2}) \end{pmatrix} \quad (2.25)$$

$$R_z(\alpha) = \exp(-i\alpha Z/2) = \begin{pmatrix} e^{-i\alpha/2} & 0 \\ 0 & e^{i\alpha/2} \end{pmatrix} \quad (2.26)$$

$$Ph(\delta) = e^{i\delta} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (2.27)$$

Consider the gate $R_z(\alpha)$. Let's see how this gate transforms an arbitrary single qubit state $|\psi\rangle = \cos(\frac{\theta}{2})|0\rangle + e^{i\phi} \sin(\frac{\theta}{2})|1\rangle$.

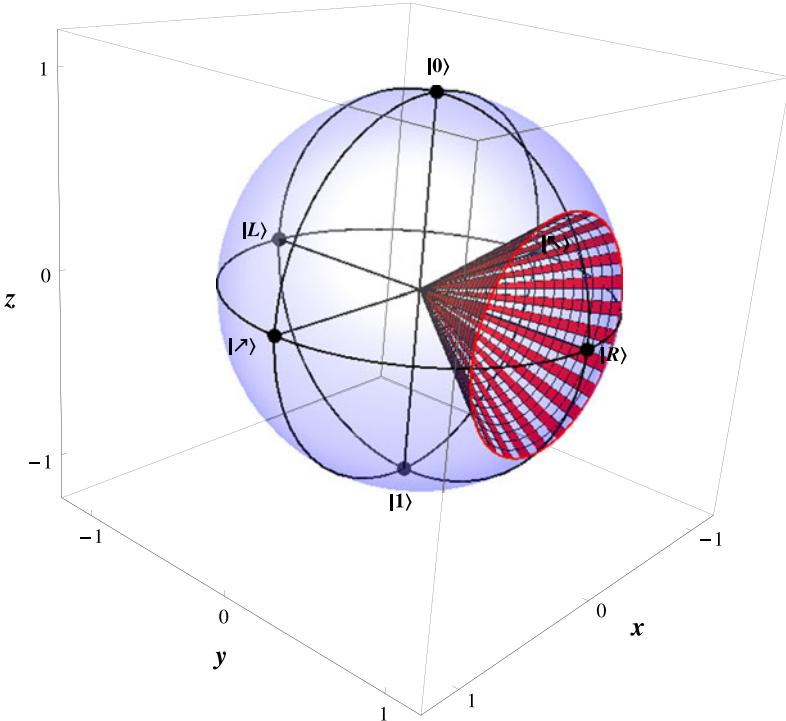


Fig. 2.21 An $R_y(\theta)$ gate maps a state $|\psi\rangle$ on the surface of the Bloch sphere to a new state, $R_y(\theta)|\psi\rangle$, represented by the point obtained by rotating a radius vector from the center of the Bloch sphere to $|\psi\rangle$ through an angle $\frac{\theta}{2}$ around the y -axis. Note that a rotation of 4π is needed to return to the original state

$$\begin{aligned}
 R_z(\alpha)|\psi\rangle &= \begin{pmatrix} e^{-i\alpha/2} & 0 \\ 0 & e^{i\alpha/2} \end{pmatrix} \cdot \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ e^{i\phi} \sin\left(\frac{\theta}{2}\right) \end{pmatrix} \\
 &= \begin{pmatrix} e^{-i\alpha/2} \cos\left(\frac{\theta}{2}\right) \\ e^{i\alpha/2} e^{i\phi} \sin\left(\frac{\theta}{2}\right) \end{pmatrix} \\
 &= e^{-i\alpha/2} \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\alpha/2} e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle
 \end{aligned} \tag{2.28}$$

We are free to multiply this state by any overall phase factor we please since for any quantum state $|\chi\rangle$, the states $|\chi\rangle$ and $e^{i\gamma}|\chi\rangle$ are indistinguishable. So let's multiply by an overall phase factor of $\exp(i\alpha/2)$, which gives us the state:

$$R_z(\alpha)|\psi\rangle \equiv \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i(\phi+\alpha)} \sin\left(\frac{\theta}{2}\right) |1\rangle \tag{2.29}$$

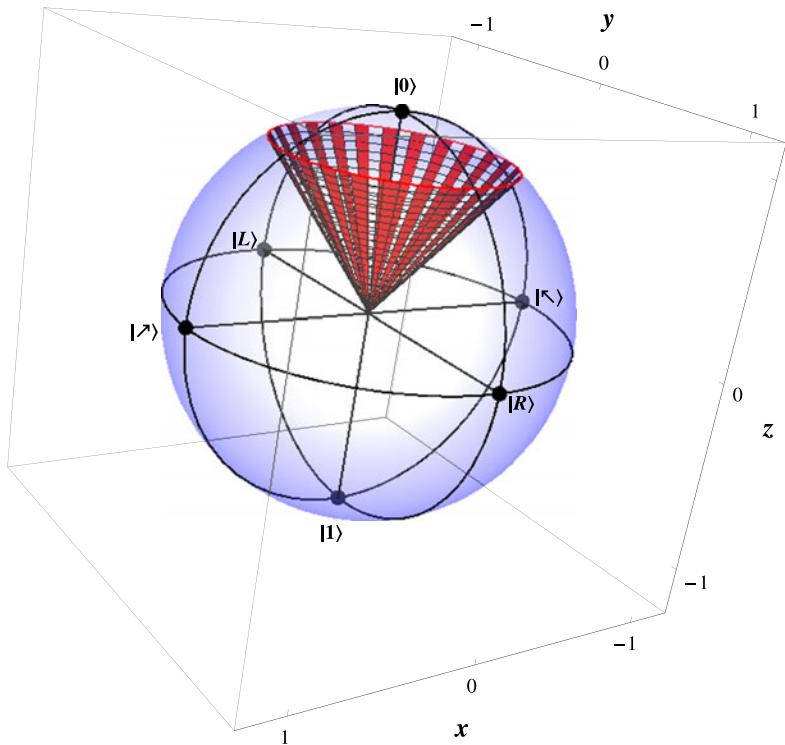


Fig. 2.22 An $R_z(\theta)$ gate maps a state $|\psi\rangle$ on the surface of the Bloch sphere to a new state, $R_z(\theta)|\psi\rangle$, represented by the point obtained by rotating a radius vector from the center of the Bloch sphere to $|\psi\rangle$ through an angle $\frac{\theta}{2}$ around the z -axis. Note that a rotation of 4π is needed to return to the original state

where \equiv is to be read as “equal up to an unimportant arbitrary overall phase factor”. Hence the action of the $R_z(\alpha)$ gate on $|\psi\rangle$ has been to advance the angle ϕ by α and hence rotate the state about the z -axis through angle α . This is why we call $R_z(\alpha)$ a z -rotation gate. We leave it to the exercises for you to prove that $R_x(\alpha)$ and $R_y(\alpha)$ rotate the state about the x - and y -axes respectively.

Rotations on the Bloch sphere do not conform to commonsense intuitions about rotations that we have learned from our experience of the everyday world. In particular, usually, a rotation of 2π radians (i.e., 360 degrees) of a solid object about any axis, restores that object to its initial orientation. However, this is not true of rotations on the Bloch sphere! When we rotate a quantum state through 2π on the Bloch sphere we don’t return it to its initial state. Instead we pick up a phase factor. To see this, let’s compute the effect of rotating our arbitrary single qubit pure state, $|\psi\rangle$ about the z -axis through 2π radians. We have:



Fig. 2.23 “Dirac’s Belt” uses a commonplace belt to illustrate that topology of a single qubit state wherein a rotation of 4π (two full twists) is required to restore the belt to its starting configuration

$$\begin{aligned}
 R_z(2\pi)|\psi\rangle &= \begin{pmatrix} e^{-i\pi} & 0 \\ 0 & e^{i\pi} \end{pmatrix} \cdot \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ e^{i\phi} \sin\left(\frac{\theta}{2}\right) \end{pmatrix} \\
 &= \begin{pmatrix} -\cos\left(\frac{\theta}{2}\right) \\ -e^{i\phi} \sin\left(\frac{\theta}{2}\right) \end{pmatrix} = -|\psi\rangle
 \end{aligned} \tag{2.30}$$

which has an extra overall phase of -1 . To restore a state back to its original form we need to rotate it through 4π on the Bloch sphere.

Have you ever encountered anything like this in your everyday world? You probably think not, but you’d be wrong! Find yourself a thick leather belt. Have a friend hold one end flat and apply a rotation of 2π to the other end, i.e., one full twist (see Fig. 2.23). Now try to loop the belt around itself without tilting either end. In so doing, can you remove the twist? After some experimentation you should be convinced that the twist is there to stay and there is no way to remove it and yet keep the orientations of the ends of the belt fixed relative to one another. By analogy, a rotation of 2π has not restored the belt to its initial (flat and twist free) state. Ok so let’s try again. Have a friend hold one end flat and apply a rotation of 4π to the other end, i.e., two full twists. Now try to loop the belt around itself without tilting either end. After a little experimentation you should find, to the surprise of most people, that the twist has gone! In other words, a rotation of 4π to one end of the belt has resulted in a state that is equivalent to the original (flat and twist free) state of the belt.

2.4.2.1 NOT, $\sqrt{\text{NOT}}$, and Hadamard from Rotation Gates

The NOT, $\sqrt{\text{NOT}}$, and Hadamard gates can all be obtained via sequences of rotation gates. For example,

$$\text{NOT} \equiv R_x(\pi) \cdot Ph\left(\frac{\pi}{2}\right) \quad (2.31)$$

$$\text{NOT} \equiv R_y(\pi) \cdot R_z(\pi) \cdot Ph\left(\frac{\pi}{2}\right) \quad (2.32)$$

$$\sqrt{\text{NOT}} \equiv R_x\left(\frac{\pi}{2}\right) \cdot Ph\left(\frac{\pi}{4}\right) \quad (2.33)$$

$$\sqrt{\text{NOT}} \equiv R_z\left(-\frac{\pi}{2}\right) \cdot R_y\left(\frac{\pi}{2}\right) \cdot R_z\left(\frac{\pi}{2}\right) \cdot Ph\left(\frac{\pi}{4}\right) \quad (2.34)$$

$$H \equiv R_x(\pi) \cdot R_y\left(\frac{\pi}{2}\right) \cdot Ph\left(\frac{\pi}{2}\right) \quad (2.35)$$

$$H \equiv R_y\left(\frac{\pi}{2}\right) \cdot R_z(\pi) \cdot Ph\left(\frac{\pi}{2}\right) \quad (2.36)$$

2.4.3 Arbitrary 1-Qubit Gates: The Pauli Decomposition

So far we have seen how *specific* 1-qubit gates can be decomposed into sequences of rotation gates, i.e., $R_x(\cdot)$, $R_y(\cdot)$, $R_z(\cdot)$, and phase gates, i.e., $Ph(\cdot)$. Next we consider how to decompose an arbitrary, maximally general, 1-qubit gate.

A maximally general 1-qubit gate will correspond to some 2×2 unitary matrix, U . As U is unitary the magnitude of its determinant must be unity, i.e., $|\det(U)| = 1$. This equation can be satisfied by $\det(U)$ taking on any of the values $+1$, -1 , $+i$, or $-i$. If $\det(U) = +1$ then U is said to be “special unitary”. If not, we can always write U in the form $U = e^{i\delta} V$ where V is a special unitary matrix, i.e., $\det(V) = +1$. So to find a circuit for the unitary matrix U it is sufficient to find a circuit for the special unitary matrix V , because simply appending a phase shift gate $Ph(\delta)$ to the circuit for V will give a circuit for U . This is easily seen by realizing

$$U = e^{i\delta} V = e^{i\delta} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot V = \begin{pmatrix} e^{i\delta} & 0 \\ 0 & e^{i\delta} \end{pmatrix} \cdot V = Ph(\delta) \cdot V$$

As V is a 2×2 special unitary matrix its rows and columns are orthonormal and, its elements, most generally, are complex numbers. Hence, V must have the form:

$$V = \begin{pmatrix} \alpha & -\bar{\beta} \\ \beta & \bar{\alpha} \end{pmatrix} \quad (2.37)$$

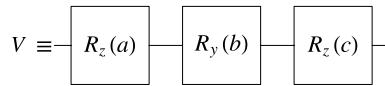


Fig. 2.24 Any 1-qubit special unitary gate can be decomposed into a rotation about the z -axis, the y -axis, and the z -axis

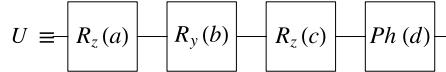


Fig. 2.25 Any 1-qubit unitary gate can be decomposed into a rotation about the z -axis, the y -axis, the z -axis, followed by a phase shift

where α and β are arbitrary complex numbers that satisfy the determinant equation $\det(V) = \alpha\bar{\alpha} - \beta(-\bar{\beta}) = |\alpha|^2 + |\beta|^2 = 1$. This equation can be satisfied by picking $\alpha = e^{i\mu} \cos(\theta/2)$, and $\beta = e^{i\xi} \sin(\theta/2)$. This means we can also write the matrix for V as:

$$\begin{aligned} V &= \begin{pmatrix} \alpha & -\bar{\beta} \\ \beta & \bar{\alpha} \end{pmatrix} \quad \text{with } \alpha \rightarrow e^{i\mu} \cos(\theta/2) \text{ and } \beta \rightarrow e^{i\xi} \sin(\theta/2) \\ &= \begin{pmatrix} e^{i\mu} \cos(\theta/2) & -e^{-i\xi} \sin(\theta/2) \\ e^{i\xi} \sin(\theta/2) & e^{-i\mu} \cos(\theta/2) \end{pmatrix} \end{aligned} \quad (2.38)$$

But this matrix can also be obtained as the product of the three gates $R_z(a) \cdot R_y(b) \cdot R_z(c)$ with $a \rightarrow -(\mu - \xi)$, $b \rightarrow \theta$, and $c \rightarrow -(\mu + \xi)$.

$$\begin{aligned} R_z(a) \cdot R_y(b) \cdot R_z(c) &= \begin{pmatrix} e^{-\frac{ia}{2}-\frac{ic}{2}} \cos\left(\frac{b}{2}\right) & -e^{\frac{ic}{2}-\frac{ia}{2}} \sin\left(\frac{b}{2}\right) \\ e^{\frac{ia}{2}-\frac{ic}{2}} \sin\left(\frac{b}{2}\right) & e^{\frac{ia}{2}+\frac{ic}{2}} \cos\left(\frac{b}{2}\right) \end{pmatrix} \\ &\quad \text{with } a \rightarrow -(\mu - \xi), b \rightarrow \theta, \text{ and } c \rightarrow -(\mu + \xi) \\ &= \begin{pmatrix} e^{i\mu} \cos(\theta/2) & -e^{-i\xi} \sin(\theta/2) \\ e^{i\xi} \sin(\theta/2) & e^{-i\mu} \cos(\theta/2) \end{pmatrix} = V \end{aligned} \quad (2.39)$$

Thus, any 1-qubit special unitary gate V can be decomposed into the form $R_z(a) \cdot R_y(b) \cdot R_z(c)$ as shown in Fig. 2.24. Hence, any 1-qubit unitary gate, U can be decomposed into the form:

$$U \equiv R_z(a) \cdot R_y(b) \cdot R_z(c) \cdot Ph(d) \quad (2.40)$$

as shown in Fig. 2.25.

2.4.4 Decomposition of R_x Gate

Lest it seem peculiar that we can achieve an arbitrary 1-qubit gate without performing a rotation about the x -axis, we note that it is possible to express rotations about the x -axis purely in terms of rotations about the y - and z -axes. Specifically, we have the identities:

$$\begin{aligned} R_x(\theta) &= \exp(-i\theta X/2) = \begin{pmatrix} \cos(\frac{\theta}{2}) & i \sin(\frac{\theta}{2}) \\ i \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} \\ &\equiv R_z(-\pi/2) \cdot R_y(\theta) \cdot R_z(\pi/2) \\ &\equiv R_y(\pi/2) \cdot R_z(\theta) \cdot R_y(-\pi/2) \end{aligned} \quad (2.41)$$

2.5 Controlled Quantum Gates

To perform non-trivial computations it is often necessary to change the operation applied to one set of qubits depending upon the values of some other set of qubits. The gates that implement these “if-then-else” type operations are called *controlled* gates. Some examples of controlled gates that appeared earlier in this chapter are CNOT (controlled-NOT), FREDKIN (controlled-SWAP), and TOFFOLI (controlled-controlled-NOT). The justification for calling these gates “controlled” stems from their effect on the computational basis states. For example, CNOT transforms the computational basis states such that the second qubit is negated if and only if the first qubit is in state $|1\rangle$.

$$|00\rangle \xrightarrow{\text{CNOT}} |00\rangle \quad (2.42)$$

$$|01\rangle \xrightarrow{\text{CNOT}} |01\rangle \quad (2.43)$$

$$|10\rangle \xrightarrow{\text{CNOT}} |11\rangle \quad (2.44)$$

$$|11\rangle \xrightarrow{\text{CNOT}} |10\rangle \quad (2.45)$$

Hence, the value of the second qubit (called the “target” qubit) is *controlled* by the first qubit (called the “control” qubit).

Likewise, under the action of the FREDKIN gate the second and third qubits are swapped if and only if the first qubit is in state $|1\rangle$. So the FREDKIN gate performs a controlled-SWAP operation.

$$|000\rangle \xrightarrow{\text{FREDKIN}} |000\rangle \quad (2.46)$$

$$|001\rangle \xrightarrow{\text{FREDKIN}} |001\rangle \quad (2.47)$$

$$|010\rangle \xrightarrow{\text{FREDKIN}} |010\rangle \quad (2.48)$$

$$|011\rangle \xrightarrow{\text{FREDKIN}} |011\rangle \quad (2.49)$$

$$|100\rangle \xrightarrow{\text{FREDKIN}} |100\rangle \quad (2.50)$$

$$|101\rangle \xrightarrow{\text{FREDKIN}} |110\rangle \quad (2.51)$$

$$|110\rangle \xrightarrow{\text{FREDKIN}} |101\rangle \quad (2.52)$$

$$|111\rangle \xrightarrow{\text{FREDKIN}} |111\rangle \quad (2.53)$$

It is also possible to have controlled gates with multiple control qubits and multiple target qubits. The action of the TOFFOLI gate is to negate the third qubit (i.e., the target qubit) if and only if the first two qubits (the control qubits) are in state $|11\rangle$. Thus the TOFFOLI gate has two control qubits and one target qubit.

$$|000\rangle \xrightarrow{\text{TOFFOLI}} |000\rangle \quad (2.54)$$

$$|001\rangle \xrightarrow{\text{TOFFOLI}} |001\rangle \quad (2.55)$$

$$|010\rangle \xrightarrow{\text{TOFFOLI}} |010\rangle \quad (2.56)$$

$$|011\rangle \xrightarrow{\text{TOFFOLI}} |011\rangle \quad (2.57)$$

$$|100\rangle \xrightarrow{\text{TOFFOLI}} |100\rangle \quad (2.58)$$

$$|101\rangle \xrightarrow{\text{TOFFOLI}} |101\rangle \quad (2.59)$$

$$|110\rangle \xrightarrow{\text{TOFFOLI}} |111\rangle \quad (2.60)$$

$$|111\rangle \xrightarrow{\text{TOFFOLI}} |110\rangle \quad (2.61)$$

Now all this is very well, but aren't CNOT, FREDKIN and TOFFOLI not just classical reversible gates? Well yes they are! But in addition they are also quantum gates because the transformations they perform (i.e., permutations of computational basis states) also happen to be unitary. But indeed, controlled *quantum* gates can be far more sophisticated than controlled classical gates. For example, the natural quantum generalization of the controlled-NOT gate is the controlled-*U* gate:

$$\text{controlled-}U \equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{11} & U_{12} \\ 0 & 0 & U_{21} & U_{22} \end{pmatrix} \quad (2.62)$$

where $U = \begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix}$ is an arbitrary 1-qubit gate.

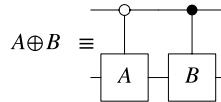


Fig. 2.26 The quantum circuit corresponding to a gate that performs different control actions according to whether the top qubit is $|0\rangle$ or $|1\rangle$

2.5.1 Meaning of a “Controlled” Gate in the Quantum Context

If we are using CNOT, FREDKIN or TOFFOLI gates within the context of classical reversible computing their inputs are only ever classical bits. Hence, there is no problem imagining *reading* each control bit to determine what action to perform on the target bit. But if we use these gates in the context of quantum computing, where they may be required to act on arbitrary superposition states, we ought to question whether it continues to make sense to speak of “controlled” gates because, in the quantum case, the act of reading the control qubit will, in general, perturb it.

The answer is that *we do not need to read control bits* during the application of a controlled quantum gate! Instead if a controlled quantum gate acts on a superposition state *all* of the control actions are performed in parallel to a degree commensurate with the amplitude of the corresponding control qubit eigenstate within the input superposition state.

For example, suppose A and B are a pair of unitary matrices corresponding to arbitrary 1-qubit quantum gates. Then the gate defined by their direct sum:

$$A \oplus B = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & 0 & 0 \\ A_{21} & A_{22} & 0 & 0 \\ 0 & 0 & B_{11} & B_{12} \\ 0 & 0 & B_{21} & B_{22} \end{pmatrix} \quad (2.63)$$

performs a “controlled” operation in the following sense. If the first qubit is in state $|0\rangle$ then the operation A is applied to the second qubit. Conversely, if the first qubit is in state $|1\rangle$ then the operation B is applied to the second qubit. And if the control qubit is some superposition of $|0\rangle$ and $|1\rangle$ then both control actions are performed to some degree. The quantum circuit for such a gate is shown in Fig. 2.26. Don’t believe me? Let’s work it out explicitly.

If the first qubit is in state $|0\rangle$ we can write the input as a state of the form $|0\rangle(a|0\rangle + b|1\rangle)$, and if the first qubit is in state $|1\rangle$ we write the input as a state of

the form $|1\rangle(a|0\rangle + b|1\rangle)$. For the first case, when the gate acts we therefore obtain:

$$\begin{aligned}
 (A \oplus B)(|0\rangle \otimes (a|0\rangle + b|1\rangle)) &= \begin{pmatrix} A_{11} & A_{12} & 0 & 0 \\ A_{21} & A_{22} & 0 & 0 \\ 0 & 0 & B_{11} & B_{12} \\ 0 & 0 & B_{21} & B_{22} \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ 0 \\ 0 \end{pmatrix} \\
 &= \begin{pmatrix} aA_{11} + bA_{12} \\ aA_{21} + bA_{22} \\ 0 \\ 0 \end{pmatrix} \\
 &= (aA_{11} + bA_{12})|00\rangle + (aA_{21} + bA_{22})|01\rangle \\
 &= |0\rangle \otimes A(a|0\rangle + b|1\rangle)
 \end{aligned} \tag{2.64}$$

Likewise, for the second case, when the gate acts on an input of the form $|1\rangle \otimes (a|0\rangle + b|1\rangle)$ we obtain:

$$\begin{aligned}
 (A \oplus B)(|1\rangle \otimes (a|0\rangle + b|1\rangle)) &= \begin{pmatrix} A_{11} & A_{12} & 0 & 0 \\ A_{21} & A_{22} & 0 & 0 \\ 0 & 0 & B_{11} & B_{12} \\ 0 & 0 & B_{21} & B_{22} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ a \\ b \end{pmatrix} \\
 &= \begin{pmatrix} 0 \\ 0 \\ aB_{11} + bB_{12} \\ aB_{21} + bB_{22} \end{pmatrix} \\
 &= (aB_{11} + bB_{12})|10\rangle + (aB_{21} + bB_{22})|11\rangle \\
 &= |1\rangle \otimes B(a|0\rangle + b|1\rangle)
 \end{aligned} \tag{2.65}$$

Putting these results together, when the 2-qubit controlled gate $(A \oplus B)$ acts on a *general* 2-qubit superposition state $|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$ the control qubit is no longer purely $|0\rangle$ or purely $|1\rangle$. Nevertheless, the linearity of quantum mechanics guarantees that the correct control actions are performed, in the correct proportions, on the target qubit.

$$(A \oplus B)|\psi\rangle = |0\rangle \otimes A(a|0\rangle + b|1\rangle) + |1\rangle \otimes B(c|0\rangle + d|1\rangle) \tag{2.66}$$

2.5.2 Semi-Classical Controlled Gates

Note that although we do not *have* to read the values of control qubits in order for controlled actions to be imposed on target qubits, we *may* do so if we wish. Specifically, in the traditional model of quantum computation one prepares a quantum state, evolves it unitarily through some quantum circuit, and then makes a *final* measurement on the output qubits. The values of the control qubits contained within such a

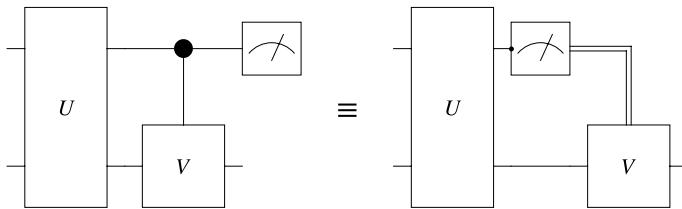


Fig. 2.27 Semi-classical quantum gates. Measurements of a control qubit made after a controlled gate can be moved before the gate and the subsequent controlled gate then be classically controlled. Griffiths and Niu used this trick in their semi-classical QFT [213], and Brassard used it in his quantum teleportation circuit [75]

quantum circuit are never read. However, we don't have to operate quantum circuits this way. If we want, we can move the final measurements on control qubits to earlier parts of the quantum circuit, and use the resulting classical bits to determine which gate operation to apply to the corresponding target qubits. Such a strategy will, of course, change the final state produced by the quantum circuit on any particular run, but it won't change their *statistical* properties averaged over many repetitions. Such intermediate measurements have been used to make a "semi-classical Fourier transform" [213] and also within a quantum circuit for teleportation [75].

For example, as shown in Fig. 2.27 the control qubits of the controlled gates in the quantum Fourier transform can be measured immediately after they have acted and the resulting classical bit used to classically condition a subsequent controlled gate operation. The ability to move some final measurements to earlier stages of a quantum circuit and then condition subsequent gate operations on their (classical) outcomes can be of practical value by lowering the engineering complexity required to achieve practical quantum computational hardware.

2.5.3 Multiply-Controlled Gates

Controlled gates can be generalized to have multiple controls as shown in Fig. 2.28. Here a different operation is performed on the third qubit depending on the state of the top two qubits. Such multiply-controlled quantum gates are quite common in practical quantum circuits. Note, however, that the number of distinct states of the controls grows exponentially with the number of controls. So it becomes more difficult to actually build multiply-controlled gates beyond just a few control qubits.

2.5.4 Circuit for Controlled- U

Regardless of when qubits are to be read, we should like to know how to decompose these controlled gates into a simpler set of standard gates. Factoring a controlled gate

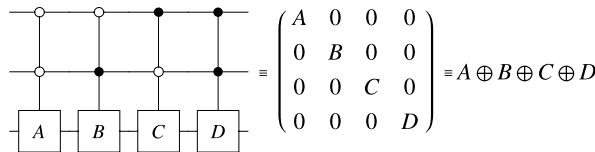


Fig. 2.28 The quantum circuit corresponding to a gate that performs different control actions according to whether the top two qubits are $|00\rangle$, $|01\rangle$, $|10\rangle$, or $|11\rangle$

as in $A \oplus B = (\mathbb{1} \otimes A) \cdot (\mathbb{1} \otimes A^{-1} \cdot B)$ where $\mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, we can see that the core “controlled” component of the gate is really a gate of the form:

$$\text{controlled-}U \equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{11} & U_{12} \\ 0 & 0 & U_{21} & U_{22} \end{pmatrix} \quad (2.67)$$

where the U_{ij} are the elements of an arbitrary 1-qubit gate $U = A^{-1} \cdot B$. We call a 2-qubit gate of the form $\begin{pmatrix} \mathbb{1} & \hat{0} \\ \hat{0} & U \end{pmatrix}$ a controlled- U gate.

We can construct a quantum circuit for a 2-qubit controlled- U gate in terms of CNOT gates and 1-qubit gates as follows. Let U be an arbitrary 1-qubit gate having a single qubit (Pauli) decomposition of the form $U = e^{ia} R_z(b) \cdot R_y(c) \cdot R_z(d)$. The action of the controlled- U gate is to do nothing to the target qubit when the control qubit is $|0\rangle$ and to apply U to the target qubit when the control qubit is $|1\rangle$. The act of “doing nothing” is mathematically equivalent to applying the identity gate to the target. So given the quantum circuit decomposition for computing U , what is a quantum circuit that computes controlled- U ?

By (2.40) there exist angles a, b, c , and d such that:

$$U = e^{ia} R_z(b) \cdot R_y(c) \cdot R_z(d) \quad (2.68)$$

Given these angles, define matrices A, B, C as follows:

$$A = R_z\left(\frac{d-b}{2}\right) \quad (2.69)$$

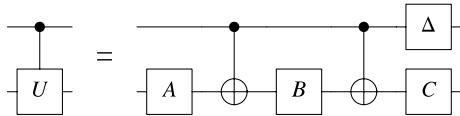
$$B = R_y\left(-\frac{c}{2}\right) \cdot R_z\left(-\frac{d+b}{2}\right) \quad (2.70)$$

$$C = R_z(b) \cdot R_y\left(\frac{c}{2}\right) \quad (2.71)$$

$$\Delta = \text{diag}(1, e^{ia}) \quad (2.72)$$

We claim that the circuit shown in Fig. 2.29 computes controlled- U . Here is how it works. When the control qubit is in state $|0\rangle$ the Δ gate does change it because $\Delta|0\rangle = |0\rangle$ (with no phase addition). The control qubits of the CNOT gates are

Fig. 2.29 A quantum circuit for a controlled- U gate, where U is an arbitrary 1-qubit gate



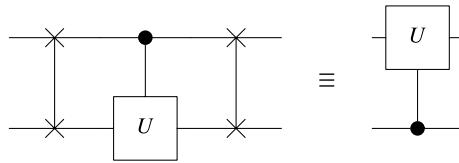
therefore also $|0\rangle$ and so the CNOTs do not do anything to the target qubit. Hence, the transformation to which the target qubit will be subject when the control qubit in the circuit is $|0\rangle$ is $C \cdot B \cdot A$. Note that the order is reversed with respect to the left to right sequence in the circuit diagram because, mathematically, if the A gate acts first, then the B gate, and then the C gate, the matrices must be multiplied in the order $C \cdot B \cdot A$ since when this object acts in an input state $|\psi\rangle$ we want the grouping to be $(C \cdot (B \cdot (A|\psi\rangle)))$ (gate A first then gate B then gate C). A little algebra shows that the net effect of these three operations is the identity (as required).

$$C \cdot B \cdot A \equiv R_z(b) \cdot R_y\left(\frac{c}{2}\right) \cdot R_y\left(-\frac{c}{2}\right) \cdot R_z\left(-\frac{d+b}{2}\right) \cdot R_z\left(\frac{d-b}{2}\right) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (2.73)$$

Next we consider what happens when the control qubit is in state $|1\rangle$. In this case the control qubit first picks up a phase factor since $\Delta|1\rangle = e^{ia}|1\rangle$. The control qubits of the CNOT gates will all be set to $|1\rangle$, and so they will apply a NOT gate (equivalent to a Pauli-X gate) to the target qubit when the CNOT gate acts. Hence, the transformation to which the target qubit will be subject when the control qubit is $|1\rangle$ is $e^{ia}C \cdot X \cdot B \cdot X \cdot A$. To simplify this expression we need to notice that $X \cdot R_y(\theta) \cdot X \equiv R_y(-\theta)$ and $X \cdot R_z(\theta) \cdot X \equiv R_z(-\theta)$. Hence we obtain:

$$\begin{aligned} C \cdot X \cdot B \cdot X \cdot A &= R_z(b) \cdot R_y\left(\frac{c}{2}\right) \cdot X \cdot R_y\left(-\frac{c}{2}\right) \cdot R_z\left(-\frac{d+b}{2}\right) \\ &\quad \cdot X \cdot R_z\left(\frac{d-b}{2}\right) \\ &= R_z(b) \cdot R_y\left(\frac{c}{2}\right) \cdot X \cdot R_y\left(-\frac{c}{2}\right) \cdot X \cdot X \cdot R_z\left(-\frac{d+b}{2}\right) \\ &\quad \cdot X \cdot R_z\left(\frac{d-b}{2}\right) \\ &= R_z(b) \cdot R_y\left(\frac{c}{2}\right) \cdot X \cdot R_y\left(-\frac{c}{2}\right) \cdot X \cdot X \cdot R_z\left(-\frac{d+b}{2}\right) \\ &\quad \cdot X \cdot R_z\left(\frac{d-b}{2}\right) \\ &= R_z(b) \cdot R_y\left(\frac{c}{2}\right) \cdot R_y\left(\frac{c}{2}\right) \cdot R_z\left(\frac{b+d}{2}\right) \cdot R_z\left(\frac{d-b}{2}\right) \\ &= R_z(b) \cdot R_y(c) \cdot R_z(d) \end{aligned} \quad (2.74)$$

Fig. 2.30 A quantum circuit for an upside down controlled- U gate, where U is an arbitrary 1-qubit gate



Hence the circuit for controlled- U performs as follows:

$$\begin{aligned}
 \text{controlled-}U |0\rangle(a|0\rangle + b|1\rangle) &= |0\rangle \otimes C \cdot B \cdot A(a|0\rangle + b|1\rangle) \\
 &= |0\rangle \otimes (a|0\rangle + b|1\rangle) \\
 \text{controlled-}U |1\rangle(a|0\rangle + b|1\rangle) &= e^{ia}|1\rangle \otimes C \cdot X \cdot B \cdot X \cdot A(a|0\rangle + b|1\rangle) \quad (2.75) \\
 &= |1\rangle \otimes e^{ia}C \cdot X \cdot B \cdot X \cdot A(a|0\rangle + b|1\rangle) \\
 &= |1\rangle \otimes U(a|0\rangle + b|1\rangle)
 \end{aligned}$$

Thus U is applied to the target qubit if and only if the control qubit is set to $|1\rangle$.

2.5.5 Flipping the Control and Target Qubits

The control qubit does not have to be the topmost qubit in a quantum circuit. An upside down controlled- U gate would be given by SWAP · controlled- U · SWAP as shown in Fig. 2.30.

$$\text{upside-down-controlled-}U = \text{SWAP} \cdot \text{controlled-}U \cdot \text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & U_{11} & 0 & U_{12} \\ 0 & 0 & 1 & 0 \\ 0 & U_{21} & 0 & U_{22} \end{pmatrix} \quad (2.76)$$

The second qubit is now the control qubit and the first qubit the target qubit. The result is the matrix corresponding to a 2-qubit controlled quantum gate inserted into a circuit “upside down”.

2.5.6 Control-on- $|0\rangle$ Quantum Gates

Furthermore, in a controlled quantum gate the value that determines whether or not a special action is performed does not have to be $|1\rangle$; it can be $|0\rangle$ (or any other state) too. A 2-qubit quantum gate with the special action conditioned on the value of the

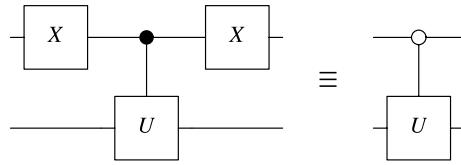


Fig. 2.31 A quantum circuit for a controlled quantum gate that acts when its control qubit is in state $|0\rangle$ (as indicated by the open circle on the control qubit) rather than state $|1\rangle$

first qubit being $|0\rangle$ instead of $|1\rangle$ is related to the usual controlled gate as follows:

$$\text{controlled}[1]-U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{11} & U_{12} \\ 0 & 0 & U_{21} & U_{22} \end{pmatrix} \quad (2.77)$$

$$\begin{aligned} \text{controlled}[0]-U &= (\text{NOT} \otimes \mathbb{1}_2) \cdot \text{controlled}[1]-U \cdot (\text{NOT} \otimes \mathbb{1}_2) \\ &= \begin{pmatrix} U_{11} & U_{12} & 0 & 0 \\ U_{21} & U_{22} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned} \quad (2.78)$$

as illustrated in Fig. 2.31.

2.5.7 Circuit for Controlled-Controlled- U

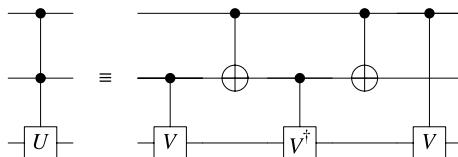
We can carry on in a similar fashion by, e.g., allowing multiple control qubits and/or target qubits. For example, earlier we interpreted the TOFFOLI gate as a controlled-controlled-NOT gate. Generalizing leads us to consider a controlled-controlled- U gate, where U is an arbitrary 1-qubit gate.

As a matrix, the controlled-controlled- U gate has the form:

$$\text{controlled-controlled-}U \equiv \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & U_{11} & U_{12} \\ 0 & 0 & 0 & 0 & 0 & 0 & U_{21} & U_{22} \end{pmatrix} \quad (2.79)$$

We can decompose a controlled-controlled- U gate into a circuit built from only CNOT gates and 1-qubit gates of the form shown in Fig. 2.32 (see [33]). Here $V = U^{1/2}$. The operation of this circuit can be understood by considering what it does to

Fig. 2.32 Quantum circuit for the controlled-controlled- U operation. Here V is any quantum gate such that $V^2 = U$



the eight possible computational basis states of a three qubit system.

$$|000\rangle \xrightarrow{\text{ctrl-ctrl-}U} |000\rangle \quad (2.80)$$

$$|001\rangle \xrightarrow{\text{ctrl-ctrl-}U} |001\rangle \quad (2.81)$$

$$|010\rangle \xrightarrow{\text{ctrl-ctrl-}U} |01\rangle \otimes (V^\dagger \cdot V|0\rangle) = |010\rangle \quad (2.82)$$

$$|011\rangle \xrightarrow{\text{ctrl-ctrl-}U} |01\rangle \otimes (V^\dagger \cdot V|1\rangle) = |011\rangle \quad (2.83)$$

$$|100\rangle \xrightarrow{\text{ctrl-ctrl-}U} |10\rangle \otimes (V \cdot V^\dagger|0\rangle) = |100\rangle \quad (2.84)$$

$$|101\rangle \xrightarrow{\text{ctrl-ctrl-}U} |10\rangle \otimes (V \cdot V^\dagger|1\rangle) = |101\rangle \quad (2.85)$$

$$|110\rangle \xrightarrow{\text{ctrl-ctrl-}U} |11\rangle \otimes V^2|0\rangle = |11\rangle \otimes U|0\rangle \quad (\text{since } V^2 = U) \quad (2.86)$$

$$|111\rangle \xrightarrow{\text{ctrl-ctrl-}U} |11\rangle \otimes V^2|1\rangle = |11\rangle \otimes U|1\rangle \quad (\text{since } V^2 = U) \quad (2.87)$$

2.6 Universal Quantum Gates

A set of gates, \mathcal{S} , is “universal” if any feasible computation can be achieved in a circuit that uses solely gates from \mathcal{S} . The most interesting universal sets of gates are those containing a single gate. The NAND gate, the NOR gate, and the NMAMJORITY gate, are all known, individually, to be universal for classical irreversible computing. Similarly, the TOFFOLI and FREDKIN gates are each known to be universal for classical reversible computing. Are there similar universal gates for *quantum* computing? If so, how many qubits does the *smallest* universal quantum gate have to have?

The fact that the closest classical gates to the quantum gates are the classical reversible gates, and these need a minimum of three bits to be universal, might lead you to expect that the smallest universal quantum gate will be a 3-qubit gate too. Indeed, there is a 3-qubit gate that is universal for quantum computing. It is called a DEUTSCH gate, and any feasible quantum computation can be achieved in a circuit built only from DEUTSCH gates acting on various triplets of qubits [137]. This gate

has the form:

$$\text{DEUTSCH} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & i \cos(\theta) & \sin(\theta) \\ 0 & 0 & 0 & 0 & 0 & 0 & \sin(\theta) & i \cos(\theta) \end{pmatrix} \quad (2.88)$$

where θ is any constant angle such that $2\theta/\pi$ is an irrational number. However, circuits for an arbitrary $2^n \times 2^n$ unitary matrix built from this gate are typically very inefficient in gate count.

Surprisingly, however, Deutsch's gate is not the smallest possibility. David Di Vincenzo and John Smolin showed that DEUTSCH's gate could be built from only 2-qubit gates [149], and Adriano Barenco showed it could be obtained using only just a single type of 2-qubit gate—the BARENCO gate [32], which has the form:

$$\text{BARENCO} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\alpha} \cos(\theta) & -ie^{i(\alpha-\phi)} \sin(\theta) \\ 0 & 0 & -ie^{i(\alpha+\phi)} \sin(\theta) & e^{i\alpha} \cos(\theta) \end{pmatrix} \quad (2.89)$$

where ϕ, α and θ are fixed irrational multiples of π and each other.

Thus, quantum gates are very different from classical gates in terms of universality. Whereas in classical reversible computing there is no 2-bit gate that is both reversible and universal, in quantum computing *almost all* 2-qubit gates are universal [80, 147]. This is quite remarkable. In particular, it means that certain *classical* reversible computations (which are described by permutation matrices and are, therefore, unitary) can potentially be implemented more efficiently using *quantum* gates than using only *classical reversible* gates. Ironically, it is conceivable that one of the nearest term large scale applications of *quantum* gates will be in implementations of (perhaps spintronic-based) “classical” reversible computers for fast, low power, reversible microprocessors.

The primary reason to study universal gates is to make the life of the experimentalist a little easier. If all quantum computations can be built from a single type of gate, then an experimentalist need only focus on how to achieve that gate in order to be guaranteed that any quantum computation is, in principle, attainable. Unfortunately, in practice, it is quite hard to use the Barenco gate as a primitive gate as it requires a 2-qubit Hamiltonian having three “tunable” parameters, ϕ, α and θ . However, luckily, the BARENCO gate is clearly a controlled- U gate and can therefore be further decomposed, using the methods of Sect. 2.9, into a sequence of 1-qubit gates and a single (fixed) 2-qubit gate such as CNOT. Hence, the set of gates $\mathcal{S} = \{R_x(\alpha), R_y(\beta), R_z(\gamma), Ph(\delta), \text{CNOT}\}$ must be a universal set of gates for quantum computing (and we can even drop one of the rotation gates if we wanted

Table 2.14 Families of gates that are universal for quantum computing

Universal gate family	Meaning	Noteworthy properties
$\{R_x, R_y, R_z, Ph, \text{CNOT}\}$	The union of the set of 1-qubit gates and CNOT is universal	The most widely used set of gates in current quantum circuits
BARENCO(ϕ, α, θ)	A single type of 2-qubit gate is universal	The surprise here is that whereas in classical reversible computing no 2-bit classical reversible gate is universal, in quantum computing almost all 2-qubit gates are universal
$\{H, S, T, \text{CNOT}\}$ where $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ is the Walsh-Hadamard gate, $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ is the “phase gate”, and $T = \begin{pmatrix} 1 & 0 \\ 0 & \exp(i\pi/4) \end{pmatrix}$ is the “ $\pi/8$ gate”	Three fixed-angle 1-qubit gates together with CNOT	The surprise here is that fixed angle gates can form a universal set. In fact, the Solvay-Kitaev theorem [284] implies that any 1-qubit gate can be approximated to accuracy ϵ using $\mathcal{O}(\log^c 1/\epsilon)$ gates from the set $\{H, S, T, \text{CNOT}\}$ where c is a positive constant

to). In fact, the set of all 1-qubit gates and CNOT is the most common set of gates used in constructing practical quantum circuits. Other universal gate sets are known, summarized in Table 2.14, that involve only fixed-angle gates. However, these do not typically lead to efficient quantum circuits due to the need to repeat fixed angle rotations many times to approximate a desired 1-qubit gate to adequate precision. Moreover, even if a given set of gates is universal, and therefore in principle all that is needed to achieve any quantum circuit, in practice, certain computations can be done more efficiently if an “over-complete” family of universal gates is used.

2.7 Special 2-Qubit Gates

The decision to use the set of all 1-qubit gates and CNOT as the universal family of gates, might not be the best choice depending on your type of quantum computing hardware. Different types of quantum computing hardware are associated with different Hamiltonians. So while a CNOT gate (say) may be easy to obtain in one embodiment, it might not be easy in another. For this reason, the next sections describe several different families of 1-qubit and 2-qubit gates that are more “natural” with respect to different types of quantum computing hardware. We give rules for inter-changing between these types of 2-qubit gates so that experimentalists can look at a quantum circuit expressed using one gate family and map it into another, perhaps easier to attain, family.

2.7.1 CSIGN, SWAP $^\alpha$, iSWAP, Berkeley B

The physical interactions available within different types of quantum computer hardware can give rise to different “natural” 2-qubit gates such as iSWAP, SWAP $^\alpha$, CSIGN etc. These are typically easier to achieve than CNOT in the particular physical embodiment, and if maximally entangling, provide no less efficient decompositions of arbitrary 2-qubit operations.

The four most common alternatives to CNOT are shown below:

$$\begin{aligned} \text{CSIGN} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \\ \text{SWAP}^\alpha &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}(1+e^{i\pi\alpha}) & \frac{1}{2}(1-e^{i\pi\alpha}) & 0 \\ 0 & \frac{1}{2}(1-e^{i\pi\alpha}) & \frac{1}{2}(1+e^{i\pi\alpha}) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ \text{iSWAP} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \\ B &= \begin{pmatrix} \cos(\frac{\pi}{8}) & 0 & 0 & i \sin(\frac{\pi}{8}) \\ 0 & \cos(\frac{3\pi}{8}) & i \sin(\frac{3\pi}{8}) & 0 \\ 0 & i \sin(\frac{3\pi}{8}) & \cos(\frac{3\pi}{8}) & 0 \\ i \sin(\frac{\pi}{8}) & 0 & 0 & \cos(\frac{\pi}{8}) \end{pmatrix} \end{aligned} \tag{2.90}$$

Figure 2.33 shows the special icons for some of these gates and summarizes their properties with respect to qubit reversal and their relationship to their own inverse.

2.7.1.1 CSIGN

CSIGN arises naturally in Linear Optical Quantum Computing (LOQC).

$$\text{CSIGN} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{2.91}$$

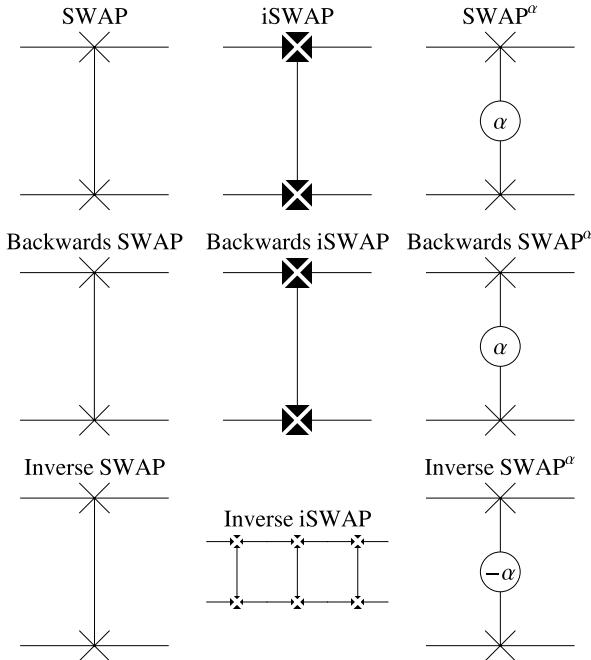


Fig. 2.33 Icons for the special quantum gates SWAP, iSWAP, and SWAP^α . The first row shows the basic gate icon. The second row emphasizes that, unlike CNOT, these gates do not have a preferred “control” qubit and can be inserted “right way up” or “upside down” without it affecting the operation the gate performs. However, whereas CNOT is its own inverse, the same is not true for iSWAP (for which $i\text{SWAP}^\dagger = i\text{SWAP}^{-1} = i\text{SWAP}^3$) and SWAP^α (for which $(\text{SWAP}^\alpha)^\dagger = (\text{SWAP}^\alpha)^{-1} = \text{SWAP}^{-\alpha}$)

2.7.1.2 iSWAP

iSWAP arises naturally in superconducting quantum computing via Hamiltonians implementing the so-called XY model.

$$\text{iSWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.92)$$

2.7.1.3 $\sqrt{\text{SWAP}}$

$\sqrt{\text{SWAP}}$ arises naturally in spintronic quantum computing as that approach employs the “exchange interaction”.

$$\sqrt{\text{SWAP}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} + \frac{i}{2} & \frac{1}{2} - \frac{i}{2} & 0 \\ 0 & \frac{1}{2} - \frac{i}{2} & \frac{1}{2} + \frac{i}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.93)$$

2.7.1.4 SWAP $^\alpha$

SWAP $^\alpha$ also arises naturally in spintronic quantum computing. The duration of the exchange operation determines the exponent achieved in SWAP $^\alpha$.

$$\text{SWAP}^\alpha = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}(1 + e^{i\pi\alpha}) & \frac{1}{2}(1 - e^{i\pi\alpha}) & 0 \\ 0 & \frac{1}{2}(1 - e^{i\pi\alpha}) & \frac{1}{2}(1 + e^{i\pi\alpha}) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.94)$$

2.7.1.5 The Berkeley B Gate

Hamiltonian is $\mathcal{H} = \frac{\pi}{8}(2X \otimes X + Y \otimes Y)$. Gate is $U = \exp(i\mathcal{H})$.

$$\begin{aligned} B &= e^{i\frac{\pi}{8}(2X \otimes X + Y \otimes Y)} \\ &= \begin{pmatrix} \cos(\frac{\pi}{8}) & 0 & 0 & i \sin(\frac{\pi}{8}) \\ 0 & \cos(\frac{3\pi}{8}) & i \sin(\frac{3\pi}{8}) & 0 \\ 0 & i \sin(\frac{3\pi}{8}) & \cos(\frac{3\pi}{8}) & 0 \\ i \sin(\frac{\pi}{8}) & 0 & 0 & \cos(\frac{\pi}{8}) \end{pmatrix} \\ &= \frac{\sqrt{2 - \sqrt{2}}}{2} \begin{pmatrix} 1 + \sqrt{2} & 0 & 0 & i \\ 0 & 1 & i(1 + \sqrt{2}) & 0 \\ 0 & i(1 + \sqrt{2}) & 1 & 0 \\ i & 0 & 0 & 1 + \sqrt{2} \end{pmatrix} \quad (2.95) \end{aligned}$$

2.7.2 Interrelationships Between Types of 2-Qubit Gates

In experimental quantum computing one is faced with having to work with the physical interactions Nature provides. A priori, there is no reason to expect that the most accessible and controllable physical interactions should happen to permit a quantum mechanical evolution that can be interpreted as a CNOT gate. However, if one

looks at the Hamiltonians available in different types of physical systems one can always find 2-qubit gates from which we can, in conjunction with 1-qubit gates, build CNOT gates. In the following sections we give explicit constructions for how to build CNOT gates out of the kinds of 2-body interactions that are commonly available in real physical systems.

2.7.2.1 CNOT from CSIGN

We can obtain a CNOT gate given the ability to achieve 1-qubit gates and CSIGN.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \equiv \left(\mathbb{1}_2 \otimes R_y\left(\frac{\pi}{2}\right) \right) \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \cdot \left(\mathbb{1}_2 \otimes R_y\left(-\frac{\pi}{2}\right) \right) \quad (2.96)$$

An equivalent quantum circuit diagram is shown in Fig. 2.34.

2.7.2.2 CNOT from $\sqrt{\text{SWAP}}$

We can obtain a CNOT gate given the ability to achieve 1-qubit gates and $\sqrt{\text{SWAP}}$.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \equiv \left(R_z\left(-\frac{\pi}{2}\right) \otimes \left(R_y\left(-\frac{\pi}{2}\right) \cdot R_z\left(-\frac{\pi}{2}\right) \right) \right) \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} + \frac{i}{2} & \frac{1}{2} - \frac{i}{2} & 0 \\ 0 & \frac{1}{2} - \frac{i}{2} & \frac{1}{2} + \frac{i}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

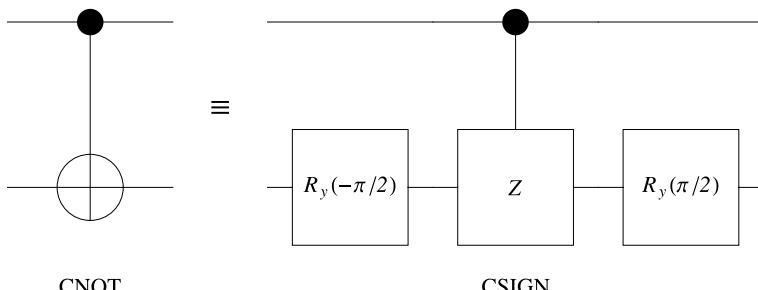


Fig. 2.34 Quantum circuit for obtaining a CNOT gate given the ability to achieve 1-qubit gates and CSIGN

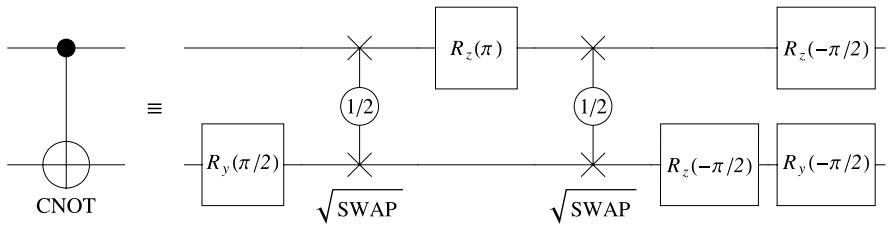


Fig. 2.35 Quantum circuit for obtaining a CNOT gate given the ability to achieve 1-qubit gates and $\sqrt{\text{SWAP}}$

$$\cdot (R_z(\pi) \otimes \mathbb{1}_2) \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} + \frac{i}{2} & \frac{1}{2} - \frac{i}{2} & 0 \\ 0 & \frac{1}{2} - \frac{i}{2} & \frac{1}{2} + \frac{i}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \left(\mathbb{1}_2 \otimes R_y\left(\frac{\pi}{2}\right) \right) \quad (2.97)$$

An equivalent quantum circuit diagram is shown in Fig. 2.35.

2.7.2.3 CNOT from iSWAP and one SWAP

We can obtain a CNOT gate given the ability to achieve 1-qubit gates, iSWAP, and SWAP.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \equiv \left(\mathbb{1}_2 \otimes R_y\left(-\frac{\pi}{2}\right) \right) \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \left(R_z\left(\frac{\pi}{2}\right) \otimes \left(R_z\left(-\frac{\pi}{2}\right) \cdot R_y\left(\frac{\pi}{2}\right) \right) \right) \quad (2.98)$$

An equivalent quantum circuit diagram is shown in Fig. 2.36.

2.7.2.4 CNOT from Two iSWAPs

We can obtain a CNOT gate given the ability to achieve 1-qubit gates and iSWAP.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \equiv \left(\mathbb{1}_2 \otimes \left(R_z\left(\frac{\pi}{2}\right) \cdot R_y\left(-\frac{\pi}{2}\right) \right) \right) \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

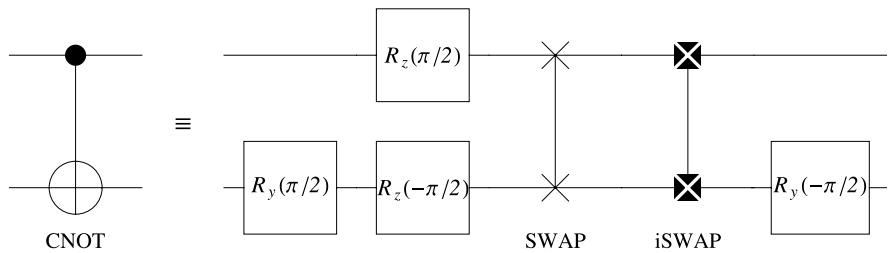


Fig. 2.36 Quantum circuit for obtaining a CNOT gate given the ability to achieve 1-qubit gates, iSWAP, and SWAP

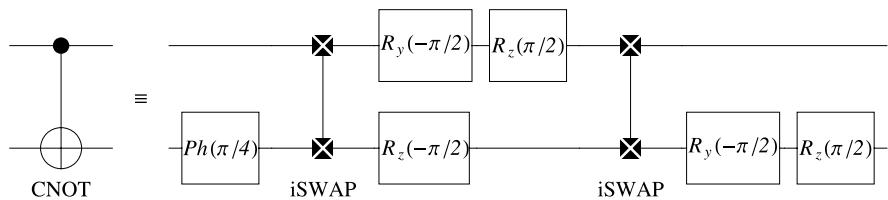


Fig. 2.37 Quantum circuit for obtaining a CNOT gate given the ability to achieve 1-qubit gates and iSWAP

$$\begin{aligned}
 & \cdot \left(\left(R_z\left(\frac{\pi}{2}\right) \cdot R_y\left(-\frac{\pi}{2}\right) \right) \otimes R_z\left(-\frac{\pi}{2}\right) \right) \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 & \cdot \left(\mathbb{1}_2 \otimes Ph\left(\frac{\pi}{4}\right) \right)
 \end{aligned} \tag{2.99}$$

An equivalent quantum circuit diagram is shown in Fig. 2.37.

2.8 Entangling Power of Quantum Gates

A set of qubits is *entangled* if the operations performed on one subset of qubits affect the complementary subset of qubits, even though those qubits are not operated upon directly. For example, imagine partitioning a set of n qubits S into two subsets $\mathcal{A} \subset S$ and $\mathcal{B} = S \setminus \mathcal{A}$. If operations performed on the qubits in \mathcal{A} affect the state of the qubits in \mathcal{B} then there is entanglement between the qubits in \mathcal{A} and those in \mathcal{B} . In such a circumstance, the state of the system cannot be written as the direct product of a state for the qubits in subset \mathcal{A} and a state for the qubits in the complementary subset \mathcal{B} . Such entanglement is unmediated and undiminished by distance and gives rise to so-called “non-local” effects which Einstein dubbed “spooky action at a distance”.

The most striking difference between quantum logic gates and classical logic gates lies in the fact that quantum logic gates can cause the qubits upon which they

act to become more or less entangled, whereas classical gates cannot. In fact, the entire notion of entanglement is absent in classical computing and classical gates can neither entangle nor disentangle the bits upon which they act. Thus entanglement is a quintessentially quantum resource that is only available to quantum computers. Consequently, entanglement is believed to be essential in achieving the exponential speedups seen in quantum algorithms without other computational resources, such as space (memory), time and energy, ballooning exponentially.

Given the apparent importance of entanglement in quantum computing, it is natural to wonder whether all 2-qubit gates are equally good at generating entanglement or whether some are better than others? A little thought should convince you that some 2-qubit gates, such as those built as the direct product of two 1-qubit gates, cannot generate any entanglement whatsoever. But other gates, such as CNOT, seem able to map unentangled inputs into maximally entangled outputs. So clearly there is some variability in the potential for 2-qubit gates to generate entanglement. To make our study precise, however, we need a way to quantify the degree of entanglement within a state, i.e., we need an *entanglement measure*, and we need to define an ensemble of input states over which we would like to average this entanglement measure. Intuitively, if we pick an ensemble of initially unentangled inputs, i.e., product states, then we ought to be able to characterize how effective a given gate is at generating entanglement by seeing how entangled, on average, its outputs will be given it received initially unentangled inputs. This is the essential idea between the notion of the “entangling power” of a quantum gate. Intuitively, the more the output is entangled, the greater the entangling power of the gate.

2.8.1 “Tangle” as a Measure of the Entanglement Within a State

It turns out that there are many ways one could characterize the degree of entanglement within a 2-qubit quantum state. Fortunately, in the case of 2-qubit states, all the different entanglement measures turn out to be equivalent to one another. However, no such equivalence is found for entanglement measures of n -qubit states and attempts to find a unifying entanglement measure for n -qubit states have been fraught with difficulties spawning a cottage industry of “entanglement monotones” on which many Ph.D. theses have been written. For us, however, here we are concerned only with the entangling power of 2-qubit gates, and so any of the equivalent 2-qubit entanglement measures will serve us equally well.

Specifically, the *tangle* provides a quantitative measure of the degree of entanglement within a quantum state. Formally, the *tangle* is the square of the concurrence, which for a 2-qubit pure state, $|\psi\rangle$, is given by:

$$\text{Concurrence}(|\psi\rangle) = |\langle\psi|\tilde{\psi}\rangle| \quad (2.100)$$

where $|\tilde{\psi}\rangle$ is the spin-flipped version of $|\psi\rangle$. This is defined as $|\tilde{\psi}\rangle = (Y \otimes Y)|\psi^*\rangle$, where Y is the Pauli- Y matrix, and $|\psi^*\rangle$ is $|\psi\rangle$ with its amplitudes complex conjugated. Thus, if $|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$, then $|\psi^*\rangle = a^*|00\rangle + b^*|01\rangle +$

$c^*|10\rangle + d^*|11\rangle$ and $|\tilde{\psi}\rangle = -d^*|00\rangle + c^*|01\rangle + b^*|10\rangle - a^*|11\rangle$. Hence, the concurrence of a general 2-qubit state $|\psi\rangle$ is given by:

$$\text{Concurrence}(a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle) = |2b^*c^* - 2a^*d^*| \quad (2.101)$$

The “spin-flip” transformation maps the state of each component qubit into its orthogonal state. Hence the spin-flip transformation is not unitary and cannot, therefore, be performed deterministically by any isolated quantum system. So there can be no such thing as a perfect spin-flip “gate” as such. (If there were it would be a universal NOT gate.) Nevertheless, the spin-flip transformation is a perfectly legitimate mathematical specification of a transformation. One of the properties of the spin-flip transformation is that, if the 2-qubit state $|\psi\rangle$ happens to be a product state (i.e., an unentangled state) its spin-flipped version, $|\tilde{\psi}\rangle$, will be orthogonal to $|\psi\rangle$. Hence, the overlap $\langle\psi|\tilde{\psi}\rangle$ will be zero and hence the concurrence of state $|\psi\rangle$ will be zero. So unentangled states have a concurrence of zero.

At the other extreme, under the spin-flip transformation maximally entangled states, such as Bell states, remain invariant up to an unimportant overall phase. To see this, the four Bell states are given by: Bell states

$$\begin{aligned} |\beta_{00}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\beta_{01}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\beta_{10}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\beta_{11}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \end{aligned} \quad (2.102)$$

Under the spin-flip transformation these states transform, respectively, into:

$$\begin{array}{ccc} |\beta_{00}\rangle & \xrightarrow{\text{spin-flip}} & -|\beta_{00}\rangle \\ |\beta_{01}\rangle & \xrightarrow{\text{spin-flip}} & |\beta_{01}\rangle \\ |\beta_{10}\rangle & \xrightarrow{\text{spin-flip}} & |\beta_{10}\rangle \\ |\beta_{11}\rangle & \xrightarrow{\text{spin-flip}} & -|\beta_{11}\rangle \end{array} \quad (2.103)$$

Hence, the overlap between a maximally entangled state and its spin-flipped counterpart is unity, which is the most it can be, implying that maximally entangled states have a concurrence of one.

Thus the tangle, as defined above, provides a quantitative measure for the degree of entanglement within a pure 2-qubit state. Generalizations of tangle to mixed states and multi-partite states are discussed in Chap. 11.

2.8.2 “Entangling Power” as the Mean Tangle Generated by a Gate

Having quantified the degree of entanglement within a state, it becomes possible to quantify the degree to which different gates generate entanglement when acting upon initially unentangled inputs. Specifically we can define the *entangling power* of a gate as follows [559]:

Entangling Power The entangling power of a 2-qubit gate U , $\text{EP}(U)$, is the mean tangle that U generates averaged over all input product state inputs sampled uniformly on the Bloch sphere.

Mathematically this is expressed as:

$$\text{EP}(U) = \langle E(U|\psi_1\rangle \otimes |\psi_2\rangle) \rangle_{|\psi_1\rangle, |\psi_2\rangle} \quad (2.104)$$

where $E(\cdot)$ is the tangle of any other 2-qubit entanglement measure such as the linear entropy (as all the 2-qubit entanglement measures are equivalent to one another), and $|\psi_1\rangle$ and $|\psi_2\rangle$ are single qubit states sampled uniformly on the Bloch sphere.

Although formally correct, the definition of entangling power given in (2.104) is not an easy thing to compute. However, since we have fixed the probability distribution over which the samples $|\psi_1\rangle$ and $|\psi_2\rangle$ are to be taken to be the uniform distribution on the surface of the Bloch sphere, we can build this assumption into the definition of entangling power and derive a more explicit, and effectively computable, formula for entangling power.

Let's begin by writing the arbitrary pure states $|\psi_1\rangle$ and $|\psi_2\rangle$ as:

$$|\psi_1\rangle = \cos\left(\frac{\theta_1}{2}\right)|0\rangle + e^{i\phi_1} \sin\left(\frac{\theta_1}{2}\right)|1\rangle \quad (2.105)$$

$$|\psi_2\rangle = \cos\left(\frac{\theta_2}{2}\right)|0\rangle + e^{i\phi_2} \sin\left(\frac{\theta_2}{2}\right)|1\rangle \quad (2.106)$$

For state $|\psi_1\rangle$, θ_1 is the angle between the z -axis and the state vector, and ϕ_1 is the angle around the z -axis in the x - y plane. Hence, as we desire to compute an average over the product of such states sampled *uniformly* over the Bloch sphere, we need to weight the contributions depending on the values of θ_1 and θ_2 . Otherwise, the samples would be biased towards product states in the vicinity of the poles. To see this imagine that the density of states around the circumference of the Bloch sphere in the x - y plane is N states in a distance $2\pi R$, where R is the radius of the Bloch sphere, so the density of states at the equator is $N/(2\pi R)$. As we ascend the z -axis, to be unbiased, we still want to sample points around the circumference of a plane parallel to the x - y plane at height z at the same density. Hence we require $n/(2\pi r) = N/(2\pi R)$ which implies $n/N = r/R = \sin(\theta_1)$. Thus we must dilute states by a factor of $\sin(\theta_1)$ as we ascend the z -axis to maintain constant density.

Likewise for $|\psi_2\rangle$, giving an overall weighting function of $\sin(\theta_1) \sin(\theta_2)$. Hence, we have:

$$\begin{aligned} \text{EP}(U) &= \langle E(U|\psi_1\rangle \otimes |\psi_2\rangle) \rangle_{|\psi_1\rangle, |\psi_2\rangle} \\ &= 2\text{tr}\left((U \otimes U) \cdot \mathcal{Q}_p \cdot (U^\dagger \otimes U^\dagger) \cdot \frac{1}{2}(\mathbb{1}_{16} - \text{SWAP}_{1,3;4})\right) \end{aligned} \quad (2.107)$$

where $\mathbb{1}_{16}$ is the 16×16 identity matrix, and $\text{SWAP}_{i,j;k}$ is the operator that swaps the i -th and j -th of k qubits.

$$\mathcal{Q}_p = \frac{1}{16\pi^2} \int_0^{2\pi} \int_0^{2\pi} \int_0^\pi \int_0^\pi \sin(\theta_1) \sin(\theta_2) (|\psi_1\rangle \langle \psi_1| \otimes |\psi_2\rangle \langle \psi_2|)^{\otimes 2} d\theta_2 d\theta_1 d\phi_2 d\phi_1 \quad (2.108)$$

and the normalization factor $1/(16\pi^2)$ comes from the average of the weighting function:

$$\int_0^{2\pi} \int_0^{2\pi} \int_0^\pi \int_0^\pi \sin(\theta_1) \sin(\theta_2) d\theta_2 d\theta_1 d\phi_2 d\phi_1 = 16\pi^2 \quad (2.109)$$

With these definitions, \mathcal{Q}_p evaluates to the matrix

$$\mathcal{Q}_p = \begin{pmatrix} \frac{1}{9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{18} & 0 & 0 & \frac{1}{18} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{18} & 0 & 0 & 0 & 0 & 0 & \frac{1}{18} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{36} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{18} & 0 & 0 & \frac{1}{18} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{36} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{18} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{18} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{18} & 0 & 0 & 0 & 0 & 0 & \frac{1}{18} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{36} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{9} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{18} & 0 & 0 & \frac{1}{18} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{36} & 0 & 0 & \frac{1}{36} & 0 & 0 & 0 & \frac{1}{36} & 0 & 0 & \frac{1}{36} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{18} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{18} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{18} & 0 & 0 & 0 & \frac{1}{18} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{18} & 0 & 0 & 0 & \frac{1}{18} \end{pmatrix} \quad (2.110)$$

Table 2.15 Entangling power of some common 2-qubit gates. Here $\mathbb{1}_2 \oplus U$ is a controlled gate with U defined as $U = R_x(a) \cdot R_y(b) \cdot R_z(c) \cdot Ph(d)$, and $\mathbb{1}_2 \oplus V$ is a controlled gate with V defined as $V = R_z(a) \cdot R_y(b) \cdot R_z(c) \cdot Ph(d)$. Notice that there can be no angle α that would make the $SWAP^\alpha$ a maximally entangling gate

U	$EP(U)$
$U \otimes V$	0
CNOT	$\frac{2}{9}$
iSWAP	$\frac{2}{9}$
B	$\frac{2}{9}$
SWAP	0
\sqrt{SWAP}	$\frac{1}{6}$
$SWAP^\alpha$	$\frac{1}{6} \sin^2(\pi\alpha)$
$R_x(a) \oplus R_x(b)$	$\frac{1}{9}(1 - \cos(a - b))$
$R_x(a) \oplus R_y(b)$	$\frac{1}{18}(-\cos(b) - \cos(a)(\cos(b) + 1) + 3)$
$R_x(a) \oplus R_z(b)$	$\frac{1}{18}(-\cos(b) - \cos(a)(\cos(b) + 1) + 3)$
$R_y(a) \oplus R_x(b)$	$\frac{1}{18}(-\cos(b) - \cos(a)(\cos(b) + 1) + 3)$
$R_y(a) \oplus R_y(b)$	$\frac{1}{9}(1 - \cos(a - b))$
$R_y(a) \oplus R_z(b)$	$\frac{1}{18}(-\cos(b) - \cos(a)(\cos(b) + 1) + 3)$
$R_z(a) \oplus R_x(b)$	$\frac{1}{18}(-\cos(b) - \cos(a)(\cos(b) + 1) + 3)$
$R_z(a) \oplus R_y(b)$	$\frac{1}{18}(-\cos(b) - \cos(a)(\cos(b) + 1) + 3)$
$R_z(a) \oplus R_z(b)$	$\frac{1}{9}(1 - \cos(a - b))$
$\mathbb{1}_2 \oplus U$	$\frac{1}{6} + \frac{1}{18}(\sin(a)\sin(b)\sin(c) - \cos(a)\cos(b) - \cos(c)\cos(b) - \cos(a)\cos(c))$
$\mathbb{1}_2 \oplus V$	$\frac{1}{6} - \frac{1}{18}(\cos(a+c)\cos(b) + \cos(b) + \cos(a+c))$

Although it is non-obvious, an equivalent way to compute $EP(U)$ is from the formula:

$$\begin{aligned}
 EP(U) = & \frac{5}{9} - \frac{1}{36} [\text{tr}((U \otimes U)^\dagger \cdot SWAP_{1,3;4} \cdot (U \otimes U) \cdot SWAP_{1,3;4}) \\
 & + \text{tr}((SWAP_{1,2;2} \cdot U \otimes SWAP_{1,2;2} \cdot U)^\dagger \cdot SWAP_{1,3;4} \\
 & \cdot (SWAP_{1,2;2} \cdot U \otimes SWAP_{1,2;2} \cdot U) \cdot SWAP_{1,3;4})] \quad (2.111)
 \end{aligned}$$

The entangling power of a gate ranges from 0 for non-entangling gates (such as SWAP), to $\frac{2}{9}$ for maximally entangling gates (such as CNOT, iSWAP, and Berkeley B). Other gates, such as \sqrt{SWAP} and more generally $SWAP^\alpha$, have intermediate values of entangling power. Table 2.15 lists the entangling powers for some common types of 2-qubit gates. Typically, the entangling powers of parameterized gates, such as $SWAP^\alpha$ and $R_y(a) \oplus R_y(b)$, varies with the parameter values used.

2.8.3 CNOT from any Maximally Entangling Gate

In experimental quantum computing, one often needs to find a way to obtain a CNOT gate from whatever physically realizable 2-qubit interaction, is available. It turns out that the ease with which a CNOT can be obtained from the physically available 2-qubit gate, U , is intimately connected to the entangling power of U . In particular, if $\text{EP}(U) = \frac{2}{9}$, i.e., maximal, but U is itself not a CNOT gate, then we can always create a CNOT gate from just *two* calls to U via a decomposition of the form:

$$\text{CNOT} \equiv (A_1 \otimes A_2) \cdot U \cdot (H \otimes \mathbb{1}_2) \cdot U \quad (2.112)$$

where H is the Hadamard gate and A_1 and A_2 are 1-qubit gates.

This result is of practical importance to experimentalists since it may not always possible to achieve a CNOT gate directly from whatever interaction Hamiltonian is attainable within some physical context. Nevertheless, this result shows that once it is understood how a maximally entangling operation can be achieved from the available interaction Hamiltonians, then we can use it, in conjunction with 1-qubit gates, to achieve a CNOT.

2.8.4 The Magic Basis and Its Effect on Entangling Power

As you might recall, a quantum gate with unitary matrix U in the computational basis can be viewed as the matrix $V \cdot U \cdot V^\dagger$ in the “ V -basis”. In the case of 2-qubit gates there is a special basis, called the *magic basis*, that turns out to several have remarkable properties [54, 232, 296].

The “magic basis” is a set of 2-qubit states that are phase shifted versions of the Bell states. In particular, we have:

$$|00\rangle \xrightarrow{\mathcal{M}} |\mathcal{M}_1\rangle = |\beta_{00}\rangle \quad (2.113)$$

$$|01\rangle \xrightarrow{\mathcal{M}} |\mathcal{M}_2\rangle = i|\beta_{10}\rangle \quad (2.114)$$

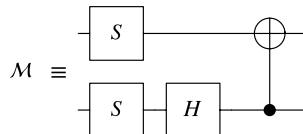
$$|10\rangle \xrightarrow{\mathcal{M}} |\mathcal{M}_3\rangle = i|\beta_{01}\rangle \quad (2.115)$$

$$|11\rangle \xrightarrow{\mathcal{M}} |\mathcal{M}_4\rangle = |\beta_{11}\rangle \quad (2.116)$$

where $|\beta_{00}\rangle$, $|\beta_{01}\rangle$, $|\beta_{10}\rangle$, and $|\beta_{11}\rangle$ are the Bell states defined by:

$$\begin{aligned} |\beta_{00}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\beta_{01}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\beta_{10}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\beta_{11}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \end{aligned} \quad (2.117)$$

Fig. 2.38 Quantum circuit that implements the magic basis transformation. Here $S = Ph(\frac{\pi}{4}) \cdot R_z(\frac{\pi}{2})$ and $H = Z \cdot R_y(-\frac{\pi}{2})$



Thus, the matrix, \mathcal{M} , which maps the computational basis into the “magic” basis is:

$$\begin{aligned} \mathcal{M} &= |\mathcal{M}_1\rangle\langle 00| + |\mathcal{M}_2\rangle\langle 01| + |\mathcal{M}_3\rangle\langle 10| + |\mathcal{M}_4\rangle\langle 11| \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i & 0 & 0 \\ 0 & 0 & i & 1 \\ 0 & 0 & i & -1 \\ 1 & -i & 0 & 0 \end{pmatrix} \end{aligned} \quad (2.118)$$

The reason this basis is called the “magic basis” is because it turns out that any partially or maximally entangling 2-qubit gate, described by a *purely real* unitary matrix, U , becomes an *non-entangling* gate in the “magic” basis. In other words, no matter how entangling U may be, $\mathcal{M} \cdot U \cdot \mathcal{M}^\dagger$ is always a non-entangling gate, and hence $EP(\mathcal{M} \cdot U \cdot \mathcal{M}^\dagger) = 0$.

We can make use of this observation in order to find a circuit for any 2-qubit gate described by a purely real unitary matrix, U , because either $\mathcal{M} \cdot U \cdot \mathcal{M}^\dagger = A \otimes B$ (one kind of non-entangling circuit) or else is related to a single SWAP gate (another non-entangling gate). And it is pretty easy to spot which is the case. Therefore, if we know the simplest quantum circuit implementing the magic basis transformation, we can then invert $\mathcal{M} \cdot U \cdot \mathcal{M}^\dagger = A \otimes B$ (or the similar one involving SWAP) to find a circuit for U . Luckily, it is easy to find a quantum circuit for the magic basis transformation. A simple quantum circuit that achieves the magic basis gate is shown in Fig. 2.38.

If that was not magical enough, we can also use the magic basis transformation to relate a given purely real unitary, via a mathematical procedure involving \mathcal{M} , to a gate that is guaranteed to be *maximally* entangling! Specifically, for any purely real 4×4 unitary matrix, U , then, regardless of its entangling power, the entangling power of the gate defined by $\mathcal{M} \cdot U \cdot \mathcal{M}$ is maximal, i.e., $\frac{2}{9}$. Amazing!

2.9 Arbitrary 2-Qubit Gates: The Krauss-Cirac Decomposition

Given that qubit-qubit interactions are essential to performing non-trivial quantum computations, it is important to understand how an *arbitrary* 2-qubit gate can be decomposed into more elementary gates such as CNOTs and 1-qubit gates. A priori it is not at all obvious how many CNOTs we will need. As we shall see the answer depends on the structure of the 2-qubit gate in question, but in no case do we ever need to use more than three CNOT gates [90, 452, 512, 517].

The key to finding a general circuit that can implement *any* 2-qubit gate is to use the magic basis transformation in conjunction with a factorization of an arbitrary

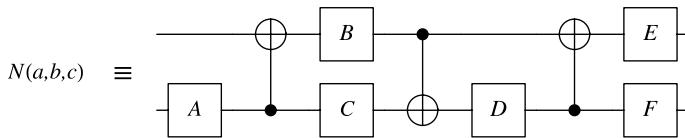


Fig. 2.39 Quantum circuit for the core entangling gate $N(a, b, c)$. Here $A = R_z(-\frac{\pi}{2})$, $B = R_z(\frac{\pi}{2} - 2c)$, $C = R_y(2a - \frac{\pi}{2})$, $D = R_y(\frac{\pi}{2} - 2b)$, $E = R_z(\frac{\pi}{2})$, and $F = Ph(\frac{\pi}{4})$

2-qubit gate discovered by Krauss and Cirac. Krauss and Cirac found that any 4×4 unitary matrix can be factored into the form:

$$U \equiv (A_1 \otimes A_2) \cdot e^{i(aX \otimes X + bY \otimes Y + cZ \otimes Z)} \cdot (A_3 \otimes A_4) \quad (2.119)$$

where X , Y , and Z are the three Pauli matrices, and $e^M = \mathbb{1} + M + \frac{1}{2!}(M \cdot M) + \frac{1}{3!}(M \cdot M \cdot M) + \frac{1}{3!}(M \cdot M \cdot M \cdot M) + \dots$ is the *matrix exponential*⁵ and $a, b, c \in \mathbb{R}$ [277, 296, 562]. Since we already know how to find quantum circuits for any 1-qubit gate, we can always find decompositions for the A_j whatever they may happen to be. We also know that the 1-qubit gates cannot change the entangling power of the core 2-qubit gate $N(a, b, c)$. So all the action is really concentrated in the 2-qubit gate $N(a, b, c)$, which is equivalent to the following unitary matrix:

$$N(a, b, c)$$

$$\equiv \begin{pmatrix} e^{ic} \cos(a-b) & 0 & 0 & ie^{ic} \sin(a-b) \\ 0 & e^{-ic} \cos(a+b) & ie^{-ic} \sin(a+b) & 0 \\ 0 & ie^{-ic} \sin(a+b) & e^{-ic} \cos(a+b) & 0 \\ ie^{ic} \sin(a-b) & 0 & 0 & e^{ic} \cos(a-b) \end{pmatrix} \quad (2.120)$$

A quantum circuit for $N(a, b, c)$ is shown in Fig. 2.39. Algebraically, we have: $N(a, b, c) = (E \otimes F) \cdot \text{CNOT}_{2,1;2} \cdot (\mathbb{1} \otimes D) \cdot \text{CNOT}_{1,2;2} \cdot (B \otimes C) \cdot \text{CNOT}_{2,1;2} \cdot (\mathbb{1} \otimes A)$ where $A = R_z(-\frac{\pi}{2})$, $B = R_z(\frac{\pi}{2} - 2c)$, $C = R_y(2a - \frac{\pi}{2})$, $D = R_y(\frac{\pi}{2} - 2b)$, $E = R_z(\frac{\pi}{2})$, and $F = Ph(\frac{\pi}{4})$.

The matrix, U , corresponding to any 2-qubit quantum gate is always unitary, and the *magnitude* of its determinant is always unity, i.e., $|\det(U)| = 1$. However, the ease with which we can implement U depends upon whether its elements are real or complex and whether its determinant is $+1$ or one of the other possibilities, consistent with $|\det(U)| = 1$, namely -1 , $+i$, or $-i$. We classify the possibilities as follows:

1. $U \in \mathbf{SU}(2^n)$ implies U is a $2^n \times 2^n$ dimensional special unitary matrix containing real or complex elements and having a determinant $|\det(U)| = 1$, i.e., $\det(U) = \pm 1$ or $\pm i$.

⁵N.B. the leading “1” in the series expansion of the exponential function is replaced with the identity matrix, $\mathbb{1}$.

2. $U \in \mathbf{U}(2^n)$ implies U is a $2^n \times 2^n$ dimensional unitary matrix containing real or complex elements and having a determinant $|\det(U)| = 1$, i.e., $\det(U) = \pm 1$ or $\pm i$.
3. $U \in \mathbf{SO}(2^n)$ implies U is a $2^n \times 2^n$ dimensional special unitary matrix containing only real elements and having a determinant $\det(U) = +1$.
4. $U \in \mathbf{O}(2^n)$ implies U is a $2^n \times 2^n$ dimensional unitary matrix containing only real elements and having a determinant $\det(U) = \pm 1$.

The number of CNOT gates needed to implement U depends upon which the class into which U falls.

Using the upside down CNOT, we can write a circuit that implements the core entangling gate $N(a, b, c)$:

$$\begin{aligned}
N(a, b, c) &\equiv \left(R_z\left(\frac{\pi}{2}\right) \otimes Ph\left(\frac{\pi}{4}\right) \right) \\
&\cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot \left(\mathbb{1}_2 \otimes R_y\left(\frac{\pi}{2} - 2b\right) \right) \\
&\cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \left(R_z\left(\frac{\pi}{2} - 2c\right) \otimes R_y\left(2a - \frac{\pi}{2}\right) \right) \\
&\cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot \left(\mathbb{1}_2 \otimes R_z\left(-\frac{\pi}{2}\right) \right)
\end{aligned} \tag{2.121}$$

2.9.1 Entangling Power of an Arbitrary 2-Qubit Gate

An arbitrary 2-qubit gate, U , can be factored according to the Krauss-Cirac decomposition as $U = (A_1 \otimes A_2) \cdot N(a, b, c) \cdot (A_3 \otimes A_4)$, where the A_j are 1-qubit gates, and $N(a, b, c) = \exp(i(aX \otimes X + bY \otimes Y + cZ \otimes Z))$ is the core entangling operation. As the entangling power of any gate is not affected by 1-qubit operations, the entangling power of an arbitrary 2-qubit gate must be determined entirely by the entangling power of its core factor $N(a, b, c)$. Using the formulae given earlier, we can calculate the entangling power of $N(a, b, c)$. In particular, one finds $EP(N(a, b, c))$ is given by:

$$\begin{aligned}
EP(N(a, b, c)) &= -\frac{1}{18} \cos(4a) \cos(4b) - \frac{1}{18} \cos(4c) \cos(4b) \\
&\quad - \frac{1}{18} \cos(4a) \cos(4c) + \frac{1}{6}
\end{aligned} \tag{2.122}$$

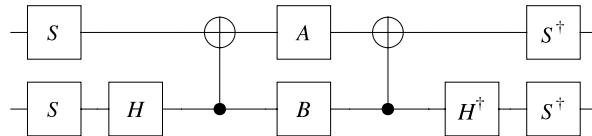


Fig. 2.40 Quantum circuit sufficient to implement any 2-qubit gate $U \in \mathbf{SO}(4)$. The unitary matrix for such a gate is purely real and has a determinant of +1

Notice that this immediately gives us a way of proving that the greatest entangling power of any 2-qubit gate is the largest value that $\text{EP}(N(a, b, c))$ can assume, namely, $\frac{2}{9}$. The CNOT, iSWAP, and Berkeley B gates introduced earlier are all maximally entangling gates in this sense. However, the SWAP^α gate is not a maximally entangling gate.

2.9.2 Circuit for an Arbitrary Real 2-Qubit Gate

2.9.2.1 Case of $U \in \mathbf{SO}(4)$

If $U \in \mathbf{SO}(4)$ then the elements of U are purely real numbers and $\det(U) = +1$.

Theorem 2.1 *In the magic basis, \mathcal{M} , any purely real special unitary matrix $U \in \mathbf{SO}(4)$, can be factored as the tensor product of two special unitary matrices, i.e., we always have $\mathcal{M} \cdot U \cdot \mathcal{M}^\dagger = A \otimes B$ where $A, B \in \mathbf{SU}(2)$.*

A quantum circuit implementing the magic basis transformation (2.118) was shown in Fig. 2.38. Therefore, every 2-qubit quantum gate in $\mathbf{SO}(4)$ can be realized in a circuit consisting of 12 elementary 1-qubit gates and two CNOT gates (see Fig. 2.40).

2.9.2.2 Case of $U \in \mathbf{O}(4)$

If $U \in \mathbf{O}(4)$ then the elements of U are purely real numbers and $\det(U) = \pm 1$.

Theorem 2.2 *In the magic basis, \mathcal{M} , any purely real unitary matrix $U \in \mathbf{O}(4)$ with $\det(U) = -1$, can be factored as the tensor product of two special unitary matrices, i.e., we always have $\mathcal{M} \cdot U \cdot \mathcal{M}^\dagger = (A \otimes B) \cdot \text{SWAP} \cdot (\mathbb{1} \otimes Z)$ where $A, B \in \mathbf{U}(2)$ and Z is the Pauli-Z matrix.*

Every 2-qubit quantum gate in $\mathbf{O}(4)$ with $\det(U) = -1$ can be realized in a circuit consisting of 12 elementary gates, two CNOT gates, and one SWAP gate (see Fig. 2.41). As you will show in Exercise 2.29 this circuit can be simplified further to one involving at most three CNOT gates.

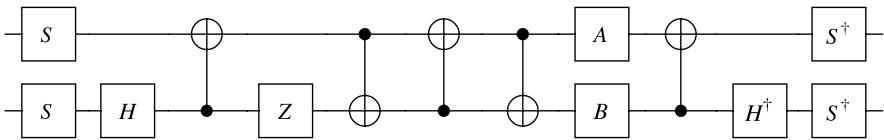


Fig. 2.41 Quantum circuit sufficient to implement any 2-qubit gate $U \in \mathbf{O}(4)$. The unitary matrix for such a gate is purely real and has a determinant of ± 1 . Those gates having a determinant of $+1$ can be implemented using at most two CNOT gates. Those having a determinant of -1 can be implemented in a circuit of the form shown. In Exercise 2.29 you will simplify this circuit further to show that an arbitrary 2-qubit gate $U \in \mathbf{O}(4)$ requires at most three CNOT gates

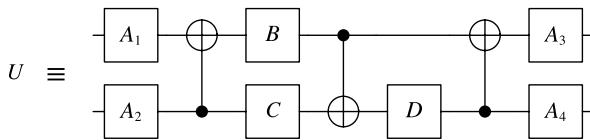


Fig. 2.42 Quantum circuit for an arbitrary 2-qubit gate, U . By the Krauss-Cirac decomposition U can be written in the form $(A_1 \otimes A_2) \cdot N(a, b, c) \cdot (A_3 \otimes A_4)$. As in the quantum circuit for $N(a, b, c)$, $B = R_z(\frac{\pi}{2} - 2c)$, $C = R_y(2a - \frac{\pi}{2})$, $D = R_y(\frac{\pi}{2} - 2b)$. The leftmost and rightmost single qubit gates needed to obtain $N(a, b, c)$ can be absorbed into the single qubit gates A_1, A_2, A_3, A_4

2.9.3 Circuit for an Arbitrary Complex 2-Qubit Gate

An arbitrary 2-qubit gate SWAP^α and can therefore have elements whose values are complex numbers. Every 2-qubit quantum gate in $\mathbf{U}(4)$ can be realized, up to an overall global phase factor, in a circuit consisting of 15 elementary 1-qubit gates, three CNOT gates (see Fig. 2.42).

2.9.4 Circuit for an Arbitrary 2-Qubit Gate Using SWAP^α

Ponder for a moment whether you would expect the quantum circuit for an arbitrary 2-qubit using the $\text{CNOT} \cup 1\text{-qubit gates}$ family to require more, less, or the same number of 2-qubit gates than the equivalent circuits based on different a gate family relying on a less than maximally entangling gate, such as SWAP^α . Since a general 2-qubit gate needs three CNOTs (and CNOT is a maximally entangling gate) one might expect that one needs more than three SWAP^α gates to implement a general 2-qubit gate. Surprisingly, this is not the case! In fact, three SWAP^α gates, having three different values for the exponents, are sufficient. The proof is by explicit

construction of the central entangling gate of the Krauss-Cirac decomposition:

$$\begin{aligned}
 N(a, b, c) \equiv & (Ph(a + b - c) \otimes \mathbb{1}_2) \cdot \left(R_z\left(\frac{\pi}{2}\right) \otimes R_z\left(-\frac{\pi}{2}\right) \cdot R_y(\pi) \right) \\
 & \cdot \text{SWAP}^{1-\frac{2(b-c)}{\pi}} \cdot (R_y(\pi) \cdot R_z(-\pi) \otimes R_y(\pi)) \\
 & \cdot \text{SWAP}^{\frac{2(c-a)}{\pi}} \cdot (R_z(\pi) \otimes R_y(\pi) \cdot R_z(-\pi)) \\
 & \cdot \text{SWAP}^{1-\frac{2(a+b)}{\pi}} \cdot \left(R_z\left(\frac{\pi}{2}\right) \otimes R_z\left(\frac{\pi}{2}\right) \right)
 \end{aligned} \tag{2.123}$$

2.10 Summary

Quantum gates are not always to be thought of in the same way we picture classical gates. In a conventional electronic circuits we are used to thinking of bits passing through logic gates. In quantum circuits this notion may or may not be accurate depending on how qubits are actually encoded within the physical system. If one is using photons to encode qubits and optical elements (such as beam-splitters or phase shifters) to perform gate operations, then the qubits are quite literally moving through the quantum gates. However, if we are using say trapped ions to encode the qubits, the logical state of the qubits is encoded within the internal excitation state of the ions, and the ions are held more or less in place. This distinction illustrates that a quantum gate is really nothing more than a deliberate manipulation of a quantum state.

In this chapter we introduced the idea of a quantum gate, and contrasted it with logically irreversible and logically reversible classical gates. Quantum gates are, like classical reversible gates, logically reversible, but they differ markedly on their universality properties. Whereas the smallest universal classical reversible gates have to use three bits, the smallest universal quantum gates need only use two bits. As the classical reversible gates are also unitary, it is conceivable that one of the first practical applications of quantum gates is in non-standard (e.g., “spintronic”) implementations of classical reversible computers.

We described some of the more popular quantum gates and why they are useful, explained how these gates can be achieved via the natural evolution of certain quantum systems, and discussed quantum analogs of controlled and universal gates. Controlled gates are key to achieving non-trivial computations, and universal gates are key to achieving practical hardware.

In the theory of classical computing you would interpret a controlled gate operation as implying that you *read* (i.e., measure) the control bit and, depending on its value, perform the appropriate action on the target bits. However, such explicit measurement operations on the control qubits are neither implied nor necessary in *quantum* controlled gates. Instead, the controlled quantum gates

apply *all* the control actions consistent with the quantum state of the control qubits.

We showed that there are several 2-qubit gates that are as powerful as the CNOT gate when used in conjunction with 1-qubit gates, and gave explicit interconversions between these types of gates. Such alternatives to CNOT gates may be easier to achieve than CNOT in specific schemes for quantum computing hardware. For example, iSWAP, SWAP α , and CSIGN are more naturally suited to superconducting, spintronic, and optical quantum computers than CNOT.

We introduced the “tangle” as a way of quantifying the entanglement within a quantum state, and used it to define the “entangling power” of a quantum gate. We also introduced the magic basis and demonstrated its effects on entangling power.

We ended the chapter with exact minimal quantum circuits sufficient to implement an arbitrary 2-qubit gate and gave an analytic scheme for converting a given unitary matrix into a minimal 2-qubit circuit.

2.11 Exercises

2.1 Which of the following matrices are unitary?

1.
$$\begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

2.
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

3.
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

4.
$$\begin{pmatrix} 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ i & 0 & 0 & 0 \\ 0 & 0 & 0 & i \end{pmatrix}$$

Which of those could describe quantum gates that act on qubits? Explain your answer.

2.2 What is the output state from the quantum circuit shown in Fig. 2.43.

2.3 How would a CNOT gate transform an entangled input state of the form $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$? Are the qubits still entangled after the CNOT has been applied? Explain your answer by making reference to the definition of an entangled state.

2.4 Show that X does not negate a general quantum state $|\psi\rangle = \cos(\frac{\theta}{2})|0\rangle + \exp(i\phi)\sin(\frac{\theta}{2})|1\rangle$.

2.5 Given a qubit whose state is known to lie in the equatorial x - y plane in the Bloch sphere is it possible to find a quantum gate that will always negate this qubit? If so, exhibit such a gate. If not, explain why it is impossible.

2.6 The circuit for controlled-controlled- U that was given earlier in this chapter assumed the existence of a controlled- V gate defined such that $V^2 = U$ with V unitary. Prove, for any unitary matrix U , that such a V always exists, i.e. that there exists a unitary matrix V such that $V^2 = U$.

2.7 Decompose the Hadamard gate, $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, in terms of $R_y(\theta)$ and $R_z(\phi)$ gates.

2.8 The “magic” basis is defined by the matrix . . .

2.9 Given real numbers x , y , and z and the Pauli matrices defined as

$$\mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (2.124)$$

prove the identity

$$e^{i(xX+yY+zZ)} = \cos(r)\mathbb{1} + \frac{\sin(r)}{r}i(xX + yY + zZ) \quad (2.125)$$

where $r = \sqrt{x^2 + y^2 + z^2}$. You might find the following identities to be useful: $\cos(\alpha) = \cosh(i\alpha)$ and $\sin(\beta) = -i \sinh(i\beta)$, and $i\sqrt{x^2 + y^2 + z^2} = \sqrt{-x^2 - y^2 - z^2}$.

2.10 Prove any 2×2 hermitian matrix can be written as a sum of Pauli matrices. This shows that any 1-qubit Hamiltonian can be expressed in terms of just Pauli matrices.

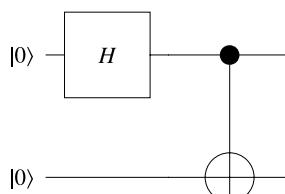


Fig. 2.43 This quantum circuit applies a Hadamard gate to the first qubit followed by a CNOT gate to both qubits

2.11 Show that a state, $|\psi\rangle$, is orthogonal to its antipodal state, $|\psi^\perp\rangle$, i.e., show $\langle\psi|\psi^\perp\rangle = 0$.

2.12 Prove that $R_x(\alpha)$ and $R_y(\alpha)$ rotate a general single qubit pure state about the x - and y -axes respectively through angle α .

2.13 Show that the NOR gate defined by $a \text{ NOR } b \equiv \neg(a \vee b)$ is, like the NAND gate, also universal for classical irreversible computing. [Hint: Show that any logical proposition can be written in terms of just \neg and \vee , and that both \neg and \vee can be expressed using only NOR gates.]

2.14 One of the most fundamental tasks we might imagine a computer doing is to decide whether two items in memory are the same and, if so, to output TRUE and, if not, to output FALSE. If we imagine the items in memory are represented by bit strings, our task becomes one of determining whether two bit strings are the same. Show that you can accomplish this task in a circuit that uses only \neg and \wedge gates. That is, provide a Boolean circuit for the \Leftrightarrow (equivalence) relation in terms of just \neg and \wedge gates.

2.15 Quantum gates are supposed to be unitary and hence logically reversible. How then, do you explain why, when you apply a Hadamard gate to state $|0\rangle$ and observe what state you obtain, that some of the time you find the result to be $|0\rangle$ and some of the time you find the result to be $|1\rangle$? How can a Hadamard gate be logically reversible if it is not producing a deterministic output. Where has our logic failed us?

2.16 What measurement, or repeated measurements, might you make on a quantum system in order to verify that the action of a box purported to enact a Hadamard gate is functioning correctly. The Hadamard gate enacts the transformations $|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$? How many measurements would you need to make to have a 99% confidence in your answer?

2.17 The Hadamard gate, $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ can be obtained, up to an overall global phase factor, using one R_x gate and one R_y gate, or using one R_y gate and one R_z gate. Can you obtain a Hadamard gate, up to an overall global phase factor, using just one R_x gate and one R_z gate? If so, exhibit the construction, else explain why it is impossible.

2.18 The FREDKIN and TOFFOLI gates are not the only (3-bit)-to-(3-bit) universal gates for reversible computing. For example, consider the reversible gate having the truth table given in Table 2.16 or, equivalently, the reversible gate represented by the matrix:

$$\text{NAND/NOR} \equiv \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (2.126)$$

If the first bit in the input is set to 0, then the gate computes the NAND of the second and third input bits. Conversely, if the first bit in the input is set to 1, the gate computes the NOR of the second and third qubits.

Find (a) a classical reversible circuit and (b) a quantum circuit that implements the NAND/NOR gate.

2.19 If U is a maximally entangling gate, show that a CNOT gate can always be obtained from U via a decomposition of the form $\text{CNOT} \equiv (A_1 \otimes A_2) \cdot U \cdot (R_y(\frac{\pi}{2}) \otimes \mathbb{1}) \cdot U^{-1}$ where A_1 and A_2 are single qubit gates.

2.20 Find the general form for a 2-qubit circuit, which uses only 1-qubit gates and Berkeley B gates, that will implement an arbitrary 2-qubit gate, U . How many Berkeley B gates are necessary? How does this compare to the number of CNOT gates needed for an arbitrary 2-qubit gate?

2.21 Given an arbitrary 1-qubit gate defined as $U = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix}$, what is the unitary matrix for the multiply controlled- U gate shown in Fig. 2.44?

2.22 What are the unitary matrices implied by the circuits shown in Fig. 2.45?

2.23 Determine the eigenvalues and normalized eigenvectors of the following operators built from the Pauli matrices:

(a) $\frac{1}{\sqrt{3}}(X + Y + Z)$

Table 2.16 Truth table of the NAND/NOR gate, which is a reversible gate containing the NAND and NOR gates quite explicitly

	Input bits	Output bits
NAND/NOR:	000	111
	001	101
	010	110
	011	011
	100	100
	101	001
	110	010
	111	000

- (b) $\frac{1}{\sqrt{2}}(X \cdot Y + Y \cdot Z)$
 (c) $\mathbb{1} \oplus X \oplus Y \oplus Z$
 (d) $e^{i\alpha(X \otimes X + Y \otimes Y)}$ (N.B. this is a *matrix* exponential).

2.24 Construct the unitary matrix, $U = e^{-i\mathcal{H}t/\hbar}$, of the quantum gate one would obtain from the Hamiltonian, \mathcal{H} , at time $t = 1$, assuming you are working in units of $\hbar = 1$, for each of the following Hamiltonians:

- (a) $\mathcal{H} = \alpha X \otimes \mathbb{1}$,
 (b) $\mathcal{H} = \alpha X \otimes X$,
 (c) $\mathcal{H} = \alpha X \otimes X + \beta Y \otimes Y$,
 (d) $\mathcal{H} = \alpha X \otimes Y + \beta Y \otimes X$,

where $X, Y, \mathbb{1}$ are Pauli matrices, and $\alpha, \beta \in \mathbb{R}$.

2.25 Decompose the following 2×2 unitary matrices into sequences of $R_y(\alpha)$, $R_z(\beta)$, and $Ph(\gamma)$ gates:

$$(a) \begin{pmatrix} \frac{1}{2}i\sqrt{\frac{1}{2}(5+\sqrt{5})} & \frac{1}{4}i(1-\sqrt{5}) \\ -\frac{1}{4}i(-1+\sqrt{5}) & -\frac{1}{2}i\sqrt{\frac{1}{2}(5+\sqrt{5})} \end{pmatrix}$$

$$(b) \begin{pmatrix} \frac{\sqrt{\frac{3}{2}}}{2} + \frac{1}{2\sqrt{2}} & \frac{\sqrt{\frac{3}{2}}}{2} - \frac{1}{2\sqrt{2}} \\ \frac{\sqrt{\frac{3}{2}}}{2} - \frac{1}{2\sqrt{2}} & -\frac{\sqrt{\frac{3}{2}}}{2} - \frac{1}{2\sqrt{2}} \end{pmatrix}$$

$$(c) \begin{pmatrix} \frac{1}{4}(3+i\sqrt{3}) & \frac{1}{4}(1-i\sqrt{3}) \\ \frac{1}{4}(1-i\sqrt{3}) & \frac{1}{4}(3+i\sqrt{3}) \end{pmatrix}$$

2.26 Assess the degree to which the following 2-qubit states are entangled by computing their “tangle”, i.e., $\text{tangle}(|\psi\rangle)$ where:

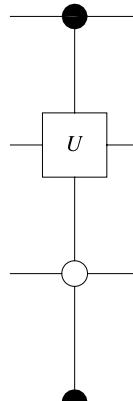


Fig. 2.44 A single qubit gate having an unusual pattern of control qubits

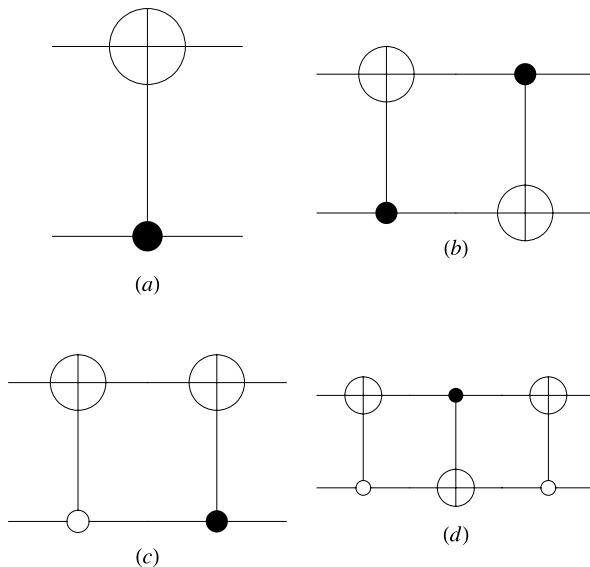


Fig. 2.45 Some 2-qubit gates involving “control-on-|0⟩” CNOT gates and reversed embeddings

- (a) $|\psi\rangle = \frac{1}{\sqrt{3}}|00\rangle + \frac{1}{\sqrt{3}}|01\rangle + \frac{1}{\sqrt{3}}|11\rangle$. Is the state entangled?
- (b) $|\psi\rangle = \frac{1}{3\sqrt{5}}|00\rangle + \frac{2}{3\sqrt{5}}|01\rangle + \frac{2}{3}\sqrt{\frac{2}{5}}|10\rangle + \frac{4}{3}\sqrt{\frac{2}{5}}|11\rangle$. Is the state entangled?
- (c) $|\psi\rangle = \frac{3}{2\sqrt{31}}|00\rangle + \frac{5}{2\sqrt{31}}|01\rangle + \frac{9}{2\sqrt{31}}|10\rangle + \frac{3}{2\sqrt{31}}|11\rangle$. Is the state entangled?
- (d) $|\psi\rangle = \frac{1}{\sqrt{2}}|01\rangle - \frac{i}{\sqrt{2}}|10\rangle$. Is the state entangled?
- (e) $|\psi\rangle = -\frac{e^{i\frac{\pi}{3}}}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|10\rangle$. Is the state entangled?

2.27 Consider the Bloch sphere with perpendicular axes x , y , and z . What 1-qubit gates, up to overall phase factors, perform the following operations on the Bloch sphere:

- (a) Map the state at the North pole of the Bloch sphere to the state at the South pole?
- (b) Map the state at $(x, y, z) = (0, 0, 1)$ to the state at $(x, y, z) = (0, 1, 0)$?
- (c) Map the state at $(x, y, z) = (0, 0, 1)$ to the state at $(x, y, z) = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)$?
- (d) Map the state at $(x, y, z) = (0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ to the state at $(x, y, z) = (0, -1, 0)$?
- (e) Map the state at $(x, y, z) = (0, 0, 1)$ to the state at $(x, y, z) = (\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$?

2.28 Compute the entangling power of the following 2-qubit quantum gates, and determine which ones are maximal entanglers:

$$\begin{aligned}
 \text{(a)} & \quad \begin{pmatrix} \cos\left(\frac{\alpha}{2}\right) & 0 & 0 & -\sin\left(\frac{\alpha}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right) & 0 & 0 & \cos\left(\frac{\alpha}{2}\right) \\ 0 & \cos\left(\frac{\alpha}{2}\right) & -\sin\left(\frac{\alpha}{2}\right) & 0 \\ 0 & \sin\left(\frac{\alpha}{2}\right) & \cos\left(\frac{\alpha}{2}\right) & 0 \end{pmatrix} \\
 \text{(b)} & \quad \begin{pmatrix} e^{-\frac{i\alpha}{2}} & 0 & 0 & 0 \\ 0 & \left(\frac{1}{2} + \frac{i}{2}\right)e^{\frac{i\alpha}{2}} & \left(\frac{1}{2} - \frac{i}{2}\right)e^{\frac{i\alpha}{2}} & 0 \\ 0 & \left(\frac{1}{2} - \frac{i}{2}\right)e^{-\frac{i\alpha}{2}} & \left(\frac{1}{2} + \frac{i}{2}\right)e^{-\frac{i\alpha}{2}} & 0 \\ 0 & 0 & 0 & e^{\frac{i\alpha}{2}} \end{pmatrix} \\
 \text{(c)} & \quad \begin{pmatrix} \cos\left(\frac{\alpha}{2}\right) & 0 & -\sin\left(\frac{\alpha}{2}\right) & 0 \\ \sin\left(\frac{\alpha}{2}\right) & 0 & \cos\left(\frac{\alpha}{2}\right) & 0 \\ 0 & \cos\left(\frac{\alpha}{2}\right) & 0 & -\sin\left(\frac{\alpha}{2}\right) \\ 0 & \sin\left(\frac{\alpha}{2}\right) & 0 & \cos\left(\frac{\alpha}{2}\right) \end{pmatrix} \\
 \text{(d)} & \quad \begin{pmatrix} e^{-\frac{i\alpha}{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{\frac{i\alpha}{2}} \\ 0 & 0 & ie^{-\frac{i\alpha}{2}} & 0 \\ 0 & ie^{\frac{i\alpha}{2}} & 0 & 0 \end{pmatrix} \\
 \text{(e)} & \quad \begin{pmatrix} \cos\left(\frac{\pi}{18}\right) & -\sin\left(\frac{\pi}{18}\right) & 0 & 0 \\ 0 & 0 & \cos\left(\frac{\pi}{18}\right) & \sin\left(\frac{\pi}{18}\right) \\ \sin\left(\frac{\pi}{18}\right) & \cos\left(\frac{\pi}{18}\right) & 0 & 0 \\ 0 & 0 & -\sin\left(\frac{\pi}{18}\right) & \cos\left(\frac{\pi}{18}\right) \end{pmatrix}
 \end{aligned}$$

2.29 In Fig. 2.41 we show a circuit sufficient to implement an arbitrary real unitary $U \in \mathbf{O}(4)$ that uses four CNOT gates. However, this circuit is not in its simplest form. Prove the following circuit identities and use them to show an arbitrary purely real unitary matrix having $\det(U) = -1$ can be implemented in a circuit requiring at most *three* CNOT gates:

- $(\mathbb{1} \otimes Z) \cdot \text{CNOT}_{2,1;2} \equiv \text{CNOT}_{2,1;2} \cdot (\mathbb{1} \otimes R_z(\pi) \cdot \text{Ph}(\frac{\pi}{2}))$ (i.e., prove the identity illustrated in Fig. 2.46)
- $\text{CNOT}_{1,2;2} \cdot \text{CNOT}_{2,1;2} \cdot \text{CNOT}_{1,2;2} \cdot \text{CNOT}_{2,1;2} \equiv \text{CNOT}_{2,1;2} \cdot \text{CNOT}_{1,2;2}$ (i.e., prove the circuit identity illustrated in Fig. 2.47)
- Hence, prove that any $U \in \mathbf{O}(4)$ with $\det(U) = -1$ can be implemented in a quantum circuit requiring at most three CNOT gates.

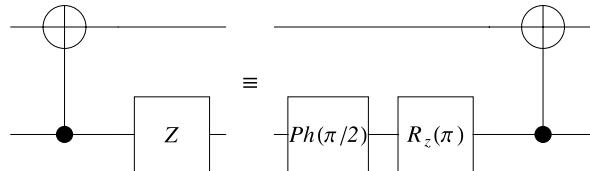


Fig. 2.46 A circuit identity that allows a Z gate to be moved through the control qubit of a CNOT gate

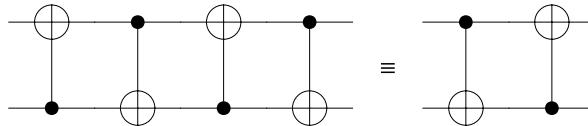


Fig. 2.47 A circuit identity that allows four CNOT gates to be contracted to two CNOT gates

2.30 Let \mathcal{M} be the 2-qubit gate that maps the computational basis states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$, into the “magic basis”:

$$|00\rangle \xrightarrow{\mathcal{M}} |\mathcal{M}_1\rangle = |\beta_{00}\rangle \quad (2.127)$$

$$|01\rangle \xrightarrow{\mathcal{M}} |\mathcal{M}_2\rangle = i |\beta_{10}\rangle \quad (2.128)$$

$$|10\rangle \xrightarrow{\mathcal{M}} |\mathcal{M}_3\rangle = i |\beta_{01}\rangle \quad (2.129)$$

$$|11\rangle \xrightarrow{\mathcal{M}} |\mathcal{M}_4\rangle = |\beta_{11}\rangle \quad (2.130)$$

where $|\beta_{00}\rangle$, $|\beta_{01}\rangle$, $|\beta_{10}\rangle$, and $|\beta_{11}\rangle$ are the Bell states defined by:

$$\begin{aligned} |\beta_{00}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\beta_{01}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\beta_{10}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\beta_{11}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \end{aligned} \quad (2.131)$$

- (a) Verify that the matrix, \mathcal{M} , which maps the computational basis into the magic basis, is given by:

$$\mathcal{M} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i & 0 & 0 \\ 0 & 0 & i & 1 \\ 0 & 0 & i & -1 \\ 1 & -i & 0 & 0 \end{pmatrix} \quad (2.132)$$

- (b) Prove that if U is a purely real unitary matrix then, regardless of the entangling power of U , the entangling power of $\mathcal{M} \cdot U \cdot \mathcal{M}$ is maximal, i.e., $\frac{2}{3}$.

- (c) Prove that if U is a purely real unitary matrix then, regardless of the entangling power of U , the entangling power of $\mathcal{M} \cdot U \cdot \mathcal{M}^\dagger$ is zero.
- (d) Check these claims by computing the entangling powers of U , $\mathcal{M} \cdot U \cdot \mathcal{M}$, and $\mathcal{M} \cdot U \cdot \mathcal{M}^\dagger$ for U given by:

$$U = \begin{pmatrix} \frac{\sqrt{\frac{3}{2}}}{2} & \frac{1}{2\sqrt{2}} & -\frac{\sqrt{\frac{3}{2}}}{2} & -\frac{1}{2\sqrt{2}} \\ \frac{1}{2\sqrt{2}} & -\frac{\sqrt{\frac{3}{2}}}{2} & -\frac{1}{2\sqrt{2}} & \frac{\sqrt{\frac{3}{2}}}{2} \\ \frac{1}{2\sqrt{2}} & \frac{\sqrt{\frac{3}{2}}}{2} & \frac{1}{2\sqrt{2}} & \frac{\sqrt{\frac{3}{2}}}{2} \\ \frac{\sqrt{\frac{3}{2}}}{2} & -\frac{1}{2\sqrt{2}} & \frac{\sqrt{\frac{3}{2}}}{2} & -\frac{1}{2\sqrt{2}} \end{pmatrix} \quad (2.133)$$

These remarkable properties explain why the vectors $|\mathcal{M}_1\rangle$, $|\mathcal{M}_2\rangle$, $|\mathcal{M}_3\rangle$ and $|\mathcal{M}_4\rangle$ are called the “magic basis”.

- 2.31** The nice properties of the “magic basis”, \mathcal{M} , do not, in general, carry over to complex unitary matrices.

- (a) Experiment by generating a complex 4×4 unitary matrix, U , at random, and compute $\det(U)$, $|\det(U)|$, $\text{EP}(U)$, $\text{EP}(\mathcal{M} \cdot U \cdot \mathcal{M})$, and $\text{EP}(\mathcal{M} \cdot U \cdot \mathcal{M}^\dagger)$. Such a matrix is most easily generated by guessing a quantum circuit containing a few R_x , R_z , Ph , and CNOT gates. After a few experiments you should convince yourself that the nice properties of the magic basis do not hold, in general, for complex unitaries.
- (b) Show that $\det(\text{SWAP}^\alpha) = (-1)^\alpha$, rather than ± 1 as is the case for all real unitary matrices. Based on this, would you expect $\text{EP}(\mathcal{M} \cdot \text{SWAP}^\alpha \cdot \mathcal{M})$ to be maximal? Compute $\text{EP}(\mathcal{M} \cdot \text{SWAP}^\alpha \cdot \mathcal{M})$ to check your answer.
- (c) Given that $\det(i\text{SWAP}) = 1$ (just like many real unitaries), would you expect $\text{EP}(\mathcal{M} \cdot i\text{SWAP} \cdot \mathcal{M}^\dagger)$ to be non-entangling? Compute $\text{EP}(\mathcal{M} \cdot i\text{SWAP} \cdot \mathcal{M}^\dagger)$ to check your answer.

- 2.32** Prove each of the following identities:

- (a) $\text{SWAP} \cdot \text{SWAP} \cdot \text{SWAP} = \text{SWAP}$
 (b) $\text{SWAP} \cdot i\text{SWAP} \cdot \text{SWAP} = i\text{SWAP}$
 (c) $\text{SWAP} \cdot \text{SWAP}^\alpha \cdot \text{SWAP} = \text{SWAP}^\alpha$
 (d) $\text{SWAP}^\dagger = \text{SWAP}$
 (e) $i\text{SWAP}^\dagger = i\text{SWAP}^3$
 (f) $(\text{SWAP}^\alpha)^\dagger = \text{SWAP}^{-\alpha}$

The first three identities show that it makes no difference which way around you insert a SWAP, iSWAP, and SWAP^α gate into a quantum circuit. The last two identities show that, whereas SWAP and CNOT are their own inverses, iSWAP and SWAP^α are not.

2.33 Invent an icon for the Berkeley B gate. In choosing your icon, decide whether you need to make it asymmetric so that you can distinguish between embedding the gate one way around or upside down, or whether this is immaterial. Then express the inverse of the Berkeley B gate in terms of itself and 1-qubit gates if necessary.

2.34 Invent an icon for the CSIGN gate. In choosing your icon, decide whether you need to make it asymmetric so that you can distinguish between embedding the gate one way around or upside down, or whether this is immaterial. Then express the inverse of the CSIGN gate in terms of itself and 1-qubit gates if necessary. Is the CSIGN gate a maximal entangling gate?

2.35 What matrix do you obtain when you raise the matrix representing the Berkeley B gate to the sixteenth power, i.e., B^{16} ?

Chapter 3

Quantum Circuits

[*Quantum Computing*] “...means you can try to answer questions you thought the Universe was going to have to do without.”

– Bruce Knapp¹

A quantum circuit provides a visual representation of how a complicated multi-qubit quantum computation can be decomposed into a sequence of simpler, usually 1-qubit and 2-qubit, quantum gates. In general, a given unitary matrix, which specifies some desired quantum computation, will admit many different, but equivalent, decompositions depending on the set of primitive quantum gates used, and the skill of the quantum circuit designer in composing those gates in an intelligent way. In this chapter we shall look the relationship between multi-qubit unitary operators and their corresponding quantum circuits. You will learn how to compute a unitary operator from a quantum circuit description of it to compute the unitary operator corresponding to a given quantum circuit, and how find a quantum circuit sufficient to implement a desired unitary operator. We will also look at the surprisingly efficient quantum circuits for computing various key quantum transforms such as quantum versions of the Fourier, wavelet, cosine, and fractional Fourier transforms.

3.1 Quantum Circuit Diagrams

A quantum circuit *diagram* provides a visual representation of a sequence of quantum gate operations (e.g., see Fig. 3.1).

¹Source: Comment made by physicist Bruce Knapp of Columbia University to reporter William J. Broad recounted in “With Stakes High, Race is on for Fastest Computer of All” from the 1st February 1983 issue of the New York Times. In the 1980’s Japan and the U.S. were racing to make faster and faster supercomputers. Knapp was commenting on the capabilities of a new classical supercomputer he and his colleagues were developing. However, the quotation is even more fitting for quantum computers.

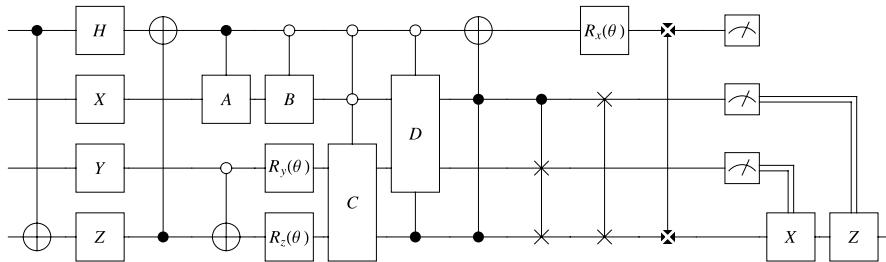


Fig. 3.1 A quantum circuit diagram illustrating different types of quantum gates. Single qubit gates are drawn as boxes on the qubit on which they act labeled with their gate name (H for Hadamard gate, X and Y for Pauli gates, $R_x(\theta)$ for a rotation gate about the x -axis through angle θ etc.). Controlled gates are drawn with their controls depicted as circles (white for control-on- $|0\rangle$ and black for control-on- $|1\rangle$) and the operation they perform on the target is show as a labeled box. At the end of the computation certain qubit values are read out. Also shown are the special 2-qubit gates, CNOT, SWAP, and iSWAP

An n -qubit circuit consists of n horizontal rails, which correspond to the n qubits. Our convention is to have the most significant qubit on the top rail and the least significant qubit on the bottom rail. Time flows from left to right in the quantum circuit, with the leftmost gates applied before the rightmost gates. If the rail is a single line, it carries a quantum value (i.e., a pure or mixed quantum state). If the rail is a double line, it carries a classical bit value. The double rails typically appears immediately after a qubit has been measured and a classical bit value readout obtained. A measurement gate is indicated by an icon that resembles a meter. This measures a qubit in the computational basis and returns the result $|0\rangle$ or $|1\rangle$.

Inputs to the quantum circuit are drawn to the left of the horizontal rails, and outputs from the circuit are drawn to right of the rails. Typically, the output will not be a product state (i.e., expressible as the direct product of a state for each qubit) but will, instead, be entangled.

A 1-qubit logic gate on the i -th qubit is depicted as a square box on the i -th rail labeled with the name of the gate. A two qubit gate acting between the i -th and j -th qubits is depicted as an icon with end points on qubits i and j . If such a gate is a controlled gate, the controlling value is depicted as a black or white dot according to whether the controlled operation is applied when the control qubit is in state $|1\rangle$ or $|0\rangle$ respectively. The quantum gate icons we use were introduced in Chap. 2 and merely summarized in Fig. 3.2.

3.2 Computing the Unitary Matrix for a Given Quantum Circuit

A quantum computation on n qubits typically requires several quantum gates to be applied, sequentially, in parallel, or conditionally, to various subsets of the qubits.

Fig. 3.2 Quantum gate icons and their corresponding descriptions in terms of unitary matrices for the most common quantum gates

$$\begin{array}{c|c} \boxed{H} & \equiv \end{array} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{array}{c|c} \boxed{Ph(\theta)} & \equiv \end{array} \begin{pmatrix} e^{i\theta} & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

$$\begin{array}{c|c} \boxed{X} & \equiv \end{array} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{array}{c|c} \boxed{R_x(\theta)} & \equiv \end{array} \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i \sin\left(\frac{\theta}{2}\right) \\ -i \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

$$\begin{array}{c|c} \boxed{Y} & \equiv \end{array} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{array}{c|c} \boxed{R_y(\theta)} & \equiv \end{array} \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

$$\begin{array}{c|c} \boxed{Z} & \equiv \end{array} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{array}{c|c} \boxed{R_z(\theta)} & \equiv \end{array} \begin{pmatrix} e^{-\frac{i\theta}{2}} & 0 \\ 0 & e^{\frac{i\theta}{2}} \end{pmatrix}$$

$$\begin{array}{c|c} \text{Controlled NOT gate} & \equiv \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{pmatrix} \begin{array}{c|c} \text{Controlled phase gate} & \equiv \end{array} \begin{pmatrix} u_{11} & u_{12} & 0 & 0 \\ u_{21} & u_{22} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{array}{c|c} \text{Controlled NOT gate} & \equiv \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{array}{c|c} \text{Controlled phase gate} & \equiv \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The net unitary transformation they perform is computed by composing the unitary matrices of the corresponding quantum gates according to certain rules. These rules make use of three matrix operations, the dot product, the direct product and the direct sum. The dot product corresponds to the usual product of two matrices. However, the direct product and direct sum are taught less often in undergraduate linear algebra courses. Therefore, in the interests of having a self-contained text, we include their definitions here.

Fig. 3.3 The net effect of gates acting in series is obtained from their dot product in reverse order

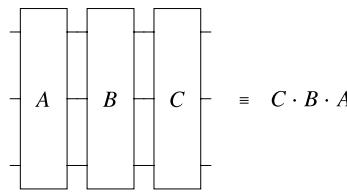
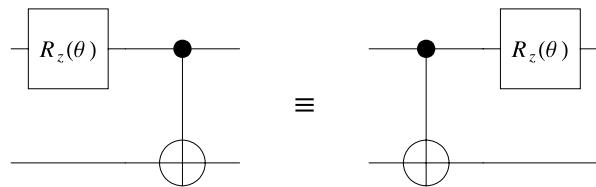


Fig. 3.4 If the matrices corresponding to two gates that are applied sequentially commute they may be applied in either order without it affecting the overall transformation they achieve



3.2.1 Composing Quantum Gates in Series: The Dot Product

The dot product arises when composing the effect of quantum gates that act in series.

Dot Product If A is a $m \times p$ dimensional matrix and B is an $p \times n$ dimensional matrix, their dot product, $A \cdot B$ is the $m \times n$ dimensional matrix defined by:

$$A \cdot B = \begin{pmatrix} a_{11} & \dots & a_{1p} \\ \vdots & \dots & \vdots \\ a_{i1} & \dots & a_{ip} \\ \vdots & \dots & \vdots \\ a_{m1} & \dots & a_{mp} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & \dots & b_{1j} & \dots & b_{1n} \\ \vdots & \dots & \vdots & \dots & \vdots \\ b_{p1} & \dots & b_{pj} & \dots & b_{pn} \end{pmatrix} = \begin{pmatrix} c_{11} & \dots & c_{1n} \\ \vdots & \dots & \vdots \\ c_{ij} & \dots & \vdots \\ \vdots & \dots & \vdots \\ c_{m1} & \dots & c_{mn} \end{pmatrix} \quad (3.1)$$

where $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{ip}b_{pj} = \sum_{k=1}^p a_{ik}b_{kj}$.

If several gates, e.g., A , B , C say, act upon the same subset of qubits, then those gates must be applied in *series* and their overall effect is computed using the dot product. In a quantum circuit (with time flowing from left to right) if A acts before B and B acts before C , their overall effect is computed by their dot product *in reverse order*, i.e., $C \cdot B \cdot A$, as shown in Fig. 3.3.

If the unitary matrices for a pair of gates that act on the same set of qubits commute, i.e., for gates A and B , $A \cdot B = B \cdot A$, then the order in which the gates are performed is immaterial.

Example Taking $A = (R_z(\theta) \otimes \mathbb{1})$ and $B = \text{CNOT}$, we have that $(R_z(\theta) \otimes \mathbb{1}) \cdot \text{CNOT} = \text{CNOT} \cdot (R_z(\theta) \otimes \mathbb{1})$ and so, as depicted in Fig. 3.4, their order does not matter.

However, if the unitary matrices for a pair of gates that act on the same set of qubits do not commute, i.e., for gates A and B , $A \cdot B \neq B \cdot A$, then the order in which the gates are performed affects the overall transformation they achieve.

Fig. 3.5 If the matrices corresponding to two gates that are applied sequentially do not commute the order in which they are performed matters

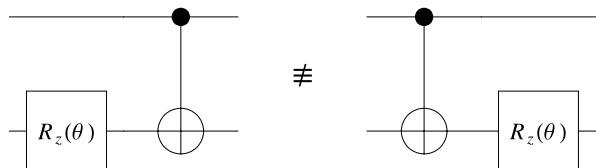
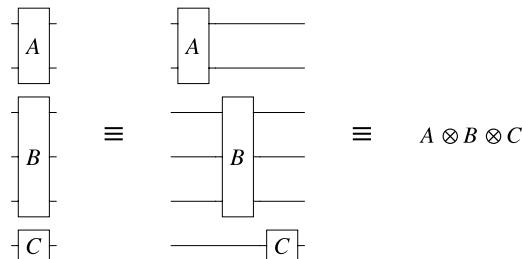


Fig. 3.6 Quantum gates that act on disjoint sets of qubits can be applied in parallel or any either order with no change to the overall operation performed



Example Taking $A = (\mathbb{1} \otimes R_z(\theta))$ and $B = \text{CNOT}$, we have that $(\mathbb{1} \otimes R_z(\theta)) \cdot \text{CNOT} \neq \text{CNOT} \cdot (\mathbb{1} \otimes R_z(\theta))$ and so their order, as depicted in Fig. 3.5, affects the net transformation that is achieved.

3.2.2 Composing Quantum Gates in Parallel: The Direct Product

If adjacent gates within a quantum circuit act on independent subsets of the qubits, then those gates can be applied simultaneously in *parallel*, as depicted in Fig. 3.6. The operation that computes the net effect of parallel gates is the direct product.

Direct Product If A is a $p \times q$ dimensional matrix and B is an $r \times s$ dimensional matrix, their direct product, $A \otimes B$ is the $pr \times qs$ dimensional matrix defined by:

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1q}B \\ a_{21}B & a_{22}B & \cdots & a_{2q}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1}B & a_{p2}B & \cdots & a_{pq}B \end{pmatrix} \quad (3.2)$$

Notice that the dimensions of the direct product matrix can grow very rapidly if we take the direct product of multiple matrices.

An especially common circumstance is when a j -qubit quantum gate, U say, acts on a subset of the qubits and there is no explicit operation on the other qubits. Mathematically, this can be regarded as *parallel* gate operations in which an i -qubit identity gate (“no-op”) is applied to qubits 1 through i , the j -qubit U gate is applied to qubits $i + 1$ through $i + 1 + j$, and a k -qubit identity gate (“no-op”) is applied to qubits $i + j + 1$ through $i + j + k$. As the direct product (“ \otimes ”) is the mathematical operation that combines gates in parallel, the net gate, shown in Fig. 3.7, is $\mathbb{1}_{2^i} \otimes$

Fig. 3.7 When a gate acts on a contiguous subset of qubits, and no other gates act, the net operation can be thought of as the parallel application of “no-op”

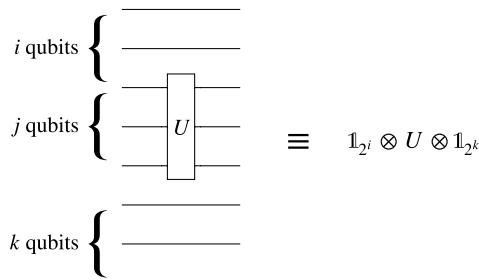
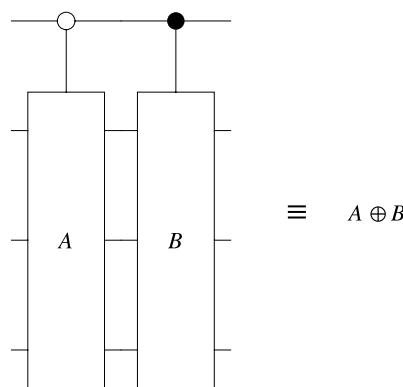


Fig. 3.8 The direct sum $A \oplus B$ describes a controlled quantum gate having one control qubit



$U \otimes \mathbb{1}_2^k$, where $\mathbb{1}_2^\ell$ is a ℓ -qubit identity (“no-op”) gate, i.e., a $2^\ell \times 2^\ell$ dimensional identity matrix.

3.2.3 Composing Quantum Gates Conditionally: The Direct Sum

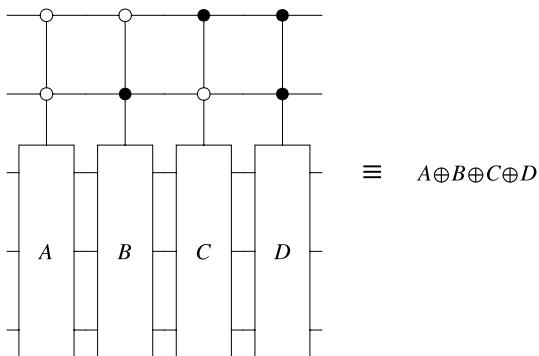
Sometimes we might want to compute the effect of a gate that is applied conditionally. Typically, we have one subset of qubits (called “controls”) whose values dictate what gate is to be applied to some other subset of qubits (called the “targets”). The mathematical operation for composing gates conditionally is the direct sum.

Direct Sum If *A* is a $p \times q$ dimensional matrix and *B* is an $r \times s$ dimensional matrix, their direct sum, $A \oplus B$ is the $(p+r) \times (q+s)$ dimensional matrix defined by:

$$A \oplus B = \begin{pmatrix} A & 0_{p \times s} \\ 0_{r \times q} & B \end{pmatrix} \quad (3.3)$$

In quantum computing the matrices involved will invariably be square and have dimensions that are a power of two. The direct sum is the basic mathematical operation by which controlled (or conditional) quantum logic gates are built. For example,

Fig. 3.9 The direct sum $A \oplus B \oplus C \oplus D$ describes a controlled quantum gate having two control qubits



if A and B are arbitrary 3-qubit quantum gates the operation $A \oplus B$ means, as shown in Fig. 3.8, that gate A is applied to the bottom three qubits if the top (control) qubit is in state $|0\rangle$, and gate B is applied to the bottom three qubits if the top (control) qubit is in state $|1\rangle$.

Direct sums can be generalized quite easily to allow for multiple controls. For example, if A , B , C , and D are all 2-qubit gates, then the circuit made from $A \oplus B \oplus C \oplus D$ applies A to the bottom two qubits if the top two (control) qubits are in state $|00\rangle$, and applies B if the top two control qubits are in state $|01\rangle$ etc. The pattern is best seen in the quantum circuit direct for this direct sum shown in Fig. 3.9.

We can use the matrix dot product, direct product, and direct sum, to translate a quantum circuit diagram into its implied unitary matrix, and thereby compute the overall transformation achieved by a quantum circuit. Summing up what we saw above, the basic rules for mapping from a quantum circuit diagram to its equivalent unitary matrix are as follows:

- Rule 1—No-op: if no gate is applied at a given step this is mathematically equivalent to applying the identity gate, $\mathbb{1}$, at that step.
- Rule 2—Sequential: if gates A , B , and C are applied sequentially to a given subset of qubits in the order A first, then B , then C , the overall unitary matrix is given by $C.B.A$, i.e., their dot product *in reverse order*.
- Rule 3—Parallel: if gates A , B , and C are applied to qubits 1, 2, and 3 simultaneously, their net effect is computed from the direct product $A \otimes B \otimes C$.
- Rule 4—Conditional: if qubit 1 is $|0\rangle$ apply gate A to qubit 2 and if qubit 1 is $|1\rangle$ apply gate B to qubit 2, is given by $A \oplus B$.
- Rule 5—Permute: if a gate A is to be applied to non-adjacent qubits, permute the qubits, according to permutation P , so they are adjacent, perform the gate and unpermute the qubits. The net effect is $P^{-1}.A.P$.

The exercises allow you to practice using these rules, and to generalize them to more complicated multi-qubit gates.

3.2.4 Measures of Quantum Circuit Complexity

In the quantum circuit model of quantum computation, one can characterize “complexity” as the width, size, and length of the quantum circuit. Here *width* is the total number of qubits on which the circuit acts (including any ancillae qubits); *size* is the total number of gates the circuit uses, and *length* is the number of serial gate operations after having parallelized the circuit to the maximum extent possible. Most often we take the length of the quantum circuit as the primary indicator of its complexity.

If the size (or any other complexity measure) of the quantum circuit grows as a polynomial in the number of qubits, n , i.e., as a function like n^k with $k > 0$, the circuit is regarded as being of “polynomial-size” and hence an *efficient* way to perform the desired computation. On the other hand, if the size of the quantum circuit grows as an exponential in the number of qubits, i.e., a function like 2^n , or e^n , the circuit is deemed of “exponential-size” and an *inefficient* way of achieving the desired computation. Luckily, many useful quantum computations admit polynomial-size quantum circuits.

For quantum computing to offer a genuine breakthrough compared to classical computing the minimum circuit complexity needed to achieve some computation quantumly must be significantly less than that needed to achieve the same computation classically. In the ideal case the complexity separation will be exponential. That is, ideally, we would like the quantum circuit complexity to grow as a polynomial function in the number of qubits, n , i.e., $\mathcal{O}(n^k)$, whereas the complexity of the corresponding classical circuit grows exponentially with the number of qubits, i.e., as $\mathcal{O}(e^n)$.

Unfortunately, it is now known that a *maximally general* quantum computation on n -qubits (i.e., a fully general $2^n \times 2^n$ unitary matrix) requires at least $\frac{23}{48}4^n - \frac{3}{2}2^n + \frac{4}{3}$ CNOT gates and this result cannot be improved by more than a factor of two, [453]. Thus, the size of the circuit for a maximally general n -qubit quantum computation is *exponential* in the number of qubits and cannot therefore be implemented “efficiently”.

However, it turns out that many of the computations that arise in practical applications are naturally unitary (which is lucky), and naturally implementable efficiently in quantum circuits (which is even more lucky). The reason for this is that the matrix elements of the practically useful unitary matrices are typically highly interrelated, which means that the matrix as a whole requires less than the full complement of degrees of freedom to specify it completely. Nature did not have to be so kind to us, but this appears to be the case. Perhaps there is a deeper reason to it, but there it is.

In the following sections we shall look at some of these specially structured unitary matrices and the polynomially-sized quantum circuits into which they can be decomposed.

3.3 Quantum Permutations

In Chap. 2 we saw that the actions performed by classical reversible gates can be represented mathematically in terms of permutation matrices, i.e., square matrices having a single 1 in each row and column and zeroes everywhere else. Each distinct n -bit classical reversible gate can be thought of as applying a different permutation to the 2^n bit strings that can be made from n bits. Thus, the classical reversible gates could equally be called classical permutation gates.

Not surprisingly, as permutation matrices are also unitary matrices, a given classical permutation gate can also be viewed as a quantum gate that applies the same permutation to the computational basis states of n -qubits, i.e., $|00\dots 0\rangle$, $|00\dots 1\rangle$, ..., $|11\dots 1\rangle$ that the classical permutation gate applies to bit strings. We call such quantum gates “*quantum permutations*” in analogy to their classical counterparts.

There are, however, important differences between classical permutation gates and quantum ones. Whereas classical permutation gates are restricted to act upon only classical bit strings, the quantum permutation gates are free to act upon arbitrary n -qubit states, including entangled states and mixed states. This allows the quantum gates can apply a given permutation to a superposition of (essentially) several bit-string inputs at once, which can be put to great advantage in many quantum algorithms.

The number of possible quantum permutations grows worse than exponentially in the number of qubits. Crudely speaking, a $2^n \times 2^n$ permutation matrix can be thought of as an $2^n \times 2^n$ identity matrix with its rows (or columns) permuted. As there are $2^n!$ ways to permute 2^n objects there are $2^n!$ possible quantum permutations. Most of these correspond to rather haphazard permutations and do not, therefore, have a useful computational interpretation. But some of them turn out to be quite useful in manipulating quantum states.

Although a given quantum permutation can be specified by a $2^n \times 2^n$ permutation matrix, it is sometimes easier to interpret its action in terms of its affect on a column vector of 2^n amplitudes or its affect on a product state of n -qubits. We will flip back and forth between interpretations in the examples below.

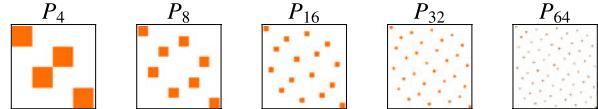
3.3.1 Qubit Reversal Permutation: P_{2^n}

The qubit reversal permutation arises in the circuit for the quantum Fourier transform (QFT) as the final step needed to put the qubits back into the order they had when they entered the circuit, i.e., with the most significant bit on the top line of the circuit, and the least significant bit on the bottom line of the circuit. By preserving the ordering of the qubits the QFT can be treated as a module that may be slotted into a quantum computation without requiring qubit re-ordering operations on the preceding or succeeding parts of the computation.

The qubit reversal permutation is defined via its affect on the computational basis states, i.e., the 2^n n -qubit states in which each bit, j_i , is 0 or 1:

$$P_{2^n} : |j_1 j_2 \dots j_n\rangle \longrightarrow |j_n j_{n-1} \dots j_1\rangle \quad (3.4)$$

Fig. 3.10 The structure of the $n = 2$ to $n = 6$ qubit reversal permutation matrices $P_{2^1}, P_{2^2}, \dots, P_{2^6}$



The qubit reversal permutation can be specified equivalently as a unitary matrix, which is shown here for the case of 1-, 2- and 3-qubits:

$$P_{2^1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad P_{2^2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$P_{2^3} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.5)$$

The structure of the qubit reversal permutation matrices can be seen more clearly in Fig. 3.10. Each figure depicts a matrix, with dimensions a power of two, and color-coded so that the (i, j) -th element of the matrix is white if that element is a 0 and orange if it is a +1. As we scale up to larger matrices this makes it easier to see that there is a special structure to these qubit reversal matrices.

Although the qubit reversal permutation is defined in terms of its action on computational basis states, it is not restricted to acting on only these kinds of states. For example, suppose a 3-qubit quantum computation returns an unentangled output state $|\psi_a\rangle|\psi_b\rangle|\psi_c\rangle$ where $|\psi_a\rangle = a_0|0\rangle + a_1|1\rangle$, $|\psi_b\rangle = b_0|0\rangle + b_1|1\rangle$, and $|\psi_c\rangle = c_0|0\rangle + c_1|1\rangle$. Then the operation P_8 will reverse the qubits, i.e., $P_8|\psi_a\rangle|\psi_b\rangle|\psi_c\rangle = |\psi_c\rangle|\psi_b\rangle|\psi_a\rangle$. Hence the name “qubit reversal permutation”. To see this note that $P_{2^n} = P_{2^n}^{-1}$, and check the amplitudes.

$$|\psi_a\rangle|\psi_b\rangle|\psi_c\rangle \equiv \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \otimes \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \otimes \begin{pmatrix} c_0 \\ c_1 \end{pmatrix}$$

$$= \begin{pmatrix} a_0b_0c_0 \\ a_0b_0c_1 \\ a_0b_1c_0 \\ a_0b_1c_1 \\ a_1b_0c_0 \\ a_1b_0c_1 \\ a_1b_1c_0 \\ a_1b_1c_1 \end{pmatrix}$$

$$\begin{aligned}
&= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} a_0 b_0 c_0 \\ a_1 b_0 c_0 \\ a_0 b_1 c_0 \\ a_1 b_1 c_0 \\ a_0 b_0 c_1 \\ a_1 b_0 c_1 \\ a_0 b_1 c_1 \\ a_1 b_1 c_1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} c_0 b_0 a_0 \\ c_0 b_0 a_1 \\ c_0 b_1 a_0 \\ c_0 b_1 a_1 \\ c_1 b_0 a_0 \\ c_1 b_0 a_1 \\ c_1 b_1 a_0 \\ c_1 b_1 a_1 \end{pmatrix} \\
&= P_8^{-1} \cdot \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} \otimes \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \otimes \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \\
&= P_8^{-1}(|\psi_c\rangle|\psi_b\rangle|\psi_a\rangle) \tag{3.6}
\end{aligned}$$

Hence $P_8|\psi_a\rangle|\psi_b\rangle|\psi_c\rangle = |\psi_c\rangle|\psi_b\rangle|\psi_a\rangle$.

A quantum circuit for P_{2^n} , when n is even, that uses only SWAP (i.e., Π_4) gates between adjacent qubits can be obtained from the following factorization:

$$P_{2^n} = \left[\underbrace{(\Pi_4 \otimes \Pi_4 \cdots \otimes \Pi_4)}_{\frac{n}{2}} (\mathbb{1}_2 \otimes \underbrace{\Pi_4 \otimes \Pi_4 \cdots \otimes \Pi_4}_{\frac{n}{2}-1} \otimes \mathbb{1}_2) \right]^{n/2} \tag{3.7}$$

This factorization corresponds to a quantum circuit for (even n) qubit-reversal of the form shown in Fig. 3.11. Conversely, when n is odd the factorization of P_{2^n} becomes:

$$P_{2^n} = \left[(\mathbb{1}_2 \otimes \underbrace{\Pi_4 \otimes \cdots \otimes \Pi_4}_{\frac{n-1}{2}}) \cdot \underbrace{(\Pi_4 \otimes \cdots \otimes \Pi_4 \otimes \mathbb{1}_2)}_{\frac{n-1}{2}} \right]^{\frac{n-1}{2}} \cdot (\mathbb{1}_2 \otimes \underbrace{\Pi_4 \otimes \cdots \otimes \Pi_4}_{\frac{n-1}{2}}) \tag{3.8}$$

and its corresponding circuit is shown in Fig. 3.12. More gate-efficient versions of the qubit-reversal permutation are possible if SWAP gates between non-adjacent qubits are allowed as in Fig. 3.13. However, physically, it is much more difficult to achieve SWAP operations that are not amongst nearest neighbor qubits.

Thus, we can interpret the qubit-reversal quite literally when the input state is a direct product of n single qubit states. In this case, the qubits emerge from the circuit in the opposite order they went in. But how are we to interpret what this operation is doing if the input state is not a product state of n single qubit states?

Fig. 3.11 Quantum circuit for the qubit reversal permutation, when the number of qubits, n , is even, using gates that only act on adjacent qubits. In the factorization we use the notation Π_4 for the 4×4 unitary matrix corresponding to a 2-qubit SWAP gate

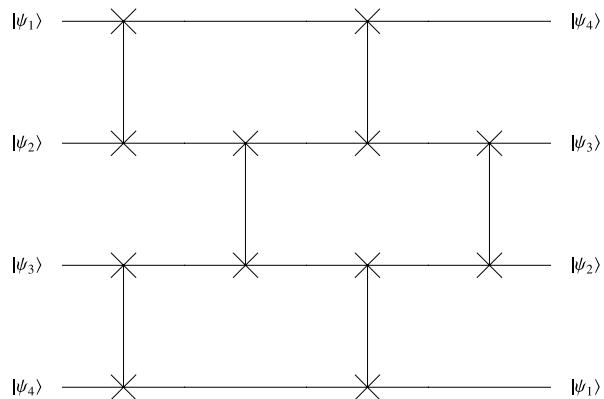
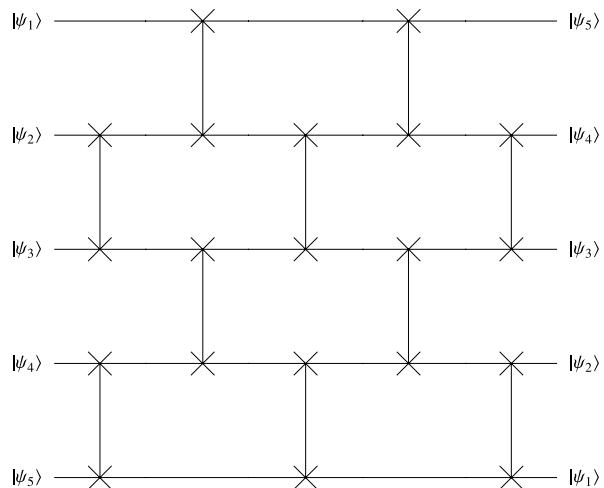


Fig. 3.12 Quantum circuit for the qubit reversal permutation, when the number of qubits, n , is odd, using gates that only act on adjacent qubits. In the factorization we use the notation Π_4 for the 4×4 unitary matrix corresponding to a 2-qubit SWAP gate



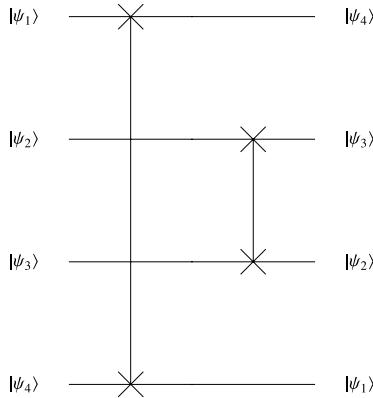
The best way to see what happens is to work with the column vectors of amplitudes corresponding to arbitrary superposition states of n qubits. Considering the $n = 3$ case as an illustrative example we have:

$$|\psi\rangle = a_0|000\rangle + a_1|001\rangle + a_2|010\rangle + a_3|011\rangle + a_4|100\rangle$$

$$+ a_5|101\rangle + a_6|110\rangle + a_7|111\rangle$$

$$\equiv \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix} \quad (3.9)$$

Fig. 3.13 Quantum circuit for the qubit reversal permutation using SWAP gates that can act between any pair of qubits



and the effect of the qubit-reversal P_8 on this state is:

$$P_8|\psi\rangle = P_8 \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix} = \begin{pmatrix} a_0 \\ a_4 \\ a_2 \\ a_3 \\ a_6 \\ a_1 \\ a_5 \\ a_7 \end{pmatrix} = \begin{pmatrix} a_{\downarrow 0} \\ a_{\downarrow 1} \\ a_{\downarrow 2} \\ a_{\downarrow 3} \\ a_{\downarrow 4} \\ a_{\downarrow 5} \\ a_{\downarrow 6} \\ a_{\downarrow 7} \end{pmatrix} \quad (3.10)$$

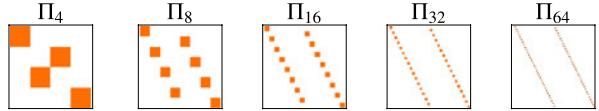
where $\downarrow i$ means to take the bits in the binary representation of integer i , padded with zeroes on the left to make the number n bits wide ($n = 3$ in our example), then reverse the order of these bits, and map the result back into an integer to give the new index. Thus, for the case of $n = 3$ we have:

$$\begin{aligned} \downarrow 0 &\rightarrow 000 \rightarrow 000 \rightarrow 0 \\ \downarrow 1 &\rightarrow 001 \rightarrow 100 \rightarrow 4 \\ \downarrow 2 &\rightarrow 010 \rightarrow 010 \rightarrow 2 \\ \downarrow 3 &\rightarrow 011 \rightarrow 110 \rightarrow 6 \\ \downarrow 4 &\rightarrow 100 \rightarrow 001 \rightarrow 1 \\ \downarrow 5 &\rightarrow 101 \rightarrow 101 \rightarrow 5 \\ \downarrow 6 &\rightarrow 110 \rightarrow 011 \rightarrow 3 \\ \downarrow 7 &\rightarrow 111 \rightarrow 111 \rightarrow 7 \end{aligned} \quad (3.11)$$

3.3.2 Qubit Cyclic Left Shift Permutation: Π_{2^n}

The qubit left shift permutation arises in applications such as wavelet pyramid and packet algorithms where shifting and shuffling of the amplitudes is required. The operation is defined in terms of its effect on the computational basis states, i.e., the

Fig. 3.14 The structure of the $n = 2$ (left) to $n = 6$ (right) qubit cyclic left shift permutation matrices



2^n n -qubit states in which each qubit, j_i , is 0 or 1:

$$\Pi_{2^n} : |j_1 j_2 \dots j_{n-1} j_n\rangle \longrightarrow |j_2 j_3 \dots j_{n-1} j_n j_1\rangle \quad (3.12)$$

The qubit left shift permutation can be specified equivalently as a unitary matrix, which is shown here for the case of 1-, 2- and 3-qubits are:

$$\begin{aligned} \Pi_{2^1} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \Pi_{2^2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \\ \Pi_{2^3} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned} \quad (3.13)$$

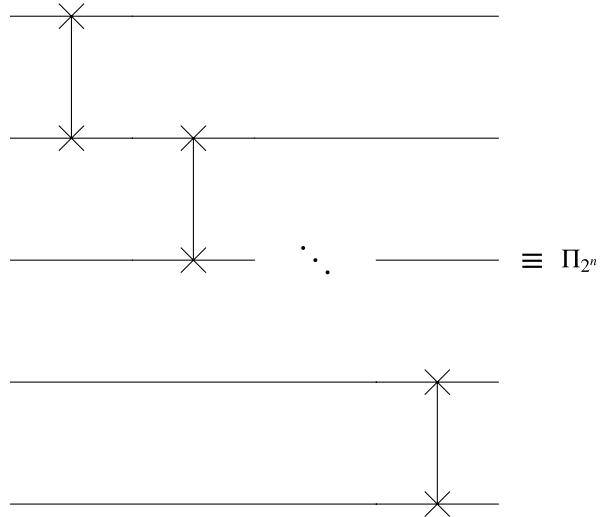
The structure of the qubit cyclic left shift permutation matrices can seen more clearly in Fig. 3.14. Notice that the qubit left shift operation, Π_{2^n} , can also be understood as the operation that performs a perfect shuffle on the column vector of amplitudes. For example, the 3-qubit left shift permutation, Π_8 acting on the general three qubit state $a_0|000\rangle + a_1|001\rangle + a_2|010\rangle + a_3|011\rangle + a_4|100\rangle + a_5|101\rangle + a_6|110\rangle + a_7|111\rangle$ shuffles the amplitudes:

$$\Pi_8 \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix} = \begin{pmatrix} a_0 \\ a_4 \\ a_1 \\ a_5 \\ a_2 \\ a_6 \\ a_3 \\ a_7 \end{pmatrix} \quad (3.14)$$

A circuit for the cyclic qubit left shift permutation can, as shown in Fig. 3.15, be derived from the recursive factorization of the Π_{2^n} matrix:

$$\Pi_{2^n} = (\mathbb{1}_{2^{n-2}} \otimes \Pi_{2^2}) \cdot (\Pi_{2^{n-1}} \otimes \mathbb{1}_2) \quad (3.15)$$

Fig. 3.15 Quantum circuit for the qubit cyclic left shift permutation operation, Π_{2^n} . This consists of a simple cascade of SWAP gates



where $\mathbb{1}_{2^i}$ is the $2^i \times 2^i$ dimensional identity matrix. Note that Π_4 is simply the 2-qubit SWAP gate. A SWAP operation that swaps qubits i and j , is given by $\text{SWAP}(i, j) \equiv \text{CNOT}(i, j) \cdot \text{CNOT}(j, i) \cdot \text{CNOT}(i, j)$, where $\text{CNOT}(i, j)$ is a CNOT gate with control qubit i and target qubit j .

The qubit cyclic left shift permutation, Π_{2^n} , is used in Sect. 3.5 within quantum circuits for the wavelet packet and quantum wavelet pyramid algorithms.

3.3.3 Amplitude Downshift Permutation: Q_{2^n}

Another permutation that turns out to be surprisingly useful is the n -qubit downshift permutation, Q_{2^n} . This matrix has the form:

$$Q_{2^n} = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \quad (3.16)$$

The structure of the Q_{2^n} matrices is seen in Fig. 3.16.

Fig. 3.16 The structure of the $n = 2$ (left) to $n = 6$ (right) downshift permutation matrix Q_{2^n}

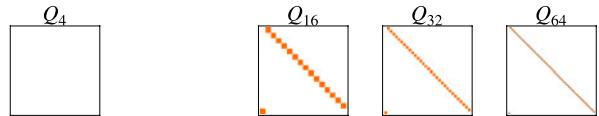
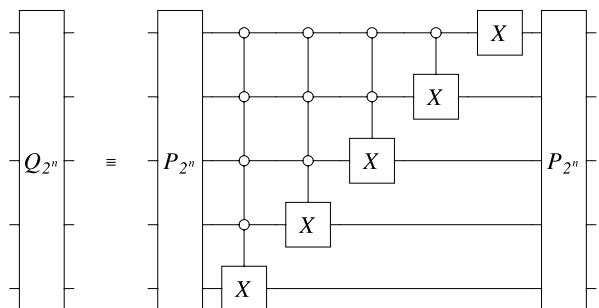


Fig. 3.17 The high level factorization of the Q_{2^n} downshift permutation matrix in terms of the qubit reversal permutation P_{2^n} and multiply controlled X gates where the control action occurs when the control qubits are in state $|0\rangle$ (as indicated by the open circles)



In turn, Q_{2^n} may be factored as:

$$Q_{2^n} = P_{2^n} \left(\bigcirc_{i=1}^n [(X \otimes \mathbb{1}_{2^{n-i}}) \oplus \mathbb{1}_{2^n - 2^{n-i+1}}] \right) \cdot P_{2^n} \quad (3.17)$$

where X is the NOT gate, $\mathbb{1}_{2^j}$ is a $2^j \times 2^j$ dimensional identity matrix, and P_{2^n} is the qubit reversal permutation introduced in Sect. 3.3.1.

A quantum circuit for Q_{2^n} is given in Fig. 3.17. This instance is specialized to the case $n = 5$ but the generalization to arbitrary n is obvious.

To understand how this circuit is constructed, it is instructive to follow the argument for a small value of n . For example, taking $n = 4$, we can rewrite the expression $(X \otimes \mathbb{1}_{2^{n-i}}) \oplus \mathbb{1}_{2^n - 2^{n-i+1}}$ for $i = 1, 2, 3, 4$ as follows:

$$\begin{aligned} i = 1: \quad & (X \otimes \mathbb{1}_{2^{4-1}}) \oplus \mathbb{1}_{2^n - 2^{n-1+1}} \\ &= (X \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1}) \\ i = 2: \quad & (X \otimes \mathbb{1}_{2^{4-2}}) \oplus \mathbb{1}_{2^n - 2^{n-2+1}} \\ &= (X \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1}) \cdot (\mathbb{1}_8 \oplus (X \otimes \mathbb{1} \otimes \mathbb{1})) \cdot (X \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1}) \\ &= (X \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1}) \cdot (\text{Controlled-}X \otimes \mathbb{1} \otimes \mathbb{1}) \cdot (X \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1}) \\ i = 3: \quad & (X \otimes \mathbb{1}_{2^{4-3}}) \oplus \mathbb{1}_{2^n - 2^{n-3+1}} \\ &= (X \otimes X \otimes \mathbb{1} \otimes \mathbb{1}) \cdot (\mathbb{1}_{12} \oplus (X \otimes \mathbb{1})) \cdot (X \otimes X \otimes \mathbb{1} \otimes \mathbb{1}) \\ &= (X \otimes X \otimes \mathbb{1} \otimes \mathbb{1}) \cdot (\text{Controlled-Controlled-}X \otimes \mathbb{1}) \\ &\quad \cdot (X \otimes X \otimes \mathbb{1} \otimes \mathbb{1}) \\ i = 4: \quad & (X \otimes \mathbb{1}_{2^{4-4}}) \oplus \mathbb{1}_{2^n - 2^{n-4+1}} \\ &= (X \otimes X \otimes X \otimes \mathbb{1}) \cdot (\mathbb{1}_{14} \oplus X) \cdot (X \otimes X \otimes X \otimes \mathbb{1}) \\ &= (X \otimes X \otimes X \otimes \mathbb{1}) \cdot \text{Controlled-Controlled-Controlled-}X \\ &\quad \cdot (X \otimes X \otimes X \otimes \mathbb{1}) \end{aligned} \quad (3.18)$$

Note that the multiply-controlled X gates have their control qubits negated so that the gate acts when the controls are in the $|0\rangle$ state. The structure of the circuit for Q_{2^n} is then apparent.

3.3.4 Quantum Permutations for Classical Microprocessor Design?

It is generally understood that, as we attempt to cram more computing power into less and less space, it is becoming more and more challenging to dissipate the heat that is generated as a side effect of computation. For example, you may already have seen chips that need to be cooled actively to handle the heat load they generate. Electronic engineers still have a few tricks left to solve the problem for the next few generations of microprocessors such as switching to new materials or new architectures. Nevertheless, as we discussed in Sect. 2.1.7, at some point they will inevitably run into the fundamental fact that there is an absolute minimum energy that must be dissipated by irreversible logic operations whenever information is lost or erased. At that time the final trick left available to them will be to switch to (classical) reversible logic as the basis for microprocessor design. However, given that the action of any reversible logic gate is merely to apply a permutation to its input bits, and given that such permutation gates can be implemented as quantum circuits, it is natural to wonder whether a *quantum* circuit implementation of classical reversible logic might offer any advantages over a purely classical reversible design?

There are clear differences. Whereas the classical circuit decompositions of classical reversible gates (i.e., classical permutation gates) may only employ other classical reversible gates, the quantum circuit decompositions of quantum permutation gates allow for the use of true quantum gates. This means that even if the inputs and outputs of classical and quantum permutation gates look similar, i.e., classical bit-strings in comparison to unentangled quantum computational basis states, *internal* to such circuits the manipulations going on can be dramatically different. In principle, one could imagine a quantum circuit implementation of a classical reversible gate, such that the inputs and outputs are always guaranteed to be classical bit-strings (or unentangled computational basis states) but internal to the circuit arbitrary quantum states may be used. Would such an architecture offer any additional advantages as a basis for implementation of classical reversible logic devices?

It is early days to say for sure but a few things are clear. We do know that any circuit for classical reversible logic is obliged to use only classical reversible gates within it. This is really quite a nasty limitation because it restricts the allowed factorizations of the overall reversible computation into simpler reversible operations. By contrast, allowing arbitrary 1-qubit and 2-qubit (say) quantum gates is considerably more freeing. In particular, it is known any *even*-permutation (i.e., one requiring an even number of transpositions) can be achieved in a circuit using only NOT/CNOT/TOFFOLI gates with no ancillae, whereas any *odd*-permutation can be achieved in a circuit using only NOT/CNOT/TOFFOLI gates, but must necessarily

use one ancilla. That is, the odd permutations can only be implemented reversibly if we allow extra storage. By comparison, it is easy to see that *any* permutation (even or odd) can be achieved in a circuit using only R_z , R_y , Ph and CNOT (or \sqrt{SWAP} or iSWAP) gates using *no* ancillae. In a spintronic implementation, for example, the R_z , R_y , and Ph would be single spin-rotations about the z -axis, x -axis and phase shifts respectively, and \sqrt{SWAP} would be the only 2-spin interaction needed and would be implemented using the spin-spin exchange interaction run for half the time required for a total spin exchange. Moreover, unlike the classical case, we do not need any 3-bit elementary gates (such as Toffoli and Fredkin gates) to have a universal gate set of spintronic reversible computing.

Thus, it is apparent that by relaxing the requirement to remain in the computational basis for all intermediate steps in a reversible circuit, one can indeed achieve more space efficient implementations of classical reversible logic. But whether the degree of advantage is sufficient to warrant such a radical change in architecture is questionable. Nevertheless, it is conceivable that quantum circuit implementations of classical reversible logic could be the first practical use of quantum gates. This is an intriguing prospect since it could provide a natural pathway by which quantum gates may be infused into the mainstream computer chip industry, and stimulate the marriage of (say) spintronic or photonic logic devices and conventional microprocessor technology. As reversible gates are the ultimate energy-efficient classical gates, this could be useful for future generations of classical computer chips, and could provide a stepping stone to a full quantum computer, but with perhaps more forgiving thresholds on error correction since the starting and ending states would always have to be computational basis states even prior to measurements being made.

3.4 Quantum Fourier Transform: QFT

A periodic signal is a continuous stream of data values that eventually repeat after some point. Such signals are commonplace in a wide variety of fields ranging from the sciences, medicine, engineering, economics, finance, and applied mathematics. For centuries mathematicians have striven to understand the nature of periodic signals and have devised many techniques for extracting useful information from them. One of the most useful of these techniques is that of *Fourier analysis*.

Fourier analysis transforms the periodic signal from the “time-domain” (i.e., from a sequence of data values that vary over time) to the “frequency-domain” (i.e., to a sequence of data values that represent the relative contributions of different frequency components within the periodic signal). The underlying reason why Fourier analysis works is that any periodic function can, in principle, be expressed as a weighted sum of sines and cosines of different amplitudes and frequencies. Surprisingly, any weird shaped periodic function can be written in terms of sums of neat and regular sines and cosines having different frequencies. Knowledge of the relative contributions of sines and cosines of different frequencies to an unusually shaped periodic function can sometimes reveal useful information about the underlying process that generates the signal.

3.4.1 Continuous Signals as Sums of Sines and Cosines

Formally, we can write any function $f(t)$ as a sum of sines and cosines of different frequencies:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi t}{L}\right) + b_n \sin\left(\frac{n\pi t}{L}\right) \quad (3.19)$$

where

$$a_0 = \frac{1}{L} \int_{-L}^L f(t) dt \quad (3.20)$$

$$a_n = \frac{1}{L} \int_{-L}^L f(t) \cos\left(\frac{n\pi t}{L}\right) dt \quad (3.21)$$

$$b_n = \frac{1}{L} \int_{-L}^L f(t) \sin\left(\frac{n\pi t}{L}\right) dt \quad (3.22)$$

where $n = 1, 2, 3, \dots$

Thus any periodic signal can be viewed as a sum of sines and cosines of different amplitudes and frequencies. The highest frequency component present in such an expansion sets a limit to the rate with which the signal must be sampled in order to guarantee that the continuous signal can be reconstructed perfectly from knowledge of only a finite number of samples of it.

For example, consider the periodic signal shown in Fig. 3.18. This signal is periodic, with the period boundaries at $\{-L, +L\} = \{-2, +2\}$, and has a sharp discontinuity in its first derivative at, e.g., the point $t = 1$, making it quite unlike any individual sine or cosine function.² Nevertheless, we can approximate this periodic

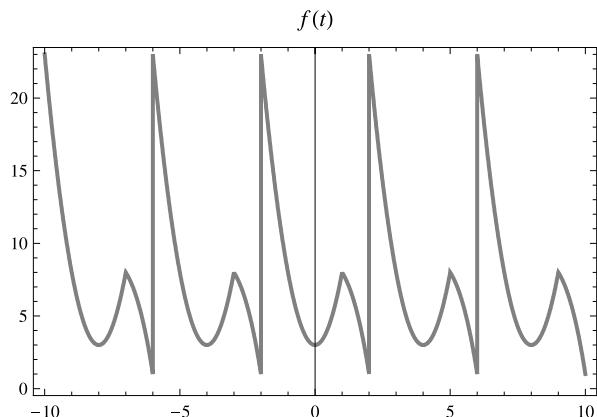


Fig. 3.18 Periodic signal showing a sharp kink. Here the *horizontal axis* is time, t , and the *vertical axis* is the signal value at that time, $f(t)$

²This data is synthetic and was generated using the piecewise continuous function $f(t) = 5t^2 + 3$ (for $-2 \leq t < 1$) and $f(t) = -t^3 + 9$ (for $1 \leq t \leq 2$) and shifted versions thereof.

signal quite well using a 10-th order truncated Fourier series given by:

$$\begin{aligned}
 f(t) \approx & -\frac{16(3 + 4\pi + \pi^2) \cos(\frac{\pi t}{2})}{\pi^4} + \frac{(12 - 5\pi^2) \cos(\pi t)}{2\pi^4} \\
 & -\frac{16(1 - 4\pi + 3\pi^2) \cos(\frac{3\pi t}{2})}{27\pi^4} + \frac{21 \cos(2\pi t)}{8\pi^2} \\
 & -\frac{16(3 + 20\pi + 25\pi^2) \cos(\frac{5\pi t}{2})}{625\pi^4} + \frac{(4 - 15\pi^2) \cos(3\pi t)}{54\pi^4} \\
 & -\frac{16(3 - 28\pi + 49\pi^2) \cos(\frac{7\pi t}{2})}{2401\pi^4} + \frac{21 \cos(4\pi t)}{32\pi^2} \\
 & -\frac{16(1 + 12\pi + 27\pi^2) \cos(\frac{9\pi t}{2})}{2187\pi^4} + \frac{(12 - 125\pi^2) \cos(5\pi t)}{1250\pi^4} \\
 & -\frac{2(24 - 44\pi - 13\pi^2 + 11\pi^3) \sin(\frac{\pi t}{2})}{\pi^4} + \frac{(-19 + 11\pi^2) \sin(\pi t)}{\pi^3} \\
 & -\frac{2(-8 - 44\pi + 39\pi^2 + 99\pi^3) \sin(\frac{3\pi t}{2})}{27\pi^4} + \frac{(-3 + 44\pi^2) \sin(2\pi t)}{8\pi^3} \\
 & -\frac{2(24 - 220\pi - 325\pi^2 + 1375\pi^3) \sin(\frac{5\pi t}{2})}{625\pi^4} + \frac{(-19 + 99\pi^2) \sin(3\pi t)}{27\pi^3} \\
 & -\frac{2(-24 - 308\pi + 637\pi^2 + 3773\pi^3) \sin(\frac{7\pi t}{2})}{2401\pi^4} + \frac{(-3 + 176\pi^2) \sin(4\pi t)}{64\pi^3} \\
 & -\frac{2(8 - 132\pi - 351\pi^2 + 2673\pi^3) \sin(\frac{9\pi t}{2})}{2187\pi^4} + \frac{(-19 + 275\pi^2) \sin(5\pi t)}{125\pi^3} \\
 & + \frac{117}{16}
 \end{aligned} \tag{3.23}$$

You can see that even the truncated Fourier series gives a pretty good approximation to the function by laying the graph of the Fourier series on top of that of the original signal as shown in Fig. 3.19. The approximation gets better and better the more terms from the Fourier series you include.

3.4.2 Discrete Signals as Samples of Continuous Signals

In practical applications, where we are monitoring some signal, we general do not know the exact functional form for the signal. Instead, we are obliged to work with a finite set of samples of the true signal spaced at regular time intervals. Such as signal is therefore a discretization of the true (but unknown) underlying continuous signal. We can however, adapt the idea of the Fourier transform to such a discrete case. The result is called the *discrete* Fourier transform (or “DFT”).

Fig. 3.19 Truncated 10th order Fourier series of the periodic signal shown in Fig. 3.18. Note that an appropriately weighted sum of sine and cosines of different frequencies approximates the given signal quite well. Here the horizontal axis is time, t , and the vertical axis is the signal value at that time, $f(t)$

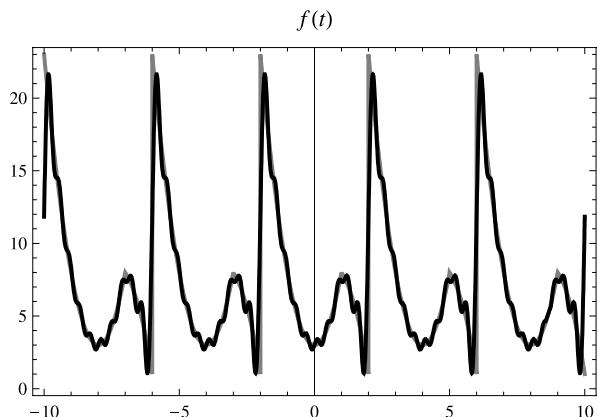
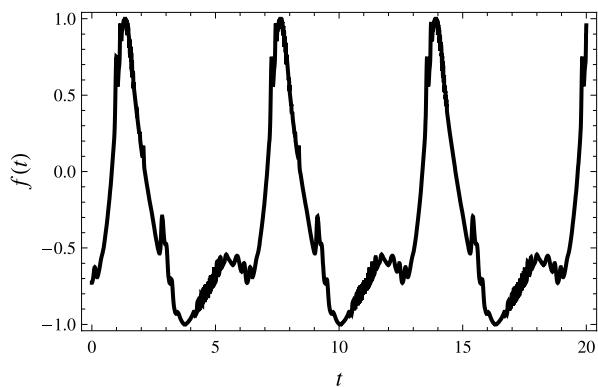


Fig. 3.20 A natural signal typically has structure on many scales simultaneously. This signal shows a periodic behavior with interesting higher frequency structure imposed upon a lower frequency oscillation



Naturally occurring signals typically have structure on many different scales. For example, the signal shown in Fig. 3.20 consists of a lower frequency oscillation on top of which is added many higher frequency components.

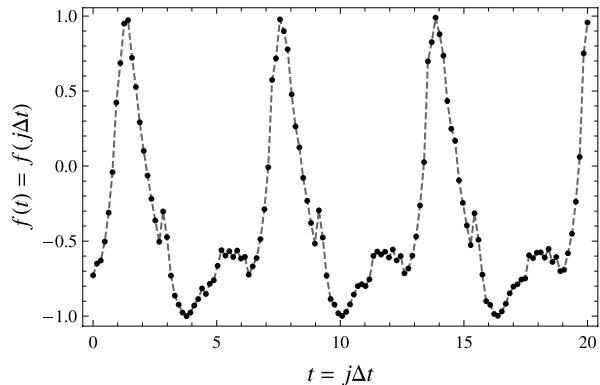
In general, we see not this entire continuous signal, but rather a finite set of samples from it over some time interval of interest. We can imagine what we are given is the values of some underlying continuous function $f(t)$ sampled at the discrete times $j\Delta t$ for $j = 0, 1, 2, \dots$ taking on integer values ranging from 1 to N (the number of sample points in the chosen interval) and Δt the spacing between samples, i.e., $(t_{\max} - t_{\min})/(N - 1)$. This gives us a snapshot of the function values at discrete time instants:

$$f(t) \approx \{f(0), f(1\Delta t), f(2\Delta t), \dots, f((N-1)\Delta t)\} \quad (3.24)$$

As an example, consider the signal in Fig. 3.20 sampled at 128 points spaced $\Delta t = (t_{\max} - t_{\min})/(N - 1) = (20 - 0)/(128 - 1)$ seconds apart.

Surprisingly, even though we may be given only a finite number of samples of a continuous signal, $f(t)$, at discrete times, it can nevertheless be re-created *perfectly*

Fig. 3.21 The same signal as shown in Fig. 3.20 sampled uniformly at 128 points



from such samples provided the sampling rate used is above a certain threshold, called the Nyquist limit. In particular, the following holds:

Nyquist Criterion If a signal $f(t)$ does not contain any frequencies greater than or equal to w Hz, it is completely determined from knowledge of its samples at evenly spaced times $\frac{1}{2w}$ seconds apart.

3.4.3 Discrete Signals as Superpositions

Assuming we have a discrete set of values of some signal, if we are to operate on them with a quantum Fourier transform we need to relate these signal values to a quantum state upon which our quantum Fourier transform operator is to act. To do so, we associate the different time points, $j\Delta t$, with different eigenstates, $|j\rangle$, and thereby encode all the signal values at the sampled times in a superposition such that the signal values are the amplitudes and the time points are the corresponding eigenstate. Hence, a discrete set of samples of a signal $f(t) \approx \{f(0), f(1\Delta t), f(2\Delta t), \dots, f((N-1)\Delta t)\}$ can be encoded as the state $|\psi_{\text{signal}}\rangle$ where:

$$\begin{aligned} |\psi_{\text{signal}}\rangle &= f(0)|0\rangle + f(1\Delta t)|1\rangle + f(2\Delta t)|2\rangle + \dots + f((N-1)\Delta t)|N-1\rangle \\ &= \sum_{j=0}^{N-1} f(j\Delta t)|j\rangle \end{aligned} \tag{3.25}$$

If we are using qubits, it is convenient to take the number of sample points, N , to be a power of two, i.e. $N = 2^n$. So henceforth we will assume $N = 2^n$.

In addition, for $|\psi_{\text{signal}}\rangle$ to be a valid quantum state, it must be normalized, i.e., $\sum_{\ell=0}^{N-1} |f(\ell\Delta t)|^2 = 1$. Therefore, if the signal values do not happen to have this property naturally, we simply re-normalize them by dividing the amplitudes by the re-normalization factor $\sqrt{\sum_{j=0}^{N-1} |f(j\Delta t)|^2}$ (see Fig. 3.22). If we

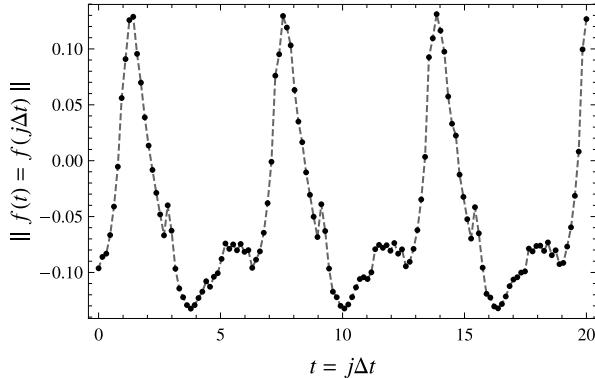


Fig. 3.22 The same signal as shown in Fig. 3.20 re-normalized and sampled uniformly at 128 points at intervals of $\Delta t = (t_{\max} - t_{\min})/(N - 1)$, where t_{\max} and t_{\min} define the time interval over which the function is being analyzed and $N = 128$ is the number of discrete points at which the function is sampled within this interval. In our case, we took $t_{\max} = 20$ and $t_{\min} = 0$, and $N = 128$. The effect of the renormalization is to scale the function values so that the sum of the squares of the function values at the sample points is unity. This allows us to encode the 128 samples within an $n = \log_2 32 = 7$ qubit quantum state

build this re-normalization step into our encoding, a finite set of samples $f(t) \approx \{f(0), f(1\Delta t), f(2\Delta t), \dots, f((N-1)\Delta t)\}$ will be encoded as the quantum superposition:

$$|\psi_{\text{signal}}\rangle = \sum_{j=0}^{N-1} \frac{f(j\Delta t)}{\sqrt{\sum_{\ell=0}^{N-1} |f(\ell\Delta t)|^2}} |j\rangle \quad (3.26)$$

In the following sections, for convenience, we will assume our signal samples are already properly normalized.

3.4.4 QFT of a Computational Basis State

Having understood how to represent a signal in a quantum state, we are ready to compute the quantum Fourier transform (QFT) of that state. The rule for transforming quantum state vectors under the QFT is exactly the same as the rule for transforming classical vectors under the DFT. We simply imagine a quantum state as being defined by a column vector of amplitudes representing the (re-normalized) signal values. Thus the component of this column vector, corresponding to eigenstate $|j\rangle$ is defined to transform, under the action of the QFT, as follows:

$$\text{QFT}_{2^n} |j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i \frac{jk}{2^n}} |k\rangle \quad (3.27)$$

Note that the output state now has amplitude smeared across all 2^n eigenstates and the quantity $\frac{j}{2^n}$ is a rational number $0 \leq \frac{j}{2^n} < 1$. This property is important in understanding the phase estimation and eigenvalue estimation algorithms in Chap. 9.

Furthermore, since j is a base-10 integer, we can re-define the QFT in terms of its affect of the individual bits which comprise the representation of j in base-2 notation. Specifically, if $(j)_{10} \equiv (j_1 j_2 j_3 \dots j_n)_2$, i.e., if j in base-10 is equivalent to the n -bit binary string $j_1 j_2 j_3 \dots j_n$ in base-2, we have:

$$\begin{aligned}(j)_{10} \equiv (j_1 j_2 j_3 \dots j_n)_2 &= (2^{n-1} j_1 + 2^{n-2} j_2 + \dots + 2^0 j_n)_{10} \\ &= 2^n (2^{-1} j_1 + 2^{-2} j_2 + \dots + 2^{-n} j_n)_{10} = 2^n (0.j_1 j_2 \dots j_n)_2\end{aligned}$$

where $0.j_1 j_2 \dots j_n$ is a binary fraction. Using this notation we can re-represent the affect of the QFT as follows:

$$\begin{aligned}\text{QFT}_{2^n}|j\rangle &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i \frac{jk}{2^n}} |k\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^1 \sum_{k_2=0}^1 \dots \sum_{k_n=0}^1 e^{2\pi i j (\sum_{\ell=1}^n k_\ell 2^{-\ell})} |k_1 k_2 \dots k_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^1 \sum_{k_2=0}^1 \dots \sum_{k_n=0}^1 \bigotimes_{\ell=1}^n e^{2\pi i j k_\ell 2^{-\ell}} |k_\ell\rangle \\ &= \frac{1}{\sqrt{2^n}} \bigotimes_{\ell=1}^n \left(\sum_{k_\ell=0}^1 e^{2\pi i j k_\ell 2^{-\ell}} |k_\ell\rangle \right) \\ &= \frac{1}{\sqrt{2^n}} \bigotimes_{\ell=1}^n (|0\rangle + e^{2\pi i j 2^{-\ell}} |1\rangle) \\ &= \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2\pi i 0.j_n} |1\rangle) (|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle) \\ &\quad \dots (|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n} |1\rangle)\end{aligned}\tag{3.28}$$

In this form it is apparent that the output state from the QFT of a computational basis state is a direct product of single qubit states and is, therefore, *unentangled*! This is a rather surprising property given how complicated the transformation appears to be. Nevertheless, the QFT of a computational basis state is unentangled. This alternative representation of the QFT finds application in many quantum algorithms based on phase estimation, where the binary fraction $0.j_1 j_2 \dots j_n = \frac{j}{2^n}$ is a binary encoding of a numerical phase factor that we wish to extract from a superposition. We will come back to this issue when we discuss the phase estimation and eigenvalue estimation algorithms in Chap. 9.

3.4.5 QFT of a Superposition

Using the definition of how the QFT is to transform a single computational basis state, $|j\rangle$, we can now use the linearity of quantum operations to predict how the QFT will transform an arbitrary superposition of computational basis states. In particular, we have:

$$|\psi_{\text{signal}}\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} f(j\Delta t) |j\rangle \xrightarrow{\text{QFT}} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \sum_{j=0}^{2^n-1} f(j\Delta t) e^{2\pi i \frac{jk}{2^n}} |k\rangle \quad (3.29)$$

Thus, if we encode a signal in the amplitude sequence of a quantum state, we can compute the DFT of the signal by applying QFT operator to this quantum state. The result will be a new state vector that will be peaked in probability amplitude at frequencies (indexed by computational eigenstates) which contribute the most strongly to the signal.

Note that in any QFT the peaks are arranged symmetrically. That is, as shown in Fig. 3.23, if eigenstate $|k\rangle$ in the QFT transformed signal is peaked, then so too will be the eigenstate $|2^n - k\rangle$. This is a normal feature of the discrete Fourier transform and happens in both the classical and quantum contexts. As an example, the QFT of the signal shown in Fig. 3.21 has decreasing peaks at computational eigenstates $|3\rangle$ (and the symmetric $|128 - 3\rangle = |125\rangle$), $|6\rangle$ (and the symmetric state $|128 - 6\rangle = |122\rangle$), and $|4\rangle$ (and its symmetric cousin $|128 - 4\rangle = |124\rangle$).

The QFT is a very important transform. Most of the known quantum algorithms showing an exponential speedup, including Shor's algorithm [455, 458], the phase and eigenvalue estimation algorithms [2], and the quantum counting algorithm [76], depend upon the QFT. Moreover, you can use the known (classical) relationships between the discrete Fourier transform (DFT) and other classical discrete transforms to infer corresponding relationships in the quantum domain [183]. This is potentially a source of new quantum transforms.

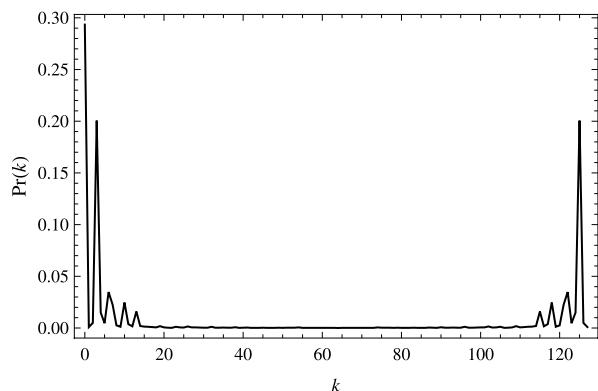


Fig. 3.23 Plot of probability (modulus amplitude squared) versus eigenstate in the QFT of the re-normalized signal shown in Fig. 3.22. Note the symmetric peaks in the QFT. If there is a peak at eigenstate $|k\rangle$ there will be a symmetric peak at eigenstate $|2^n - k\rangle$ where n is the number of qubits being acted upon

3.4.6 QFT Matrix

In classical computer science, the discrete Fourier transform (DFT) of a signal is usually computed by way of a matrix-vector operation. In this approach, a vector—the “signal *vector*”—encodes a sequence of sampled data values, and the elements of the DFT matrix encode components of the Fourier transform. These components are arranged in the matrix so that the dot product of the DFT matrix with the data vector computes the DFT of the signal vector.

As luck would have it the DFT matrix happens to be a unitary matrix. Thus, if we imagine representing a signal in the sequence of amplitudes of a quantum state, $|\psi_{\text{signal}}\rangle$,—the “signal *state*”—the quantum Fourier transform of the signal state would require us to apply exactly the same matrix that the discrete Fourier transform applies to the signal vector.

Thus, the QFT transformation specified in (3.29) can be represented, alternatively, as the unitary matrix QFT_{2^n} defined in such a way that $\text{QFT}_{2^n}|\psi_{\text{signal}}\rangle$ performs the QFT on the state vector $|\psi_{\text{signal}}\rangle$. For things to work our correctly, the elements of this QFT matrix need to be $\text{QFT}_{2^n} = \frac{1}{\sqrt{2^n}}\{\omega^{jk}\}_{j,k=0,\dots,(2^n-1)}$ where ω is the 2^n -th root of unity,³ i.e., $\omega = \exp(2\pi i/N)$ and $i = \sqrt{-1}$.

$$\text{QFT}_{2^n} := \frac{1}{\sqrt{2^n}} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega^1 & \omega^2 & \dots & \omega^{(2^n-1)} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(2^n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{(2^n-1)} & \omega^{(2^n-1)2} & \dots & \omega^{(2^n-1)(2^n-1)} \end{pmatrix} \quad (3.30)$$

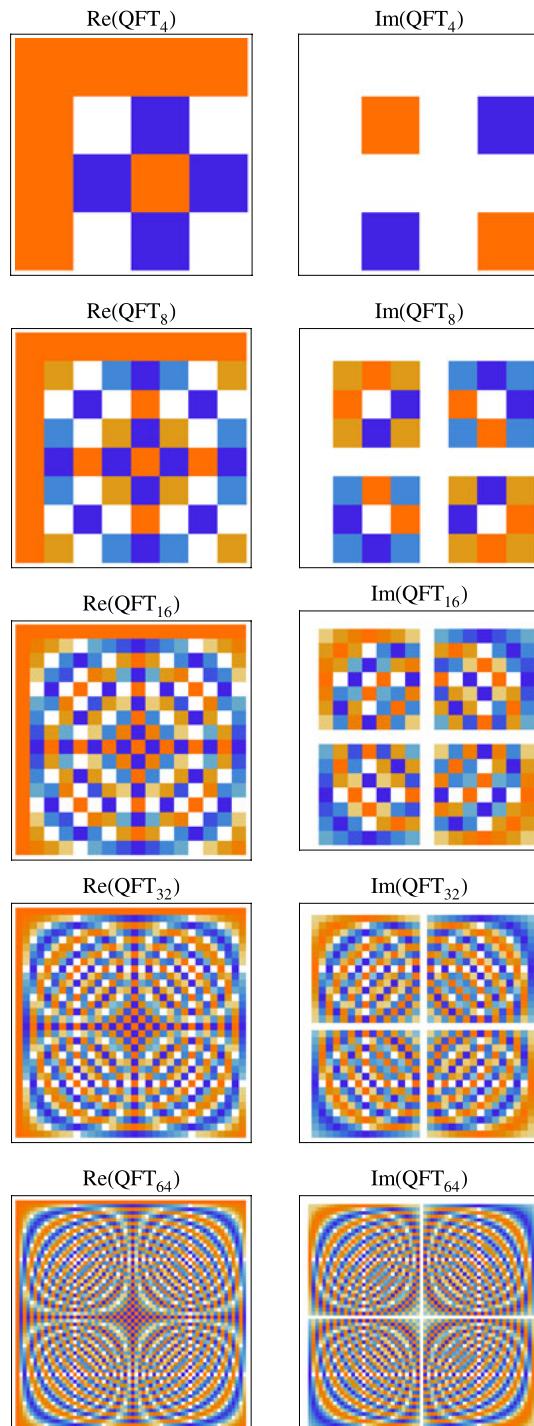
The QFT matrix is highly structured as can be seen from the visual representation depicted in Fig. 3.24. The matrices in the left hand column depict the real part of the QFT matrix and those in the right hand column the corresponding imaginary parts. In each pair, the matrices are shaded so that $-\frac{1}{\sqrt{2^n}} \rightarrow$ “orange”, $0 \rightarrow$ “white”, and $+\frac{1}{\sqrt{2^n}} \rightarrow$ “blue”.

This special structure allows the QFT matrix, QFT_{2^n} , to be implemented in a quantum circuit whose number of gates that grows only *polynomially* in n . This is *exponentially* more compact than the minimum size quantum circuit needed to implement an arbitrary unitary matrix of the same dimensions, i.e., $2^n \times 2^n$. This economy of gate count in implementation is critically important to achieving efficient quantum algorithms.

Note that, as the QFT matrix is unitary, no information is lost in taking the Fourier transform of a signal, because the signal can always be recovered by applying the inverse QFT matrix.

³Note that it is purely a matter of convention whether we pick $\omega = \exp(+2\pi i/N)$, or $\omega = \exp(-2\pi i/N)$ since $\exp(+2\pi i/N)^N = \exp(-2\pi i/N)^N = 1$. Physicists tend to use the former and electrical engineers the latter. The two versions of the transform are the inverse of one another. It does not matter which version we pick so long as we use it consistently.

Fig. 3.24 The real (left) and imaginary (right) parts of the 2-qubit to 6-qubit QFT matrices. The patterning reveals that the QFT matrices are highly structured, allowing them to be implemented far more efficiently than random, maximally general, unitary matrices of the same size



3.4.7 QFT Circuit

A quantum circuit for the QFT can be obtained from the factorization given by (3.31). This shows, if the input to the QFT is a computational basis state, i.e., an input of the form $|j_1 j_2 \dots j_n\rangle = |(j_1 j_2 \dots j_n)_2\rangle$, then the output will be an unentangled product state:

$$\text{QFT}_{2^n}|j\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i 0 \cdot j_n}|1\rangle)(|0\rangle + e^{2\pi i 0 \cdot j_{n-1} j_n}|1\rangle) \cdots (|0\rangle + e^{2\pi i 0 \cdot j_1 j_2 \dots j_n}|1\rangle) \quad (3.31)$$

Thus, the relative phase of each output qubit is controlled by the bit values of a subset of the input bits $|j_1 j_2 \dots j_n\rangle$. These can be determined via the quantum circuit shown in Fig. 3.25. In the QFT circuit,

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (3.32)$$

is the Walsh-Hadamard gate and

$$R_n = \begin{pmatrix} 1 & 0 \\ 0 & \exp(2\pi i / 2^n) \end{pmatrix} \quad (3.33)$$

is a 1-qubit gate that inserts a relative phase shift of $\exp(2\pi i / 2^n)$ between the $|0\rangle$ and $|1\rangle$ components of a qubit. The backwards inserted controlled- R_n gates can be obtained from the normally inserted controlled- R_n gates (i.e., $(\mathbb{1} \oplus R_n)$) in conjunction with SWAP gates and P_{2^n} gates. For example, the 2-qubit QFT showing these

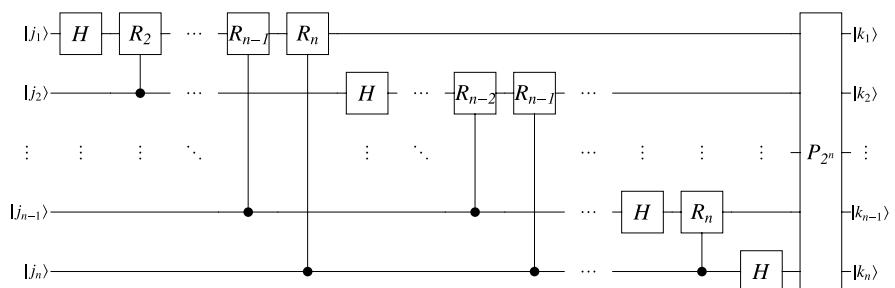


Fig. 3.25 Quantum circuit for the Quantum Fourier Transform (QFT). If the input is a computational basis state $|j_1 j_2 \dots j_n\rangle$ the output will be an unentangled product state $|k_1\rangle|k_2\rangle\dots|k_n\rangle$ where $|k_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i (0 \cdot j_n)}|1\rangle)$, $|k_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i (0 \cdot j_{n-1} j_n)}|1\rangle)$, and so on until $|k_n\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i (0 \cdot j_1 j_2 \dots j_n)}|1\rangle)$

extra embeddings explicitly is as follows:

$$\begin{aligned}
 U_1 &= (H \otimes \mathbb{1}) \\
 U_2 &= \text{SWAP}_{1,2;2} \cdot (\mathbb{1} \oplus R_2) \cdot \text{SWAP}_{1,2;2} \\
 U_3 &= (\mathbb{1} \otimes H) \\
 U_4 &= P_{2^2} \\
 \text{QFT}_{2^2} &= U_4 \cdot U_3 \cdot U_2 \cdot U_1
 \end{aligned} \tag{3.34}$$

Multiplying out the gates, you will find that the 2-qubit QFT circuit performs the following transformation:

$$\begin{aligned}
 \text{QFT}_{2^2}|j_1 j_2\rangle &= (|0\rangle + e^{2\pi i(j_2 2^{-1})}|1\rangle) \otimes (|0\rangle + e^{2\pi i(j_1 2^{-1} + j_2 2^{-2})}|1\rangle) \\
 &= (|0\rangle + e^{2\pi i(0.j_2)}|1\rangle) \otimes (|0\rangle + e^{2\pi i(0.j_1 j_2)}|1\rangle)
 \end{aligned} \tag{3.35}$$

Likewise, the 3-qubit QFT showing these embeddings explicitly is given by:

$$\begin{aligned}
 U_1 &= (H \otimes \mathbb{1} \otimes \mathbb{1}) \\
 U_2 &= \text{SWAP}_{1,2;3} \cdot ((\mathbb{1} \oplus R_2) \otimes \mathbb{1}) \cdot \text{SWAP}_{1,2;3} \\
 U_3 &= P_{2^3} \cdot \text{SWAP}_{1,2;3} \cdot (\mathbb{1} \otimes (\mathbb{1} \oplus R_3)) \cdot \text{SWAP}_{1,2;3} \cdot P_{2^3} \\
 U_4 &= (\mathbb{1} \otimes H \otimes \mathbb{1}) \\
 U_5 &= \text{SWAP}_{2,3;3} \cdot (\mathbb{1} \otimes (\mathbb{1} \oplus R_2)) \cdot \text{SWAP}_{2,3;3} \\
 U_6 &= (\mathbb{1} \otimes \mathbb{1} \otimes H) \\
 U_7 &= P_{2^3} \\
 \text{QFT}_{2^3} &= U_7 \cdot U_6 \cdot U_5 \cdot U_4 \cdot U_3 \cdot U_2 \cdot U_1
 \end{aligned} \tag{3.36}$$

Multiplying out the gates, you will see the 3-qubit QFT performs the transformation:

$$\begin{aligned}
 \text{QFT}_{2^3}|j_1 j_2 j_3\rangle &= (|0\rangle + e^{2\pi i(j_3 2^{-1})}|1\rangle) \otimes (|0\rangle + e^{2\pi i(j_2 2^{-1} + j_3 2^{-2})}|1\rangle) \\
 &\quad \otimes (|0\rangle + e^{2\pi i(j_1 2^{-1} + j_2 2^{-2} + j_3 2^{-3})}|1\rangle) \\
 &= (|0\rangle + e^{2\pi i(0.j_3)}|1\rangle) \otimes (|0\rangle + e^{2\pi i(0.j_2 j_3)}|1\rangle) \\
 &\quad \otimes (|0\rangle + e^{2\pi i(0.j_1 j_2 j_3)}|1\rangle)
 \end{aligned} \tag{3.37}$$

3.5 Quantum Wavelet Transform: QWT

The idea of a wavelet transform is to re-represent a signal or time-series as a sum of scaled and shifted versions of a fundamental function called the “mother wavelet”. The scaling and shifting is performed in such a way that the derived “daughter

wavelets” form an orthonormal basis⁴ for the set of all square integrable real valued functions. A wavelet decomposition of a signal is similar, therefore, to a Fourier decomposition in that we write the signal as a sum of *orthonormal* basis functions. In the Fourier decomposition these are the sines and cosines, but in a wavelet decomposition they are the daughter wavelets of some mother wavelet. However, unlike the Fourier decomposition the wavelet decomposition is not unique. There can be many different mother wavelets, leading to different daughter wavelets and hence wavelet representations of a given signal. Moreover, whereas sines and cosines are highly localized in frequency but spread out in space, the daughter wavelets are localized in both frequency *and* space, on scales different for each daughter. This locality property of wavelets, and the freedom to pick the mother wavelet, makes the wavelet representation ideal for describing aperiodic, and especially jagged, signals such as electrocardiograms, and seismic waves. With the appropriate choice of mother wavelet, a complicated signal can often be represented as a sum of just a handful of daughter wavelets, whereas its Fourier series may require dozens of terms. This makes the signal representation very sparse and helps accelerate signal processing operations.

3.5.1 Continuous Versus Discrete Wavelet Transforms

As in the case of Fourier transforms, there are both continuous and discrete versions of wavelet transforms. The main difference is that whereas the continuous wavelet transforms employ daughter wavelets that can be shifted and scaled over a continuum of values with respect to the mother wavelet, the discrete wavelet transform uses daughter wavelets that are shifted and scaled over only a discrete set of values. Of special interest to us is when such shifting and scaling operations are performed over powers of two. Thus, if the mother wavelet is the function, $\psi(x)$ (say), a family of the daughter wavelets, with scaling in powers of two, could be of the form $\psi_{jk}(x) = 2^{-j/2}\psi(2^{-j}x - k)$ where j and k are integers. Thus, any square integrable function, $f(x)$ can then be expanded in the form:

$$f(x) = \sum_{j,k} c_{jk} \psi_{jk}(x) \quad (3.38)$$

where

$$\psi_{jk}(x) = 2^{-j/2}\psi(2^{-j}x - k) \quad (3.39)$$

where j and k are integers and where the $c_{jk} = \int f(x)\psi_{jk}(x)dx$ are called wavelet coefficients.

⁴An orthonormal basis for a vector space is a set of vectors such that the overlap between any pair of distinct vectors is 0, i.e., $\langle \psi_i | \psi_j \rangle = 0$, if $i \neq j$, and the overlap of a vector with itself is 1, i.e., $\forall i, \langle \psi_i | \psi_i \rangle = 1$.

In this case the resulting family of discrete wavelet transforms can be represented as sparse $2^n \times 2^n$ -dimensional unitary matrices, which can be factored into quantum circuits whose size is polynomial in n . The wavelet transform of a signal can be affected via a matrix-vector operation in which the vector (containing 2^n samples of some signal) can be represented in terms of the sequence of 2^n amplitudes that define some n -qubit pure state. Thus, by focussing on the $2^n \times 2^n$ dimensional DWT matrix (from classical computing) we can make a very easy transition from the classical domain to the quantum one.

3.5.1.1 Daubechies Wavelets and Quadrature Mirror Filters

Of the many possible families of discrete wavelets, the family invented by Ingrid Daubechies in the late 1980s is especially useful [126–131]. Daubechies wavelets are orthogonal and have compact support, but they do not have a closed analytic form. Moreover, the lower order Daubechies wavelets are not differentiable everywhere and have a rather spiky appearance, whereas the higher order Daubechies wavelets are relatively smooth. To create a particular Daubechies wavelet one begins by designing a so-called “quadrature mirror filter”. In signal processing, a “filter” can be thought of as a transformation of each signal value, taking account of nearby signal values, and weighting the contributions mostly around the signal value being transformed. The precise way the weighting is done is controlled by a set of parameters called “wavelet filter coefficients”, which determine the type of mother wavelet.

Mathematically, one can model the action of a quadrature mirror filter as a “matrix-vector product” in which the “vector” is a column vector of signal values, and the matrix has rows whose elements correspond to the wavelet filter coefficients. A quadrature mirror filter uses *two* sets of inter-related wavelet filter coefficients, $\{c_0, c_1, c_2, c_3\}$ and $\{c_3, -c_2, c_1, -c_0\}$, which are designed so that one filter ($\{c_0, c_1, c_2, c_3\}$) gives a strong response to a smooth signal and a weak response to a rapidly varying signal, and the other filter ($\{c_3, -c_2, c_1, -c_0\}$) gives a strong response to a rapidly varying signal and a weak response to a smooth signal. These contrasting properties are the motivation behind the use of the term “mirror” in the name “quadrature mirror filter”. By embedding these quadrature mirror filters aligned in adjacent rows of a matrix, and then staggering the pattern two columns over each time it is repeated, we can make a filter that has the effect of partitioning an input signal into two bands. One band describes the slow frequency (smooth) behavior of the signal, whilst the other describes the high frequency (wiggly) behavior of signal. What makes this division worthwhile, is that we can often then go on to sub-sample each band separately, and thereby throw out much of the data from the original signal, without affecting our ability to reconstruct the original signal from the (now decimated) sub-bands to a very good approximation.

The different members of the family of Daubechies wavelet arise from different choices of quadrature mirror filters, which amount to different choices of mother wavelet. The simplest class of Daubechies wavelets are the Daubechies $D^{(4)}$

wavelets—so-called because they use *four* different parameters (called “wavelet filter coefficients”) in the quadrature mirror filter. Hence, the general structure of the Daubechies $D^{(4)}$ matrix is:

$$D^{(4)} \equiv \begin{pmatrix} c_0 & c_1 & c_2 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_0 & c_1 & c_2 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_0 & c_1 & c_2 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 & 0 & 0 \\ D^{(4)} \equiv & 0 & 0 & 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 \\ \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_0 & c_1 & c_2 & c_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_3 & -c_2 & c_1 & -c_0 \\ c_2 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_0 & c_1 \\ c_1 & -c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_3 & -c_2 \end{pmatrix} \quad (3.40)$$

Notice that the filters alternate from row to row, and step two columns to the right every other row. Moreover, as the $D^{(4)}$ matrices are applied, typically, to signals having 2^n samples, the final two rows always have a few wrap around elements in the bottom left corner of the matrix. The net effect of $D^{(4)}$ when applied to a column vector of signal values, is to interleave two filters—one that responds to smooth signals (and suppresses wiggly ones) and the other that responds to highly oscillatory signals (and suppresses smooth ones). Therefore, if we were to shuffle the elements in the transformed column vector, we could separate the signal into a description in terms of its smooth components and one in terms of its wiggly components. In the following discussion, if we know the $D^{(4)}$ wavelet kernel is to act on an n -qubit state, i.e., a column vector of 2^n amplitudes or on a $2^n \times 2^n$ dimensional density matrix, we indicate this with a subscript as in $D_{2^n}^{(4)}$.

The $D^{(4)}$ wavelet kernel is just one of the family of Daubechies wavelets. Other possibilities are the Daubechies $D^{(6)}, D^{(8)}, D^{(10)}, \dots, D^{(22)}$ wavelets,⁵ which as you might guess require 6, 8, 10, ..., 22 wavelet filter coefficients respectively.

3.5.2 Determining the Values of the Wavelet Filter Coefficients

So much for the *structure* of the $D^{(4)}$ wavelet kernel and its quadrature mirror filter. But what *values* are we to use for the wavelet filter coefficients, c_0, c_1, c_2, c_3 ?

It turns out that the values of the wavelet filter coefficients are determined by a set of constraint equations that follow from the properties we require our wavelet transform to possess. Specifically, if we are to be able to reconstruct a signal from

⁵N.B. The superscript is always an even number.

its wavelet transform, then the wavelet kernel matrix needs to be invertible. This is achieved by requiring the wavelet kernel matrix to be orthogonal. In this case, the inverse matrix is simply the transpose. Thus, if the 4-qubit $D^{(4)}$ wavelet kernel, $D_{2^4}^{(4)}$, is given by:

$$D_{2^4}^{(4)} = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_0 & c_1 & c_2 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_0 & c_1 & c_2 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_0 & c_1 & c_2 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_0 & c_1 & c_2 & c_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_0 & c_1 & c_2 & c_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_3 & -c_2 & c_1 & -c_0 \\ c_2 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_0 & c_1 \\ c_1 & -c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_3 & -c_2 \end{pmatrix} \quad (3.41)$$

its transpose is given by:

$$(D_{2^4}^{(4)})^T = \begin{pmatrix} c_0 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_2 & c_1 \\ c_1 & -c_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_3 & -c_0 \\ c_2 & c_1 & c_0 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ c_3 & -c_0 & c_1 & -c_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_2 & c_1 & c_0 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_3 & -c_0 & c_1 & -c_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_2 & c_1 & c_0 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_3 & -c_0 & c_1 & -c_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_2 & c_1 & c_0 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_3 & -c_0 & c_1 & -c_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_2 & c_1 & c_0 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_3 & -c_0 & c_1 & -c_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_2 & c_1 & c_0 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_3 & -c_0 & c_1 & -c_2 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.42)$$

and, for the matrix $(D_{2^4}^{(4)})^T$ to be the inverse of matrix $D_{2^4}^{(4)}$, we require $D_{2^4}^{(4)} \cdot (D_{2^4}^{(4)})^T = \mathbb{1}$, the identity matrix, which can only be true if the diagonal elements of $D_{2^4}^{(4)} \cdot (D_{2^4}^{(4)})^T$ are all 1's and off-diagonal elements are all zeroes. Taking the dot

product of $D^{(4)}$ and $(D^{(4)})^T$ we obtain a matrix of the form:

$$D_{2^4}^{(4)} \cdot (D_{2^4}^{(4)})^T = \begin{pmatrix} c_0^2 + c_1^2 + c_2^2 + c_3^2 & 0 & c_0c_2 + c_1c_3 & 0 & 0 & \dots \\ 0 & c_0^2 + c_1^2 + c_2^2 + c_3^2 & 0 & \ddots & 0 & \dots \\ c_0c_2 + c_1c_3 & 0 & c_0^2 + c_1^2 + c_2^2 + c_3^2 & 0 & \ddots & \dots \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots \end{pmatrix} \quad (3.43)$$

This matrix is supposed to be the identity matrix, which implies we need to satisfy the constraints:

$$c_0^2 + c_1^2 + c_2^2 + c_3^2 = 1 \quad (3.44)$$

$$c_0c_2 + c_1c_3 = 0 \quad (3.45)$$

In addition, for the Daubechies wavelets, we also want all moments up to p -th order to be zero. If we map the wavelet filter coefficients to the Fourier domain as in:

$$H(\omega) = \sum_{j=0}^{2\ell-1} c_j e^{ij\omega} \quad (3.46)$$

then the constraint on the moments up to p -th order being zero amounts to requiring:

$$\left. \frac{\partial H^\alpha(\omega)}{\partial \omega^\alpha} \right|_{\omega=\pi} = 0 \quad (3.47)$$

for $\alpha = 0, 1, 2, \dots, (p-1)$. In particular, for the Daubechies $D^{(4)}$ wavelets $p=2$ and the implied constraints are therefore:

$$H(\omega) = c_0 + c_1 e^{i\omega} + c_2 e^{i2\omega} + c_3 e^{i3\omega} \quad (3.48)$$

$$\left. \frac{\partial H(\omega)}{\partial \omega} \right|_{\omega=\pi} = ie^{i\omega} c_1 + 2ie^{2i\omega} c_2 + 3ie^{3i\omega} c_3 = 0 \quad (3.49)$$

$$\left. \frac{\partial H^2(\omega)}{\partial \omega^2} \right|_{\omega=\pi} = -e^{i\omega} c_1 - 4e^{2i\omega} c_2 - 9e^{3i\omega} c_3 = 0 \quad (3.50)$$

$$\left. \frac{\partial H^3(\omega)}{\partial \omega^3} \right|_{\omega=\pi} = -ie^{i\omega} c_1 - 8ie^{2i\omega} c_2 - 27ie^{3i\omega} c_3 = 0 \quad (3.51)$$

Solving the orthogonality constraints (as in (3.44)) and the moments constraint (as in (3.48)) for the c_j determines the values of the wavelet filter coefficients, $\{c_0, c_1, c_2, c_3\}$. In general, there are multiple sets of solutions for the c_j . One such

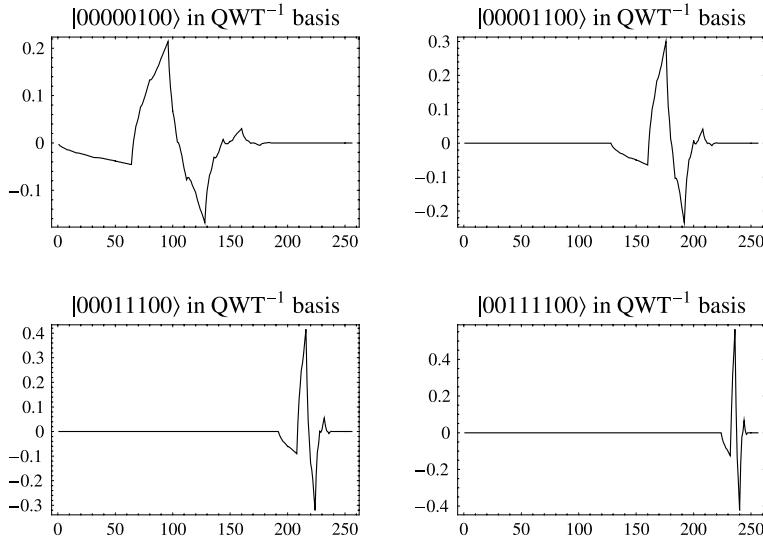


Fig. 3.26 One can visualize the shape of a wavelet by running a delta function through the inverse wavelet transform

solution is:

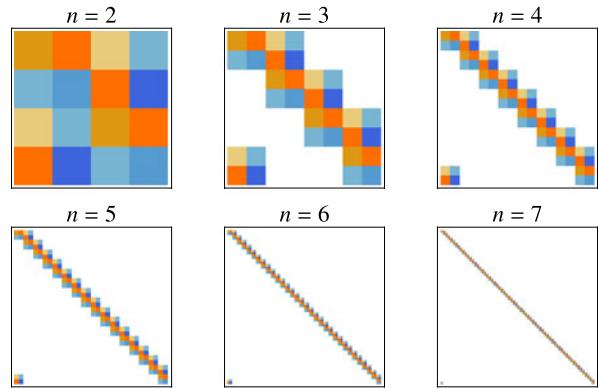
$$\begin{aligned} c_0 &= \frac{1 + \sqrt{3}}{4\sqrt{2}} \\ c_1 &= \frac{3 + \sqrt{3}}{4\sqrt{2}} \\ c_2 &= \frac{3 - \sqrt{3}}{4\sqrt{2}} \\ c_3 &= \frac{1 - \sqrt{3}}{4\sqrt{2}} \end{aligned} \tag{3.52}$$

Having found values for the wavelet filter coefficients, we can now see, as shown in Fig. 3.26, what wavelets look like by applying the *inverse* of the wavelet kernel matrix (given by (3.42)) to one of the computational basis states. Using the aforementioned values for the wavelet filter coefficients, Fig. 3.27 shows the structure if the $D_{2^n}^{(4)}$ wavelet kernel matrices for $n = 2$ to $n = 6$ qubits.

3.5.3 Factorization of Daubechies $D_{2^n}^{(4)}$ Wavelet Kernel

Next we turn to the factorization of the Daubechies $D_{2^n}^{(4)}$ wavelet kernel into 1-qubit and 2-qubit unitaries that are easily interpretable as quantum gates. We begin by

Fig. 3.27 The structure of the Daubechies quantum wavelet kernel transforms $D_{2^2}^{(4)}$, $D_{2^3}^{(4)}$, $D_{2^4}^{(4)}$ (top row) and $D_{2^5}^{(4)}$, $D_{2^6}^{(4)}$, $D_{2^7}^{(4)}$ (bottom row)



defining the single qubit gates C_0 and C_1 :

$$\begin{aligned} C_0 &= 2 \begin{pmatrix} c_3 & -c_2 \\ c_2 & c_3 \end{pmatrix} \\ C_1 &= \frac{1}{2} \begin{pmatrix} \frac{c_0}{c_3} & 1 \\ 1 & \frac{c_1}{c_2} \end{pmatrix} \end{aligned} \quad (3.53)$$

where the values of c_0, c_1, c_2 , and c_3 are as defined in (3.52). With these definitions, as you will show in Exercise 3.13, we can factor the Daubechies $D_{2^n}^{(4)}$ wavelet kernel matrix as:

$$D_{2^n}^{(4)} = (\mathbb{1}_{2^{n-1}} \otimes C_1) \cdot Q_{2^n} \cdot (\mathbb{1}_{2^{n-1}} \otimes (\text{NOT} \cdot C_0)) \quad (3.54)$$

where $\mathbb{1}_{2^{n-1}}$ is the $2^{n-1} \times 2^{n-1}$ dimensional identity matrix, and Q_{2^n} is the $2^n \times 2^n$ -dimensional downshift permutation matrix described in Sect. 3.3.3.

3.5.4 Quantum Circuit for $D_{2^n}^{(4)}$ Wavelet Kernel

To obtain the quantum circuit for the $D_{2^n}^{(4)}$ wavelet kernel, we can interpret the factorization given in (3.54). This gives $D_{2^n}^{(4)}$ in terms of Q_{2^n} and single qubit gates as shown in Fig. 3.28.

Expanding out the definition of Q_{2^n} we obtain the quantum circuit shown in Fig. 3.29.

3.5.5 Quantum Circuit for the Wavelet Packet Algorithm

As we mentioned above, a single application of the wavelet kernel transform, splits a signal into a coarse description and a fine description, but these two representations

Fig. 3.28 The high level factorization of the $D_{2^n}^{(4)}$ wavelet kernel in terms of the downshift permutation Q_{2^n} and single qubit gates

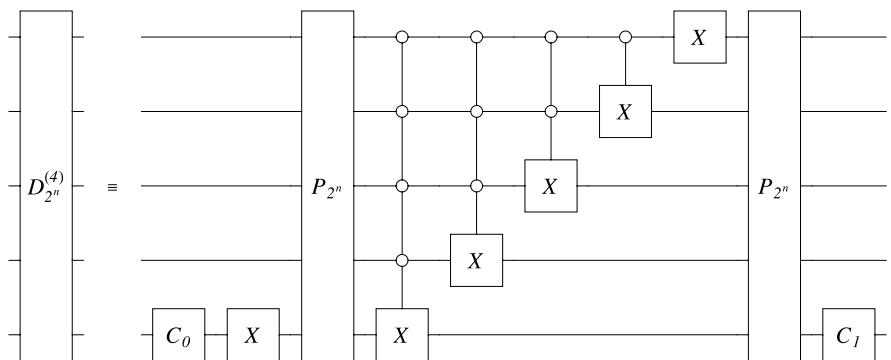
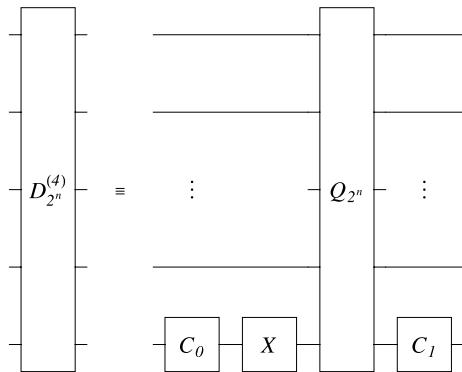
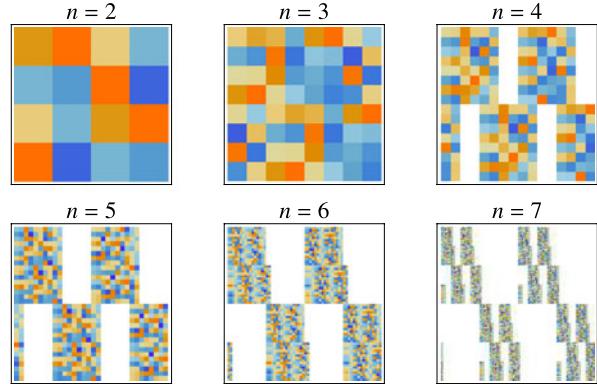


Fig. 3.29 Quantum circuit for the $2^n \times 2^n$ dimensional $D_{2^n}^{(4)}$ wavelet kernel

end up interleaved. In classical applications of the discrete wavelet transform, one therefore usually shuffles the transformed signal to group all the coarse components together, and all the fine components together, making two vectors half the length of the original. These vectors are called sub-band signals. One then repeats the process with new (half-length) discrete wavelet transforms applied to each sub-band independently. Depending on how you split and recurse on the transformed signals, one can achieve the so-called wavelet packet or wavelet pyramidal algorithms. In the quantum context, these turn out to be phenomenally more efficient than is possible classically.

A wavelet transform typically involves a wavelet kernel and a scheme for employing that kernel within a so-called “pyramid” or “packet” algorithm. The wavelet kernel splits a signal into a part describing its smooth behavior and a part describ-

Fig. 3.30 The structure of the Daubechies $D^{(4)}$ quantum wavelet packet transforms PAC_{2^2} , PAC_{2^3} , PAC_{2^4} (top row) and PAC_{2^5} , PAC_{2^6} , PAC_{2^7} (bottom row)



ing its detailed behavior, and then stops. Then other operations, i.e., the pyramid or packet operations, are used to permute the result, and then a wavelet kernel is used again on a smaller subset of the vector.

Once we have a quantum circuit for the quantum wavelet kernel, it is trivial to write the circuit for the quantum wavelet packet algorithm (based on this kernel) using the factorization:

$$\begin{aligned} \text{PAC} = & (\mathbb{1}_{2^{n-2}} \otimes D_4^{(4)}) \cdot (\mathbb{1}_{2^{n-3}} \otimes \Pi_8) \cdots (\mathbb{1}_{2^{n-i}} \otimes D_{2^i}^{(4)}) \cdot (\mathbb{1}_{2^{n-i-1}} \otimes \Pi_{2^{i+1}}) \\ & \cdots (\mathbb{1}_2 \otimes D_{2^{n-1}}^{(4)}) \cdot \Pi_{2^n} D_{2^n}^{(4)} \end{aligned} \quad (3.55)$$

because operators of the form $U \otimes \mathbb{1}$ apply U to one subset of qubits and the identity ($\mathbb{1}$) to the remaining ones. The structure of the resulting quantum wavelet packet matrices, based on the Daubechies $D^{(4)}$ wavelet kernel, are shown in Fig. 3.30.

3.5.6 Quantum Circuit Wavelet Pyramidal Algorithm

A wavelet kernel used within a pyramid algorithm splits a signal into a part describing its smooth behavior and a part describing its detailed behavior, shuffles the amplitudes to group all the smooth components together, and all the detail components together, and then recurses on the newly grouped smooth components (now half the length of the previous vector acted upon). This pyramid algorithm is best described by example. Suppose W is some wavelet kernel transform. Then W can be used within a wavelet pyramid algorithm as follows:

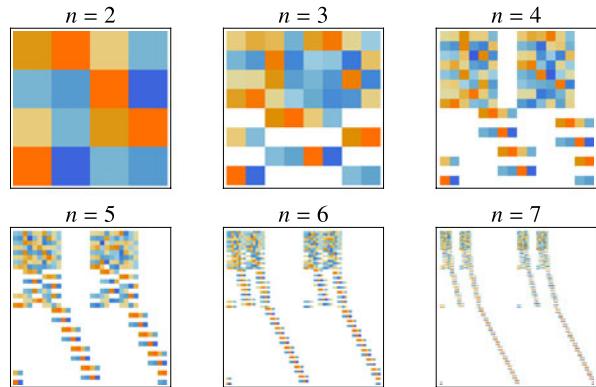
$$\begin{array}{c}
\left(\begin{array}{l} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{15} \end{array} \right) \xrightarrow{W_{16}} \left(\begin{array}{l} s_0 \\ d_0 \\ s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \\ s_4 \\ d_4 \\ s_5 \\ d_5 \\ s_6 \\ d_6 \\ s_7 \\ d_7 \end{array} \right) \xrightarrow{\Pi_{16}^T} \left(\begin{array}{l} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{array} \right) \xrightarrow{W_8} \left(\begin{array}{l} s'_0 \\ d'_0 \\ s'_1 \\ d'_1 \\ s'_2 \\ d'_2 \\ s'_3 \\ d'_3 \\ d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{array} \right) \xrightarrow{\Pi_8^T} \left(\begin{array}{l} s'_0 \\ s'_1 \\ s'_2 \\ s'_3 \\ d'_0 \\ d'_1 \\ d'_2 \\ d'_3 \\ d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{array} \right) \\
\\
\left(\begin{array}{l} s''_0 \\ d''_0 \\ s''_1 \\ d''_1 \\ d''_0 \\ d'_1 \\ d'_2 \\ d'_3 \\ d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{array} \right) \xrightarrow{W_4} \left(\begin{array}{l} s''_0 \\ s''_1 \\ d''_0 \\ d''_1 \\ d'_0 \\ d'_1 \\ d'_2 \\ d'_3 \\ d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{array} \right) \xrightarrow{\Pi_4^T} \left(\begin{array}{l} s'''_0 \\ d'''_0 \\ d''_0 \\ d''_1 \\ d'_0 \\ d'_1 \\ d'_2 \\ d'_3 \\ d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{array} \right) \xrightarrow{W_2} \left(\begin{array}{l} d'_3 \\ d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{array} \right)
\end{array} \tag{3.56}$$

The first level factorization of the wavelet pyramid algorithm is given by:

$$\begin{aligned}
\text{PYR} = & (D_4^{(4)} \oplus \mathbb{1}_{2^n-4}) \cdot (\Pi_8 \oplus \mathbb{1}_{2^n-8}) \cdots (D_{2^i}^{(4)} \oplus \mathbb{1}_{2^n-2^i}) \\
& \cdot (\Pi_{2^{i+1}} \oplus \mathbb{1}_{2^n-2^{i+1}}) \cdots \Pi_{2^n} D_{2^n}^{(4)}
\end{aligned} \tag{3.57}$$

The structure of the resulting quantum wavelet pyramid matrices, based on the Daubechies $D^{(4)}$ wavelet kernel, are shown in Fig. 3.31. Thus, just as we can obtain efficient quantum circuits for the quantum Fourier transform (QFT), so too can we obtain them for the quantum wavelet transform (QWT) as exemplified here by the particular case of Daubechies $D^{(4)}$ wavelet kernel and its subsequent use within both

Fig. 3.31 The structure of the Daubechies $D^{(4)}$ quantum wavelet pyramid transforms PYR $_{2^2}$, PYR $_{2^3}$, PYR $_{2^4}$ (top row) and PYR $_{2^5}$, PYR $_{2^6}$, PYR $_{2^7}$ (bottom row)



wavelet packet and wavelet pyramid algorithms. In all these circuit constructions permutation matrices play a pivotal role. If viewed from a conventional (classical) computer science perspective, such permutation matrices correspond to instructions specifying data movement patterns. Surprisingly, sometimes the data movement patterns that are hard to implement classically turn out to be easy to implement quantumly and vice versa. Moreover, perhaps completely counter-intuitively, the complexity of quantum circuits for the wavelet packet and wavelet pyramid schemes (which incorporate multiple calls to a wavelet kernel sub-routine) can, after simplification, be lower than the complexity of the wavelet kernels used within them. So in the quantum world one can sometimes do more with less! The discrete wavelet transform is so useful classically it is hard to believe we cannot also exploit the corresponding QWT in clever ways quantumly. Part of my motivation for including a discussion of the QWT in this book is to stimulate others to use it in creative and productive ways. Two-dimensional extensions of the QWT (and indeed, all other 1D quantum transforms) are discussed in Chap. 10.

3.6 Quantum Cosine Transform: QCT

In classical computing, the Discrete Cosine Transform (DCT) is used widely within algorithms for video, image, and audio compression [9, 287]. In particular, it is the cornerstone of the JPEG image, and MPEG video, compression schemes. The DCT's popularity comes from the fact that it is able to concentrate most of the information in a signal into a small number of low frequency components of the transformed signal. Hence, one need only send these few low frequency components to be able to reconstruct an image that is indistinguishable (by eye) from the original.

The DCT is similar to the DFT in that they both transform discretely sampled signals of finite duration or extent into new signals that reveal the frequency contributions to the original signal. However, whereas the foundation for DFT is based on the idea of Fourier series, the foundation of DCT comes from that of *cosine* series.

In a Fourier series one represents a signal of finite extent as a periodic function of infinite extent built from a sum of sinusoids and cosinusoids of different frequencies and amplitudes such that the function values match the signal values over each period. However, as the signal value at the beginning of a period is usually different from the signal value at the end of that period, it very likely that the periodic function used to represent the signal will have abrupt discontinuities at each period-to-period boundary. Due to these abrupt discontinuities it typically takes a great many sine and cosine terms in the Fourier series expansion to obtain a satisfactory approximation to the original signal.

3.6.1 Signals as Sums of Cosines Only

The cosine series is similar to the Fourier series except that it uses only cosine functions of different amplitudes and frequencies in the sum used to approximate a signal. As for Fourier series, the use of cosinusoids means that the function used to represent a signal of finite extent actually has infinite extent, and therefore has to be defined beyond the original domain of the signal. However, one has some flexibility in how one defines the function outside the signal domain. In particular, the extensions do not have to be periodic replications of the signal. In fact, if we use *discrete* samples of a continuous signal, we can choose to make the extension even or odd about an end point of the signal or about a point midway between an endpoint and the next point. Different types of DCT (called DCT-I, DCT-II, ..., DCT-VIII) come from making different choices about how to continue the signal from one domain to the next, and whether to make the symmetry be based on an end point or a point midway between an end point and the next point. In cosine series, the extended function is always chosen to be an even function on the left—because the cosine function is even—(1 choice) but may be an even or odd function on the right (2 choices). In addition, the point of symmetry on the left can be an end point or a point midway between the end point and the next point (2 choices). Likewise, the point of symmetry on the right can be an end point or a point midway between the end point and the next point (2 choices). Thus there are $1 \times 2 \times 2 \times 2 = 8$ possible ways to define the (functional) continuation of the original signal beyond its defined domain. These alternatives give rise eight variants of the DCT known DCT-I, DCT-II, ..., DCT-VIII.

3.6.2 Discrete Cosine Transform DCT-II and Its Relation to DFT

The most commonly used DCT is the DCT-II. This has boundary conditions such that the continuation of the discrete signal values, $\{x_0, x_1, \dots, x_{N-1}\}$, are made to

be an *even* function on the left *about the point* $n = -\frac{1}{2}$ and an *even* function on the right *about the point* $n = N - \frac{1}{2}$. The *classical* one-dimensional DCT-II is defined to implement the following transformation:

$$y_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right) \quad (3.58)$$

The unusual-looking factor of $\frac{1}{2}$ in the definition of DCT-II come from taking the symmetry points to be midway between end points and the next points in the extended signal in either direction.

In this form, there is an surprisingly simple relationship between DCT-II and DFT (the discrete Fourier transform). One finds that a DCT-II transform of a signal, $S = \{x_0, x_1, \dots, x_{N-1}\}$, having N sample values, is related to the DFT transformation of a signal $S' = \{0, x_0, 0, x_1, 0, \dots, x_{N-1}, 0, x_{N-1}, 0, x_{N-2}, 0, \dots, x_1, 0, x_0\}$, having $4N$ sample values. In particular, the first N elements of $\text{DFT} \cdot S'$ are exactly the same as $\text{DCT-II} \cdot S$. As DFT and QFT transforms are defined by identical matrices, this means there is a direct relationship between QFT and this version of the classical DCT-II. Amazing!

Does this mean we are seconds away from a fast quantum circuit for performing “QCT-II”, the quantum version of DCT-II? Well not so fast. Unfortunately, the DFT-II transform as defined in (3.58) is not orthogonal, and hence not unitary. So we cannot use QFT to obtain QCT-II in a straightforward way. However, there is an alternative way to define the classical DCT-II that inserts coefficients into (3.58) specifically to make the DCT-II transformation matrix orthogonal and unitary.

The (classical) orthogonalized version of DCT-II is defined by the transformation:

$$y_k = \sqrt{\frac{2}{N}} \alpha_k \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right) \quad (3.59)$$

such that $\alpha_0 = \frac{1}{\sqrt{2}}$ and for all $k \neq 0$, $\alpha_k = 1$. Unfortunately, if we use *this* definition of DCT-II, we no longer have the elegant relationship between the DCT-II and DFT that we had using the old non-orthogonalized version. Nevertheless, as the orthogonalized version of DCT-II given in (3.59) is unitary it is a much better starting point from which to attempt to construct its quantum counterpart QCT-II. Moreover, even though the simple relationship with DFT is lost, it turns out that the orthogonalized version of DCT-II can still be factored in terms of DFT (and hence QFT) and so the quantum circuit for QCT-II can still employ QFT in its construction.

3.6.3 QCT_N^{II} Transformation

We therefore choose to define the Type II quantum cosine transform acting on a signal having $N = 2^n$ sample values as the following transformation of a superposition representing a (normalized) signal $|\psi_{\text{signal}}\rangle$:

$$\begin{aligned} |\psi_{\text{signal}}\rangle &= \sum_{j=0}^{N-1} f_j |j\rangle \\ \text{QCT}_N^{\text{II}} &:= \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} \sqrt{\frac{2}{N}} \alpha_k \cos\left(\frac{\pi}{N}\left(j + \frac{1}{2}\right)k\right) f_j |j\rangle \end{aligned} \quad (3.60)$$

with $N = 2^n$, $\alpha_0 = \frac{1}{\sqrt{2}}$ and for all $k \neq 0$, $\alpha_k = 1$.

3.6.4 QCT_N^{II} Matrix

Given the definition of QCT_N^{II} in (3.60) the corresponding unitary matrix that implements this transformation is:

$$\text{QCT}_N^{\text{II}} := \left\{ \sqrt{\frac{2}{N}} \alpha_k \cos\left(\frac{\pi}{N}\left(j + \frac{1}{2}\right)k\right) \right\}_{j,k=0,1,2,\dots,N-1} \quad (3.61)$$

with $N = 2^n$, $\alpha_0 = \frac{1}{\sqrt{2}}$ and for all $k \neq 0$, $\alpha_k = 1$. This definition gives rise to highly structured unitary matrices for QCT_N^{II} transformations on increasing numbers of qubits. This structure is best revealed graphically as shown in Fig. 3.32.

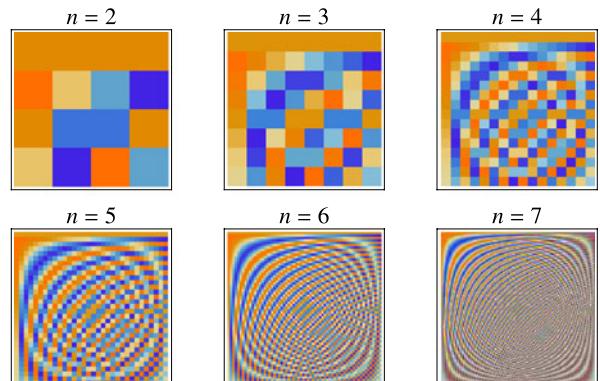


Fig. 3.32 The structure of the unitary matrices corresponding $\text{QCT}_{2^2}^{\text{II}}$, $\text{QCT}_{2^3}^{\text{II}}$, $\text{QCT}_{2^4}^{\text{II}}$ (*top row*) and $\text{QCT}_{2^5}^{\text{II}}$, $\text{QCT}_{2^6}^{\text{II}}$, $\text{QCT}_{2^7}^{\text{II}}$

3.6.5 QCT_N^{II} Circuit

A quantum circuit for QCT_N^{II} relies on the following identity:

$$U_{2N}^\dagger \cdot \text{QFT}_{2N} \cdot V_{2N} = \text{QCT}_N^{\text{II}} \oplus -\text{QST}_N^{\text{II}} \quad (3.62)$$

where QFT_{2N} is the Quantum Fourier Transform on a signal of length $2N$, U_{2N} and V_{2N} are $2N \times 2N$ dimensional unitary matrices that will be described below, and $-\text{QST}_N^{\text{II}}$ is the negative of the Type II Quantum Sine Transform, which is analogous to QCT_N^{II} except that it uses sines instead of cosines and always has an *odd* symmetry about the left hand boundary of the signal. Specifically, the unitary matrix describing the QST_N^{II} is:

$$\text{QST}_N^{\text{II}} := \left(\sqrt{\frac{2}{N}} \beta_k \sin\left(\frac{\pi}{N} \left(j + \frac{1}{2}\right)(k+1)\right) \right)_{j,k=0,1,2,\dots,N-1} \quad (3.63)$$

with $N = 2^n$, $\beta_{N-1} = \frac{1}{\sqrt{2}}$ and for all $k \neq N-1$, $\beta_k = 1$. Note that the direct sum on the right hand side of (3.62) implies it is a controlled gate, with the control value being on the topmost qubit. In this case, when the input state is $|1\rangle|\psi\rangle$, the output bottom qubits will contain the $\text{QCT}_N^{\text{II}}|\psi\rangle$.

The V_{2N} matrices are defined as:

$$V_4 = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \end{pmatrix}, \quad (3.64)$$

$$V_8 = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 \end{pmatrix}, \quad (3.65)$$

etc.

Likewise, the U_{2N} matrices are defined as:

$$U_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\bar{\omega}}{\sqrt{2}} & -\frac{i\bar{\omega}}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & -1 \\ 0 & \frac{\omega}{\sqrt{2}} & \frac{i\omega}{\sqrt{2}} & 0 \end{pmatrix}, \quad (3.67)$$

$$U_8 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\bar{\omega}}{\sqrt{2}} & 0 & 0 & -\frac{i\bar{\omega}}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{\bar{\omega}^2}{\sqrt{2}} & 0 & 0 & -\frac{i\bar{\omega}^2}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & \frac{\bar{\omega}^3}{\sqrt{2}} & 0 & 0 & -\frac{i\bar{\omega}^3}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & \frac{\omega^3}{\sqrt{2}} & 0 & 0 & \frac{i\omega^3}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{\omega^2}{\sqrt{2}} & 0 & 0 & \frac{i\omega^2}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{\omega}{\sqrt{2}} & 0 & 0 & \frac{i\omega}{\sqrt{2}} & 0 & 0 & 0 \end{pmatrix}, \quad (3.68)$$

$$U_{16} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\bar{\omega}}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{i\bar{\omega}}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\bar{\omega}^2}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & -\frac{i\bar{\omega}^2}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\bar{\omega}^3}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & -\frac{i\bar{\omega}^3}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\bar{\omega}^4}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & -\frac{i\bar{\omega}^4}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\bar{\omega}^5}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & -\frac{i\bar{\omega}^5}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\bar{\omega}^6}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & -\frac{i\bar{\omega}^6}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\bar{\omega}^7}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & -\frac{i\bar{\omega}^7}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\omega^7}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{i\omega^7}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\omega^6}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{i\omega^6}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\omega^5}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{i\omega^5}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\omega^4}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & \frac{i\omega^4}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\omega^3}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & \frac{i\omega^3}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\omega^2}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{i\omega^2}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\omega}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{i\omega}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.69)$$

where $\omega = \exp(2\pi i/4N)$ and $\bar{\omega} = \exp(-2\pi i/4N)$ is the complex conjugate of ω .

As efficient quantum circuits for the QFT_{2N} are known, we can obtain efficient quantum circuits any QCT_N^{II} if we can find efficient quantum circuits for U_{2N}^\dagger and V_{2N} . To this end, the following permutation matrices turn out to be useful:

3.6.5.1 Controlled-One's Complement

The “One’s Complement” of a computational basis state $|x_1, x_2, \dots, x_n\rangle$ is obtained by NOT-ing each bit individually. A Controlled-One’s Complement only performs the operation when a control bit is set to $|1\rangle$. Thus, we have:

$$|0, x_1, x_2, \dots, x_n\rangle \xrightarrow{\text{C-OC}} |0, x_1, x_2, \dots, x_n\rangle \quad (3.70)$$

$$|1, x_1, x_2, \dots, x_n\rangle \xrightarrow{\text{C-OC}} |1, 1-x_1, 1-x_2, \dots, 1-x_n\rangle \quad (3.71)$$

This operation can be obtained from:

$$\text{C-OC}_{2N} \equiv \bigcirc_{j=2}^{n=\log_2 2N} \text{CNOT}_{1,j;n} \quad (3.72)$$

where $\text{CNOT}_{1,j;n}$ is a CNOT gate between the first and j -th of n qubits. That is, this is simply a cascade of CNOT gates all having the control on the first qubit.

3.6.5.2 Controlled-Two's Complement

Let \mathbf{x} be the base-10 number corresponding to the n -qubit bit string x_1, x_2, \dots, x_n . “Two’s Complement” of $|\mathbf{x}\rangle$ is defined as follows:

$$\begin{aligned} |0, \mathbf{x}\rangle &\xrightarrow{\text{C-TC}} |0, \mathbf{x}\rangle \\ |1, \mathbf{0}\rangle &\xrightarrow{\text{C-TC}} |1, \mathbf{0}\rangle \\ |1, \mathbf{x}\rangle &\xrightarrow{\text{C-TC}} |1, 2^n - \mathbf{x}\rangle \end{aligned} \quad (3.73)$$

This transformation can be obtained from the unitary matrix:

$$\text{C-TC}_{2N} \equiv \mathbb{1}_N \oplus \left((\text{NOT}^{\otimes \log_2 N}) \cdot Q_N \right) \quad (3.74)$$

where Q_2, Q_4, Q_8, \dots are permutation matrices of the form:

$$Q_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (3.75)$$

$$Q_4 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad (3.76)$$

$$Q_8 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.77)$$

that rotate the sequence of amplitudes one position.

3.6.5.3 Controlled-ModularAddOne

$$|0, \mathbf{x}\rangle \xrightarrow{\text{C-MAO}} |0, \mathbf{x}\rangle \quad (3.78)$$

$$|1, \mathbf{x}\rangle \xrightarrow{\text{C-MAO}} |1, \mathbf{x} + 1 \bmod 2^n\rangle \quad (3.79)$$

This transformation can be obtained from the unitary matrix:

$$\text{C-MAO}_{2N} \equiv \mathbb{1}_N \oplus Q_N^\dagger \quad (3.80)$$

As you recall, the reason we are interested in the quantum arithmetic operations C-OC, C-TC, and C-MAO is because they arise in the factorization of the matrices V_{2N} and U_{2N}^\dagger that are, in turn, used in the factorization of the Type II quantum cosine transform QCT_N^{II} . In particular, we have:

$$U_{2N}^\dagger \cdot \text{QFT}_{2N} \cdot V_{2N} = \text{QCT}_N^{\text{II}} \oplus -\text{QST}_N^{\text{II}} \quad (3.81)$$

Thus, in any circuit implementing $U_{2N}^\dagger \cdot \text{QFT}_{2N} \cdot V_{2N}$, when the top (control) qubit is set to have the value $|0\rangle$, the alternative factorization on the right hand side of (3.81), allows us to see that the transformation the circuit will perform will then be QCT_N^{II} . This is because, if a unitary matrix of dimension $2^n \times 2^n$ can be written as a direct sum of two $2^{n-1} \times 2^{n-1}$ unitaries such as “ $A \oplus B$ ”, the resulting matrix will correspond to a conditional quantum gate which performs A on the bottom $n-1$ qubits when the top (control) qubit is set to $|0\rangle$, and B on the bottom $n-1$ qubits when the top (control) qubit is set to $|1\rangle$.

3.6.5.4 Quantum Circuit for V_{2N} Using C-OC_{2N}

By inspection, you can see that a V_{2N} matrix can be factored as:

$$V_{2N} = \text{C-OC}_{2N} \cdot (H \otimes \mathbb{1}_N) \quad (3.82)$$

where H is the 1-qubit Walsh-Hadamard gate, i.e., $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.

3.6.5.5 Quantum Circuit for U_{2N}^\dagger Using C-TC_{2N} and C-MAO_{2N}

The quantum circuit for U_{2N}^\dagger is considerably more involved. In Exercise 3.15 you are asked to verify the following factorization for U_{2N}^\dagger :

$$U_{2N}^\dagger = (\text{C-MAO}_{2N})^\dagger \cdot D_{2N} \cdot (\text{C-TC})^\dagger \cdot D1_{2N}(\omega) \quad (3.83)$$

where

$$\begin{aligned}
 D_{2N} &= P_{2N} \cdot (\text{NOT}^{\otimes(n-1)} \otimes \mathbb{1}) \cdot (\mathbb{1}_{2N-2} \oplus C) \cdot \text{NOT}^{\otimes(n-1)} \otimes \mathbb{1} \\
 &\quad \cdot P_{2N} \cdot (B \otimes \mathbb{1}_N) \\
 C &= \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{i}{\sqrt{2}} \\ -\frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \\
 B &= \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} & -\frac{i}{\sqrt{2}} \end{pmatrix} \\
 D1_{2N} &= \left[\begin{pmatrix} 1 & 0 \\ 0 & \bar{\omega} \end{pmatrix} \cdot (\Delta_1 \oplus \Delta_2) \right] \\
 \Delta_1 &= \bigotimes_{j=n}^1 L(j, \omega) \\
 \Delta_2 &= \bigotimes_{j=n}^1 K(j, \omega) \\
 L(j, \omega) &= \begin{pmatrix} 1 & 0 \\ 0 & \omega^{2^{j-1}} \end{pmatrix} \\
 K(j, \omega) &= \begin{pmatrix} \bar{\omega}^{2^{j-1}} & 0 \\ 0 & 1 \end{pmatrix} \\
 \omega &= \exp\left(\frac{2\pi i}{4N}\right)
 \end{aligned} \tag{3.84}$$

where $\bar{\omega}$ is the complex conjugate of ω . You should check this factorization for yourself by doing Exercise 3.15.

Using the given factorizations for V_{2N} , QFT_{2N} , and U_{2N}^\dagger , we can construct a complete factorization of the Type II Quantum Cosine Transform, QCT_N^{II} , and for that matter, the type II Quantum Sine Transform, QST_N^{II} from:

$$U_{2N}^\dagger \cdot \text{QFT}_{2N} \cdot V_{2N} = \text{QCT}_N^{\text{II}} \oplus -\text{QST}_N^{\text{II}} \tag{3.85}$$

Note that QCT_N^{II} and $-\text{QST}_N^{\text{II}}$ are both highly structured unitary transforms and admit anomalously small quantum circuit decompositions.

3.7 Circuits for Arbitrary Unitary Matrices

Automated circuit design tools are indispensable to the modern microelectronics industry. They span every layer of the design process including the logic-layer, architecture-layer, layout, fabrication etc. By contrast there are relatively few tools today to help quantum computing experimentalists in translating desired unitary and non-unitary operations into explicit quantum gate sequences.

Despite considerable effort being expended on trying to understand the principles of quantum circuit design we still do not know of any *efficient* method for finding the minimal quantum circuit that achieves a given n -qubit quantum computation for arbitrarily large n . To date, approaches to quantum circuit design have fallen into one of four categories. The majority of researchers use no formal scheme whatsoever but instead rely upon ad hoc trial and error, and human ingenuity, to arrive at a decomposition by hand. This approach is feasible for specially structured unitary matrices such as the Quantum Fourier Transform [34] and the quantum wavelet transform [183], because the special structure of the unitary operator reflects a regular structure in the corresponding quantum circuit.

A second approach is to exhaustively enumerate the space of possible circuit designs of increasing complexity starting from the empty circuit [149, 150]. For each topologically distinct circuit, a computer finds optimal values for the parameters of all parameterized-gates in the circuit. In principle, this method is guaranteed to find the smallest circuit sufficient to implement the desired unitary operator. However, exhaustive search composed with numerical optimization is computationally expensive because the number of possible quantum circuit topologies grows exponentially with increasing numbers of gates in the circuit. Hence the method is only feasible for unitary operators that in fact have compact circuit descriptions.

A third approach uses genetic algorithms [472, 473, 536]. A random population of circuits is created, and each is scored according to its “fitness” value, which is a measure of how closely it comes to achieving the desired unitary operator. Pairs of circuits are selected for breeding in proportion to their fitness and then mutation and crossover operations are applied to make a new generation of circuits. By iterating this process one converges on a population of circuits that tend towards implementing the desired unitary operator. For genetic algorithms to work well, one needs a degree of decomposability in the problem, i.e., that part of the solution is basically correct while ignoring the rest. Because of the way the direct product of matrices tends to spread elements throughout the resulting matrix, it can be hard for a genetic algorithm to find satisfactory circuits for highly entangling unitary operators. Nevertheless, several novel quantum circuits, algorithms and protocols have been found by genetic and evolutionary algorithms [36, 327, 328, 345, 470, 471, 474, 475, 480]. A good review of the evolutionary approach is [197].

The fourth and most systematic approach is to apply a recursive algebraic decomposition procedure such as the progressive matrix diagonalization of Reck [415], the “quantum Givens” operations of Cybenko [122] or the hierarchical CS decomposition of Tucci [498]. Algebraic factorization is guaranteed to work, but is likely to result in quantum circuits that are exponentially large unless one embeds circuit compactification rules within the decomposition procedure.

3.7.1 Uses of Quantum Circuit Decompositions

Quantum circuit decomposition of an arbitrary unitary matrix can be used to determine an optimal pathway for the direct synthesis of any pure or mixed quantum state [222], and to perfectly simulate high-dimensional stochastic processes that are hard to do faithfully using classical pseudo-random number generators [180, 184]. Moreover, in Grover's algorithm [219], if one has prior knowledge of the approximate location of the solution state(s) one can use a *biased* amplitude amplification operator which tends to pump probability amplitude preferentially into eigenstates in the presumed region of the solutions [221]. Such a unitary matrix may not have any special structure, making its quantum circuit hard to guess.

3.7.2 Choice of Which Gate Set to Use

Moreover, although the set of gates used in such quantum circuits has traditionally been taken to be the set of all one-qubit quantum gates in conjunction with CNOT, many equally good universal gate sets exist, and there might be advantages in using a non-standard gate set if certain choices happen to be easier to realize in one hardware context than another. For example, in the context of spin-based quantum computing, fractional powers of the two-qubit exchange interaction (i.e., the SWAP gate) are known to be as powerful as CNOT as far as computational universality is concerned. Likewise, in the context of charge-based quantum computing, the two-qubit gate iSWAP is easier to realize than CNOT and yet is equally as powerful computationally [163]. It makes sense therefore, to tailor the decomposition of a unitary operator to fit the chosen physical hardware, rather than to wrestle the physics to fit an ad hoc model of computation.

3.7.3 Circuit Complexity to Implement Arbitrary Unitary Matrices

What is the most general quantum gate operation that can be performed on an n -qubit state? If we imagine the n qubits to be well isolated from the environment, and to go unmeasured until after the operation is completed, then the most general operation corresponds to some n -qubit quantum gate which is mathematically equivalent to a $N \times N$ dimensional unitary matrix, where $N = 2^n$. In turn, this unitary matrix can be thought of as the matrix exponential of a maximally general n -qubit Hamiltonian, which can be represented as a $N \times N$ dimensional hermitian matrix. The fact that the Hamiltonian matrix needs to be hermitian constrains its elements along the main diagonal to be purely real numbers, but allows its off diagonal elements to be complex numbers such that $H_{ij} = H_{ji}^*$, where \star denotes taking the complex conjugate. A complex number takes two parameters to specify it (one for

the real part and one for the imaginary part). We can use this information to quantify how many free parameters go into specifying a maximally general quantum gate on n qubits. The Hamiltonian matrix is fully specified by picking N real numbers down the main diagonal, plus as many complex numbers as possible in the upper (or lower) triangular region above (or below) the main diagonal, i.e., $2 \sum_{i=1}^N N - i$. So overall we have $N + 2N(N - 1) = 2N^2 - N$. If we are free to pick $\mathcal{O}(2N^2) \approx 2^{2n}$ free parameters to specify a maximally general n -qubit unitary matrix, we ought not to be surprised if we have to use this many gates to implement such an operator. Indeed, Shende, Bullock and Markov have proved that the quantum circuit for a maximally general n -qubit unitary matrix requires at least $\frac{23}{48} 2^{2n} - \frac{3}{2} 2^n + \frac{4}{3}$ CNOT gates to implement it, and that this result cannot be improved by more than a factor of two [453]. Thus, it is a difficult problem and even writing down the circuit for a maximally general quantum gate will require exponential resources. Luckily, the unitary matrices that arise in practice are usually highly structured and admit anomalously compact quantum circuit decompositions. Nevertheless, for smallish circuits that defy obvious interpretation the use of an algebraic (always works) method in conjunction with circuit simplification rules can be the most expedient way to find a quantum circuit for a desired unitary matrix.

3.7.4 Algebraic Method

In this section, we describe a recursive algebraic scheme for constructing a quantum circuit decomposition of an arbitrary unitary operator, interleaved with circuit compactification rules that reduce the complexity of the final quantum circuit. The scheme starts with a similar mathematical decomposition to that used by Tucci [498], but uses different techniques for mapping the matrix factors into equivalent circuit fragments. Since Tucci's pioneering work two other groups have published algebraic decomposition engines for arbitrary unitary matrices along similar lines, [363, 453, 511] and Tucci has improved his compiler design further [499].

The essence of all these algebraic approaches is the following: first we decompose the $2^n \times 2^n$ dimensional unitary operator into a product of $2^n \times 2^n$ block-diagonal matrices, and direct sums of bit-reversal matrices (which need never be implemented explicitly). Next we map these block-diagonal matrices into corresponding quantum circuit fragments, each involves only one-qubit rotations about the y - and z -axes, one-qubit phase shifts, and a standard two-qubit gate, such as CNOT, the square root of SWAP ($\sqrt{\text{SWAP}}$), or iSWAP. One can pick whichever primitive two-qubit gate [147] one wants and obtain different quantum circuits accordingly. The last step is to join these quantum circuit fragments together, while again applying compactification rules to minimize the size of the resulting circuit. The net result is a quantum circuit capable of implementing any (real or complex) unitary matrix, specialized to use one of several types of two-qubit gates, appropriate for different physical implementations of quantum computing hardware.

Our procedure below relies upon the Generalized Singular Value Decomposition (GSVD) [204]. The GSVD recognizes that the SVDs of the four quadrants of an

orthogonal matrix are highly inter-related to one another. In particular, if we have a unitary matrix U , of dimension $2^n \times 2^n$, where n is the number of qubits, the GSVD yields

$$U = \begin{pmatrix} L_1 & 0 \\ 0 & L_2 \end{pmatrix} \cdot \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \cdot \begin{pmatrix} R_1 & 0 \\ 0 & R_2 \end{pmatrix}, \quad (3.86)$$

where the L_1 , L_2 , R_1 , and R_2 blocks are $2^{n-1} \times 2^{n-1}$ unitary matrices, and the matrix Σ is a tri-banded unitary matrix as with Σ_{ij} s are all diagonal matrices. The Σ matrix can be further decomposed into a product of two qubit-reversal operations and a block-diagonal unitary matrix with blocks representing one-qubit elementary gate operations:

$$\begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} = P_{2^n}^{-1} \cdot \begin{pmatrix} \Sigma'_{11} & 0 & \dots & 0 \\ 0 & \Sigma'_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Sigma'_{2^{n-1}2^{n-1}} \end{pmatrix} \cdot P_{2^n}, \quad (3.87)$$

where P_{2^n} is a qubit reversal matrix which is composed of cascaded SWAP gates, and Σ'_{11} , Σ'_{22} etc, are 2×2 unitary operations that can be expressed as R_y -rotations.

If $n > 2$, the decomposition can be iterated. The four unitary sub-blocks can be further decomposed until all resulting matrices are block-diagonal unitary matrices, with blocks representing 1-qubit elementary gates. For example, further decomposing L_1 and L_2 above

$$L_1 = \begin{pmatrix} L'_1 & 0 \\ 0 & L'_2 \end{pmatrix} \cdot P_{2^{n-1}}^{-1} \cdot \begin{pmatrix} \Sigma'_{11} & 0 \\ 0 & \Sigma'_{22} \end{pmatrix} \cdot P_{2^{n-1}} \cdot \begin{pmatrix} R'_1 & 0 \\ 0 & R'_2 \end{pmatrix}, \quad (3.88)$$

$$L_2 = \begin{pmatrix} L''_1 & 0 \\ 0 & L''_2 \end{pmatrix} \cdot P_{2^{n-1}}^{-1} \cdot \begin{pmatrix} \Sigma''_{11} & 0 \\ 0 & \Sigma''_{22} \end{pmatrix} \cdot P_{2^{n-1}} \cdot \begin{pmatrix} R''_1 & 0 \\ 0 & R''_2 \end{pmatrix}. \quad (3.89)$$

Rejoining L_1 and L_2 , we obtain

$$\begin{aligned} L &= \begin{pmatrix} L_1 & 0 \\ 0 & L_2 \end{pmatrix} \\ &= \begin{pmatrix} L'_1 & 0 & 0 \\ 0 & L'_2 & 0 \\ 0 & 0 & L''_1 & 0 \\ && 0 & L''_2 \end{pmatrix} \cdot (\mathbb{1} \otimes P_{2^n}^{-1}) \cdot \begin{pmatrix} \Sigma'_{11} & 0 & 0 & 0 \\ 0 & \Sigma'_{22} & 0 & 0 \\ 0 & 0 & \Sigma''_{11} & 0 \\ 0 & 0 & 0 & \Sigma''_{22} \end{pmatrix} \\ &\quad \cdot (\mathbb{1} \otimes P_{2^n}) \cdot \begin{pmatrix} R'_1 & 0 & 0 \\ 0 & R'_2 & 0 \\ 0 & 0 & R''_1 & 0 \\ && 0 & R''_2 \end{pmatrix}, \end{aligned} \quad (3.90)$$

where $\mathbb{1}$ is the 2×2 identity matrix. This process can be repeated until each matrix is block-diagonal, in which the blocks are 2×2 unitary matrices representing arbitrary 1-qubit gates. In turn, each of the 1-qubit gates can be decomposed into four

independent operations by application of the following lemma: Every 2×2 unitary matrix can be factored as a product of two R_z -rotations, one R_y -rotation and one phase shift [33]

$$\begin{aligned} & \begin{pmatrix} e^{i\delta} & 0 \\ 0 & e^{i\delta} \end{pmatrix} \cdot \begin{pmatrix} e^{i\alpha/2} & 0 \\ 0 & e^{-i\alpha/2} \end{pmatrix} \cdot \begin{pmatrix} \cos(\theta/2) & \sin(\theta/2) \\ -\sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \cdot \begin{pmatrix} e^{i\beta/2} & 0 \\ 0 & e^{-i\beta/2} \end{pmatrix} \\ & \equiv Ph(\delta) \cdot R_z(\alpha) \cdot R_y(\theta) \cdot R_z(\beta), \end{aligned} \quad (3.91)$$

where δ , α , θ , and β are real valued. If the unitary matrix has unit determinant, the phase gate can be dropped. Hence, any $2^n \times 2^n$ dimensional block-diagonal unitary matrix, whose blocks are 2×2 unitary matrices, can be decomposed into the product of (at most) four simpler $2^n \times 2^n$ dimensional unitary matrices corresponding to purely phase shifts, z -rotations, or y -rotations.

The next step is to map each of these (“purified”) block diagonal matrices into an equivalent quantum circuit fragment. The concatenation of all such fragments, interleaved with compactification rules, yields a complete quantum circuit for U . Different types of block diagonal matrices factorize into different circuit fragments. Consider a 4×4 block-diagonal unitary matrix, R , in which the blocks are y -rotations about different angles. As a matrix, R is expressed as

$$R = \begin{pmatrix} R_y(\theta_1) & 0 \\ 0 & R_y(\theta_2) \end{pmatrix},$$

Intuitively, we can create R from two simpler operators: one which applies the *same* angular rotation to both the upper left and lower right quadrants, and another which applies *opposing* angular rotations to the upper left and lower right quadrants. For appropriately chosen angles, the product of such operations can achieve any desired angular pair. Thus, we consider

$$\begin{aligned} \mathbb{1} \otimes R_y(\alpha) &= \begin{pmatrix} R_y(\alpha) & 0 \\ 0 & R_y(\alpha) \end{pmatrix}, \\ \text{CNOT} \cdot (\mathbb{1} \otimes R_y(\beta)) \cdot \text{CNOT} &= \begin{pmatrix} R_y(\beta) & 0 \\ 0 & R_y(-\beta) \end{pmatrix} \end{aligned}$$

We can achieve R provided $\alpha + \beta = \theta_1$ and $\alpha - \beta = \theta_2$. Hence, $\alpha = \frac{\theta_1 + \theta_2}{2}$, $\beta = \frac{\theta_1 - \theta_2}{2}$, and the quantum circuit diagram representing $R = R_y(\theta_1) \oplus R_y(\theta_2)$ is shown in Fig. 3.33.

Generalizing to the n -qubit case, we have a $2^n \times 2^n$ block-diagonal matrix whose blocks are one-qubit R_y -rotations through angles $\{\theta_1, \dots, \theta_{2^{n-1}}\}$. The quantum circuit for such a matrix can be generated recursively as

$$\begin{aligned} R_y^\oplus(n, A) &= \text{CNOT}_{1,n;n} \cdot (\mathbb{1} \otimes R_y^\oplus(n-1, A_{1 \rightarrow 2^{n-2}})) \\ &\quad \cdot \text{CNOT}_{1,n;n} \cdot (\mathbb{1} \otimes R_y^\oplus(n-1, A_{2^{n-2}+1 \rightarrow 2^{n-1}})), \end{aligned} \quad (3.92)$$

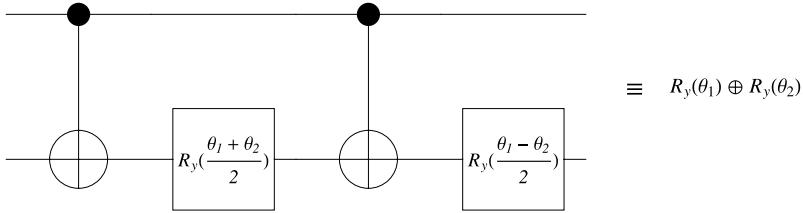


Fig. 3.33 Quantum circuit for a block-diagonal R_y operator

where A is a vector of angles given by

$$A = \begin{pmatrix} A_1 \\ \vdots \\ A_{2^{n-1}} \end{pmatrix} = W^{\otimes(n-1)} \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_{2^{n-1}} \end{pmatrix},$$

with the (intentionally non-unitary) matrix $W = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, and $\text{CNOT}_{1,n;n}$ is a CNOT gate between the first and n -th of n -qubits. The notation $A_{i \rightarrow j}$ means the vector of angles between indices i and j in A .

An identical construction applies to the case of the direct sum of many R_z rotations through different angles. Hence a $2^n \times 2^n$ block-diagonal matrix whose blocks are one-qubit R_z -rotations through different angles can be mapped into a quantum circuit generated as:

$$\begin{aligned} R_z^\oplus(n, A) = & \text{CNOT}_{1,n;n} \cdot (\mathbb{1} \otimes R_z^\oplus(n-1, A_{1 \rightarrow 2^{n-2}})) \\ & \cdot \text{CNOT}_{1,n;n} \cdot (\mathbb{1} \otimes R_z^\oplus(n-1, A_{2^{n-2}+1 \rightarrow 2^{n-1}})). \end{aligned} \quad (3.93)$$

For the 4×4 block-diagonal unitary matrix, Φ , in which the blocks are Ph -gates represented as

$$\Phi = \begin{pmatrix} Ph(\theta_1) & 0 \\ 0 & Ph(\theta_2) \end{pmatrix}. \quad (3.94)$$

The quantum circuit achieving Φ is $R_z(\theta_1 - \theta_2) \otimes Ph((\theta_1 + \theta_2)/2)$. It follows that the quantum circuit fragment for a $2^n \times 2^n$ block-diagonal matrix whose blocks are one-qubit Ph -gates can be defined recursively as:

$$Ph^\oplus(n, A) = U_3 \cdot U_2 \cdot U_1 \quad (3.95)$$

where

$$U_3 = \mathbb{1} \otimes Ph^\oplus(n-1, A_{1 \rightarrow 2^{n-2}}),$$

$$U_2 = R_z(A_{2^{n-2}+1}) \otimes \mathbb{1} \otimes \cdots \otimes \mathbb{1},$$

$$U_1 = \text{CNOT} \otimes R_z(A_{2^{n-2}+2 \rightarrow 2^{n-1}}) \otimes \text{CNOT}.$$

Hence, all three primitive types of $2^n \times 2^n$ dimensional block-diagonal matrices can be mapped into corresponding quantum circuit fragments, which use only CNOT gates, and one-qubit Ph -, R_y -, and R_z -operations.

3.7.5 Simplification via Rewrite Rules

To complete the decomposition we concatenate the circuit fragments, and apply final compactification in an attempt to minimize the overall circuit complexity. The compactification rules eliminate unnecessary gate operations including rotations and phase shifts through an angle of zero or $2n\pi$, combine contiguous sequences of Ph -, R_y -, or R_z -gate operations, accumulate phase gates into an overall phase, compress sequences of CNOT gates having the same embedding, and implement bit-reversals by explicitly rewiring the circuit (rather than implementing such operations computationally). These compactification rules are found to reduce the complexity of the final quantum circuit significantly. Specifically, whereas naive use of algebraic schemes would *always* result in exponentially large circuits, if augmented with rewrite rules, unitary operators having a direct product structure are mapped to compact quantum circuits, real unitary operators are mapped into smaller circuits than complex unitary operators, and known “special case” unitary operators (such as the QFT) are found to have smaller circuits than random unitary operators.

The idea is that we have found, by one means or another, a quantum circuit sufficient to realize a desired n -qubit quantum computation. We now wish to “compactify” the circuit so that the n -qubit operation can be accomplished using the fewest quantum gates. This can be accomplished by developing rewrite rules for quantum circuits.

Term rewriting is a general purpose technique used in automated theorem proving [91]. To be effective, a rewrite rule system must be “Canonical” and “Church-Rosser”. “Canonical” means that equivalent expressions are rewritten to a common form. “Church-Rosser” means that some measure of the structural complexity of the expression being rewritten is reduced after each rule invocation. We can guarantee that rewrite rules are Canonical and Church-Rosser by using a strict syntactic grammar for circuits, and ensuring that a rewrite is only applied if it reduces or leaves constant the gate count.

It makes sense, initially, to focus attention on rewrite rules for simplifying 2-qubit quantum circuits. The rationale for this is that all quantum circuits can be reduced to 1-qubit and 2-qubit quantum gates. However, it is very possible that higher levels of circuit abstraction, e.g., treating a QFT as a special group of gates, could facilitate recognition of more simplification opportunities. However, by grouping CNOT gates with the surrounding 1-qubit gates that share the same embedding, and then compactifying them, we systematically reduce the complexity of the overall n -qubit operation. The number of possibilities are enormous. Here is an example of that gives you the flavor of what is involved.

There are several ways to find rewrite rules. However, one must exercise good judgment in rule selection as there are infinitely many potential rules. We need to

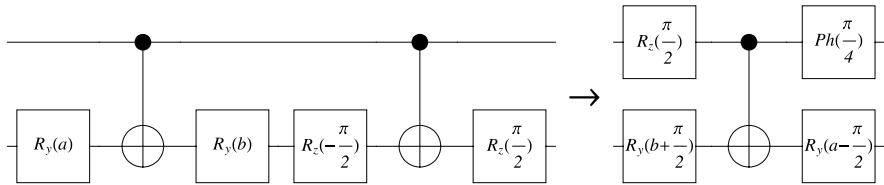


Fig. 3.34 Example of a rewrite rule that eliminates a CNOT gate. The structure shown in this circuit fragment happens to arise very often when using the algebraic circuit design method using a gate family consisting of CNOT, $R_z(\cdot)$, $R_y(\cdot)$, and $Ph(\cdot)$ gates

find those rules that (a) allow us to recognize special structure (should it be present) and (b) tend to arise in practice during the operation of the algebraic decomposition procedure. Hence, permutation matrices, and other sparse unitary matrices having a single non-zero element in each row and column, are especially good sources of inspiration for discovering rewrite rules. Being sparse, they require fewer gates than a general purpose 2-qubit unitary, and often give rise to gates involving special angles, often $\pi/2^k$.

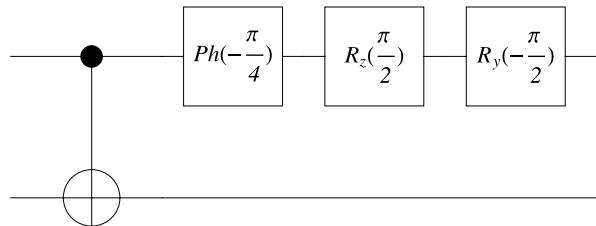
The following example is typical of the kinds of rewrite rules one can find. The circuit fragment on the left hand side of Fig. 3.34 arises commonly in the algebraic decomposition method that uses the GSVD, which we described above. This circuit fragment contains two CNOT gates. However, it can be rewritten into the form shown on the right hand side of Fig. 3.34, which contains only one CNOT gate. Thus, the application of the rewrite rule of Fig. 3.34 eliminates a 2-qubit gate at the expense of inserting extra 1-qubit gates. This is preferable, however, because any contiguous sequence of 1-qubit gates, no matter how long, that are all acting on the same qubit, can be compressed into a sequence of at most four 1-qubit gates. This is because the net effect of any such contiguous sequence of gates is still just some 1-qubit gate. However, we know that any 1-qubit gate can be factored into the form $Ph(\delta) \cdot R_z(\gamma) \cdot R_y(\beta) \cdot R_z(\alpha)$.

Example: Finding a Quantum Circuit for an Arbitrary Unitary Matrix As an example, the unitary operator generated by the Hamiltonian $\sigma_y \otimes \sigma_x$ in Mermin's version of the Bell-Kochen-Specker theorem [415] is

$$\frac{1}{\sqrt{2}} \begin{pmatrix} -i & 0 & 0 & 1 \\ 0 & -i & 1 & 0 \\ i & 0 & 0 & 1 \\ 0 & i & 1 & 0 \end{pmatrix}.$$

Figure 3.35 shows a quantum circuit sufficient to implement this operator.

Fig. 3.35 Quantum circuit for the unitary operator in Mermin’s version of the Bell-Koche-Specker theorem [415]



3.7.6 Numerical Method

Another way to find a quantum circuit that implements a given unitary matrix is via an exploration of the space of possible circuits templates in conjunction with numerical optimization. This method is only feasible for few qubit circuits due to the combinatorial explosion in the number of possible circuit templates with increasing numbers of qubits.

The motivation behind a numerical approach is as follows. Given a unitary matrix, U_{target} , describing some desired quantum computation, one is typically most interested in finding the *smallest* quantum circuit, with respect to a particular universal gate set, sufficient to implement it? For 2-qubit quantum computations, and the gate set consisting of CNOT and all 1-qubit gates, a solution is already known using the circuit templates outlined in Chap. 2. Conversely, for quantum computations on large numbers of qubits, the problem appears utterly intractable at present. In this case the best we can do is to synthesize a (likely) sub-optimal quantum circuit using algebraic or genetic methods, and then apply rewrite rules recursively to simplify and compress the circuit until no more rules fire. However, such an approach is neither guaranteed, nor likely, to find an the smallest quantum circuit for a given target unitary matrix.

In the regime between “two” qubits and “many” qubits, exhaustive enumeration of all possible circuit templates followed by numerical optimization can work surprisingly well, even on modest modern computers. The idea is to exhaustively generate all possible circuit topologies of increasing complexity, and to use numerical methods to find the values for the free parameters in those circuit topologies that minimize the *discrepancy*, $\text{discrepancy}(U_{\text{target}}, U_{\text{template}})$, between the target unitary matrix, U_{target} and the actual unitary matrix such a circuit template achieves, U_{template} . A simple measure of discrepancy (but by no means the only one) is the absolute value of the difference between respective elements of the matrices U_{target} and U_{template} , i.e.,

$$\text{discrepancy}(U_{\text{target}}, U_{\text{template}}) = \frac{1}{N^2} \sum_{j=1}^N \sum_{k=1}^N |U_{jk} - V_{jk}| \quad (3.96)$$

By finding values for the parameters within a particular circuit template that make the discrepancy between the target unitary and one implied by the template go to zero, an *exact* quantum circuit for U_{target} can be found. Moreover, if the different

circuit templates are generated in the order of increasing circuit size, the *first* circuit template found that achieves U_{target} will indeed be the smallest circuit for achieving the target with respect to the chosen family of gates. A nice feature of this approach is that we do not have to limit ourselves to a minimal set of universal gates. We can, if we choose, use an over-complete set of universal gates. If we use an under-complete set of gates there is no guarantee a solution can be found. Nevertheless, a numerical search is sometimes worthwhile, e.g., if one is limited in the types of gates one can achieve physically in a particular embodiment of a quantum computer.

Example: Numerical Design of a Circuit for a 1-Qubit Gate Suppose the target unitary matrix U_{target} be the 1-qubit Walsh-Hadamard gate:

$$U_{\text{target}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (3.97)$$

and we wish to find how to achieve a Walsh-Hadamard gate in terms of 1-qubit rotations about the z -axis and y -axis, and a single phase gate. We know a solution in terms of such gates is always possible because we learned in Chap. 2 that any 1-qubit gate can be expressed in terms of the circuit template:

$$U_{\text{target}} = Ph(\delta) \cdot R_z(\gamma) \cdot R_y(\beta) \cdot R_z(\alpha) \quad (3.98)$$

where α, β, γ and δ are angles. Our job is to find values for these angles that achieves the Walsh-Hadamard gate.

We can solve this problem using computer algebra tools such as *Mathematica*—a superb software package for doing all things mathematical on a computer. Specifically, we can use Mathematica’s “NMinimize” function to find values of the angles α, β, γ , and δ that minimize discrepancy as follows:

$$\begin{aligned} U_{\text{target}} &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}; \\ U_{\text{template}} &= Ph(\delta) \cdot R_z(\gamma) \cdot R_y(\beta) \cdot R_z(\alpha) \\ &= \begin{pmatrix} e^{-\frac{i\alpha}{2} + i\delta - \frac{i\gamma}{2}} \cos\left(\frac{\beta}{2}\right) & -e^{\frac{i\alpha}{2} + i\delta - \frac{i\gamma}{2}} \sin\left(\frac{\beta}{2}\right) \\ e^{-\frac{i\alpha}{2} + \frac{i\gamma}{2} + i\delta} \sin\left(\frac{\beta}{2}\right) & e^{\frac{i\alpha}{2} + \frac{i\gamma}{2} + i\delta} \cos\left(\frac{\beta}{2}\right) \end{pmatrix}; \end{aligned} \quad (3.99)$$

answer = NMinimize [discrepancy [$U_{\text{target}}, U_{\text{template}}$], $\{\alpha, \beta, \gamma, \delta\}]$

$$\rightarrow \left\{ 0, \left\{ \alpha \rightarrow \pi, \beta \rightarrow \frac{\pi}{2}, \gamma \rightarrow 0, \delta \rightarrow \frac{\pi}{2} \right\} \right\} \quad (3.100)$$

where we replaced approximate numerical values in the answer with rational multiples of π or zero and then checked the result. Thus, we find that numerical minimization of the discrepancy reveals that $U_{\text{target}} = Ph(\pi/2) \cdot R_z(0) \cdot R_y(\pi/2) \cdot R_z(\pi) = Ph(\pi/2) \cdot R_y(\pi/2) \cdot R_z(\pi)$.

Example: Numerical Design of a Circuit for a 3-Qubit Gate Numerical discrepancy minimization can be used to find circuits for multi-qubit gates too. Unfortunately, the number of potential quantum circuit templates grows exponentially with the depth of the circuit. So brute force numerical minimization of the discrepancy between a target unitary matrix and that induced by a particular circuit template is only feasible for multi-qubit circuits that are not too deep. Moreover, any quantum circuit needs at most one phase gate $Ph(\delta)$. All the other 1-qubit gates can be factored in the form $R_z(\gamma) \cdot R_y(\beta) \cdot R_z(\alpha)$. Therefore, when enumerating circuit templates we should allow for the possibility that each 1-qubit gate may require up to three free parameters to specify it completely. Hence, the number of parameters over which one is optimizing can grow rapidly with the number of 1-qubit gates. Luckily, the unitary matrices that usually arise in purposeful quantum computations tend to be sparse and hence realizable in quantum circuits that are neither too deep (in step count) nor too large (in gate count).

As an example, consider the unitary matrix U_{target} defined as follows:

$$U_{\text{target}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.101)$$

This is an 8×8 unitary matrix corresponding to a circuit having three qubits. However, whereas random 8×8 unitary matrices would have a non-zero complex value for every element, this unitary matrix is relatively sparse, has only real elements, and is somewhat symmetric. These are all clues that the matrix may admit an anomalously compact quantum circuit. Our goal is to find the smallest quantum circuit template, which when suitably optimized, will provide an exact circuit for implementing U_{target} .

To apply the numerical design method, we need to pick an ordering in which to enumerate quantum circuit templates that can be built from a given universal set of gates, such as the set of 1-qubit gates and CNOT. One such ordering is given by allowing each “step” of the circuit to be taken up by either a single general-up-to-phase-factor 1-qubit gate ($R_z(\gamma) \cdot R_y(\beta) \cdot R_z(\alpha)$) or a single CNOT gate between the i -th and j -th of k qubits ($\text{CNOT}_{i,j;k}$). In an n -qubit circuit, there are therefore n ways to embed the 1-qubit gate plus $2\binom{n}{2} = n(n-1)$ ways to embed the CNOT gate. Hence, using this enumeration scheme, which is certainly not the only possibility, each step in the circuit can be taken up by one of n^2 distinct templates. Hence, the number of possible templates that need to be tested in a circuit k steps deep will be approximately n^{2k} . In the present (3-qubit) example the number of templates to test

template($(U_1 \otimes U_2 \otimes U_3)$, CNOT _{1,2;3})	template($(U_1 \otimes U_2 \otimes U_3)$, CNOT _{2,1;3})
template($(U_1 \otimes U_2 \otimes U_3)$, CNOT _{1,3;3})	template($(U_1 \otimes U_2 \otimes U_3)$, CNOT _{3,1;3})
template($(U_1 \otimes U_2 \otimes U_3)$, CNOT _{2,3;3})	template($(U_1 \otimes U_2 \otimes U_3)$, CNOT _{3,2;3})
template(CNOT _{1,2;3} , $(U_1 \otimes U_2 \otimes U_3)$)	template(CNOT _{1,2;3} , CNOT _{2,1;3})
template(CNOT _{1,2;3} , CNOT _{1,3;3})	template(CNOT _{1,2;3} , CNOT _{3,1;3})
template(CNOT _{1,2;3} , CNOT _{2,3;3})	template(CNOT _{1,2;3} , CNOT _{3,2;3})
template(CNOT _{2,1;3} , $(U_1 \otimes U_2 \otimes U_3)$)	template(CNOT _{2,1;3} , CNOT _{1,2;3})
template(CNOT _{2,1;3} , CNOT _{1,3;3})	template(CNOT _{2,1;3} , CNOT _{3,1;3})
template(CNOT _{2,1;3} , CNOT _{2,3;3})	template(CNOT _{2,1;3} , CNOT _{3,2;3})
template(CNOT _{1,3;3} , $(U_1 \otimes U_2 \otimes U_3)$)	template(CNOT _{1,3;3} , CNOT _{1,2;3})
template(CNOT _{1,3;3} , CNOT _{2,1;3})	template(CNOT _{1,3;3} , CNOT _{3,1;3})
template(CNOT _{1,3;3} , CNOT _{2,3;3})	template(CNOT _{1,3;3} , CNOT _{3,2;3})
template(CNOT _{3,1;3} , $(U_1 \otimes U_2 \otimes U_3)$)	template(CNOT _{3,1;3} , CNOT _{1,2;3})
template(CNOT _{3,1;3} , CNOT _{2,1;3})	template(CNOT _{3,1;3} , CNOT _{1,3;3})
template(CNOT _{3,1;3} , CNOT _{2,3;3})	template(CNOT _{3,1;3} , CNOT _{3,2;3})
template(CNOT _{2,3;3} , $(U_1 \otimes U_2 \otimes U_3)$)	template(CNOT _{2,3;3} , CNOT _{1,2;3})
template(CNOT _{2,3;3} , CNOT _{2,1;3})	template(CNOT _{2,3;3} , CNOT _{1,3;3})
template(CNOT _{2,3;3} , CNOT _{3,1;3})	template(CNOT _{2,3;3} , CNOT _{3,2;3})
template(CNOT _{3,2;3} , $(U_1 \otimes U_2 \otimes U_3)$)	template(CNOT _{3,2;3} , CNOT _{1,2;3})
template(CNOT _{3,2;3} , CNOT _{2,1;3})	template(CNOT _{3,2;3} , CNOT _{1,3;3})
template(CNOT _{3,2;3} , CNOT _{3,1;3})	template(CNOT _{3,2;3} , CNOT _{2,3;3})

Fig. 3.36 Enumeration of quantum circuit templates for a 3-qubit circuit of depth two using a universal set consisting of all 1-qubit gates and CNOT. Note that, at a given step, a single qubit gate can be inserted in one of three ways, and a single CNOT gate can be inserted in one of six ways. Thus, a quantum circuit template of depth two has $9^2 = 81$ possible forms. Of these we exclude those that involve a sequence of same type of gate with the same embedding. This brings the number of templates down to 72

in a depth k circuit is approximately 3^{2k} , which grows exponentially in k , the depth of the circuit. As an explicit example, Fig. 3.36 shows the templates that would be tested in an attempt to find a decomposition of a 3-qubit unitary matrix into a depth-2 quantum circuit.

The number of quantum circuit templates to test can be reduced by excluding redundant circuit topologies (e.g., $\text{CNOT}_{i,j;k} \cdot \text{CNOT}_{i,j;k} = \mathbb{1}$, merging abutting 1-qubit gates (e.g., $R_z(\alpha) \cdot R_z(\alpha) = R_z(2\alpha)$), recognizing circuits that achieve U^\dagger (in which case you reverse the ordering of the gates) and recognizing those that achieve $P_{2^n} \cdot P_{2^n}$ (in which case you reverse the order of the qubits). Nevertheless, such

numerical methods are still demanding computationally.

$$U_{\text{target}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.102)$$

$$U_{\text{template}} = \text{CNOT}_{3,1;3} \cdot \text{CNOT}_{2,1;3} \cdot (R_x(a) \otimes R_y(b) \otimes R_z(c))$$

$$\cdot \text{CNOT}_{2,3;3} \cdot \text{CNOT}_{1,3;3};$$

$$\text{answer} = \text{NMinimize} [\text{discrepancy}[U_{\text{target}}, U_{\text{template}}], \{\alpha, \beta, \gamma\}]$$

$$\rightarrow \{0, \{\alpha \rightarrow \pi, \beta \rightarrow \frac{\pi}{2}, \gamma \rightarrow \pi\}\}$$

where numerical optimization of this last template yields an exact solution with $a \rightarrow \pi$, $b \rightarrow \pi/2$, and $c \rightarrow \pi$.

3.7.7 Re-use Method

“If I have seen further it is by standing on the shoulders of Giants”

– Isaac Newton

So far we have seen how to synthesize quantum circuits algebraically, using matrix decompositions, and numerically, using discrepancy minimization. Although both these approaches have their uses, neither builds upon any insights we may have gleaned about efficient quantum circuits for other unitary transforms. An alternative approach to quantum circuit design, pioneered by computer scientists extraordinaire Andreas Klappenecker and Martin Röettler, is to design a quantum circuit for a desired unitary transform by understanding how that unitary transform is related *functionally* to another unitary transform for which an efficient quantum circuit is already known. In particular, in the “design-by-re-use” method a *known* quantum circuit for an operator U , and a *known* (typically polynomial) functional relationship $V = f(U) = \sum_i \alpha_i U^i$ is used to predict a new *efficient* quantum circuit for V [289, 290].

Moreover, surprisingly, the structural form of the new (efficient) quantum circuit for $V = f(U)$ is essentially fixed: the same basic circuit structure works regardless of the function f . All that needs to be changed is the form of a central gate whose

elements depend upon the coefficients α_i in the expansion $V = f(U) = \sum_i \alpha_i U^i$. Therefore, when applicable, the “design-by-re-use” method is better than generic algebraic circuit design for maximally general unitaries, which always yields an exponentially large circuit unless compactification rules are applied, and it is better than numerical circuit design, which quickly becomes intractable due to the combinatorial explosion in the number of potential circuit templates . Moreover, the design by re-use method operates at a higher level of abstraction than the 1-qubit and 2-qubit gate level, allowing a more meaningful interpretation of what the circuit is doing. Frankly, it is a beautiful and insightful approach to quantum circuit design and is deserving of far greater attention.

3.7.7.1 Functions of Matrices

The foundation of the “design-by-re-use” strategy is the idea of working with functions of *matrices*. We represent one unitary matrix, V , as a function of some other unitary matrix, U , i.e., $V = f(U)$. The reason this is possible is that if U is unitary, then it is guaranteed to be unitarily equivalent to a diagonal matrix, i.e.,

$$U = T \cdot \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N) \cdot T^\dagger \quad (3.103)$$

where $\{\lambda_i\}$ are the eigenvalues of U , and T is some (so-called “diagonalizing”) unitary matrix. Similarly, powers of U are diagonalized by the same matrix T :

$$U^i = T \cdot \text{diag}(\lambda_1^i, \lambda_2^i, \dots, \lambda_N^i) \cdot T^\dagger \quad (3.104)$$

Thus, if we have some function defined by its Taylor series:

$$f(x) = \sum_{i=0}^{\infty} \alpha_i x^i \quad (3.105)$$

we can see that the corresponding *function* of U , i.e., $V = f(U)$, can be written as:

$$V = f(U) = T \cdot \text{diag}(f(\lambda_1), f(\lambda_2), \dots, f(\lambda_N)) \cdot T^\dagger \quad (3.106)$$

We can then rewrite this formula to confirm that $f(U)$ can also be written as a linear combination of powers of U . Specifically, we have:

$$\begin{aligned} f(U) &= T \cdot \text{diag}(f(\lambda_1), f(\lambda_2), \dots, f(\lambda_N)) \cdot T^\dagger \\ &= T \cdot \text{diag}\left(\sum_{i=0}^{\infty} \alpha_i \lambda_1^i, \sum_{i=0}^{\infty} \alpha_i \lambda_2^i, \dots, \sum_{i=0}^{\infty} \alpha_i \lambda_N^i\right) \cdot T^\dagger \\ &= T \cdot \left[\sum_{i=0}^{\infty} \text{diag}(\alpha_i \lambda_1^i, \alpha_i \lambda_2^i, \dots, \alpha_i \lambda_N^i) \right] \cdot T^\dagger \end{aligned}$$

$$\begin{aligned}
&= T \cdot \left[\sum_{i=0}^{\infty} \alpha_i \text{diag}(\lambda_1^i, \lambda_2^i, \dots, \lambda_N^i) \right] \cdot T^\dagger \\
&= \sum_{i=0}^{\infty} \alpha_i T \cdot D^i \cdot T^\dagger = \sum_{i=0}^{\infty} \alpha_i U^i
\end{aligned} \tag{3.107}$$

which implies we can write a function of the matrix U as a linear combination of integer powers of U , i.e.,

$$V = f(U) = \sum_{i=0}^{\infty} \alpha_i U^i \tag{3.108}$$

where the α_i are, in general, complex numbers. In many cases of interest, the sum need not run to infinity to obtain an exact equivalence.

3.7.7.2 Quantum Hartley Transform as a Polynomial in QFT

For example, the discrete Hartley transform can be expressed as a polynomial in the discrete Fourier transform. Analogizing to the quantum case, the quantum Hartley transform will be given by:

$$\text{QHT}_N = \alpha \text{ QFT}_N + \beta \text{ QFT}_N^3 = \text{QFT}_N \cdot (\alpha \mathbb{1}_N + \beta \text{ QFT}_N^2) \tag{3.109}$$

where $\alpha = (\frac{1-i}{2})$ and $\beta = (\frac{1+i}{2})$. As an efficient quantum circuit for the QFT is known, if we can find an efficient quantum circuit for $(\alpha \mathbb{1}_N + \beta \text{ QFT}_N^2)$ we will find an efficient quantum circuit for the quantum Hartley transform (QHT).

3.7.7.3 Quantum Fractional Fourier Transform as a Polynomial in QFT

Similarly, another useful transform, the fractional Fourier transform, can also be expressed as a low order polynomial in the Fourier transform. Specifically, in the quantum case we have:

$$\begin{aligned}
\text{QFFT}_{N;\alpha} &= \text{QFT}_N^{2\alpha/\pi} \\
&= a_0(\alpha) \text{ QFT}_N^0 + a_1(\alpha) \text{ QFT}_N^1 \\
&\quad + a_2(\alpha) \text{ QFT}_N^2 + a_3(\alpha) \text{ QFT}_N^3
\end{aligned} \tag{3.110}$$

where

$$\begin{aligned} a_0(\alpha) &= \frac{1}{2}(1 + e^{i\alpha}) \cos(\alpha) \\ a_1(\alpha) &= \frac{1}{2}(1 - ie^{i\alpha}) \sin(\alpha) \\ a_2(\alpha) &= \frac{1}{2}(-1 + e^{i\alpha}) \cos(\alpha) \\ a_3(\alpha) &= \frac{1}{2}(-1 - ie^{i\alpha}) \sin(\alpha) \end{aligned} \quad (3.111)$$

Like the Fourier transform, the fractional Fourier transform is a time-frequency transform, but by involving the parameter α it can transform a signal to a domain that is *intermediate* between time and frequency. Clearly:

- when $\alpha = 0$ the QFFT collapses to the identity, i.e., $\text{QFFT}_{N;0} = \mathbb{1}_N$;
- when $\alpha = \frac{\pi}{2}$ the QFFT collapses to the QFT, i.e., $\text{QFFT}_{N;\frac{\pi}{2}} = \text{QFT}_N$;
- the indices of two QFTs add, i.e., $\text{QFFT}_{N;\alpha} \cdot \text{QFFT}_{N;\beta} = \text{QFFT}_{N;\alpha+\beta}$.

The design by re-use method exploits the ability to express the QFFT as a polynomial in the QFT to find an efficient quantum circuit for QFFT.

3.7.7.4 Fixed Structure of the “Design by Re-use” Circuit

It turns out that any unitary matrix V that can be written as a linear combination of integer powers of a unitary matrix U for which efficient quantum circuits are known, also admits an efficient quantum circuit decomposition. Furthermore, the *structure* of a circuit for $V = \sum_i \alpha_i U^i$ is essentially the same in all cases and is shown in Fig. 3.37. In every case the gate C corresponds to a unitary circulant matrix whose elements are related to the particular coefficients α_i in the series expansion of $V = f(U) = \sum_i \alpha_i U^i$.

A circulant matrix is matrix in which the elements in each row are rotated one element to the right relative to the preceding row. Therefore circulant matrices have the structure:

$$C = \begin{pmatrix} c_0 & c_3 & c_2 & c_1 \\ c_1 & c_0 & c_3 & c_2 \\ c_2 & c_1 & c_0 & c_3 \\ c_3 & c_2 & c_1 & c_0 \end{pmatrix} \quad (3.112)$$

A key property of circulant matrices is that they are diagonalized by the QFT matrix. That is, $\text{QFT}^\dagger \cdot C \cdot \text{QFT}$ is always a diagonal matrix.

The generic circuit structure shown in Fig. 3.37 can be “programmed” to implement a particular $V = \sum_i \alpha_i U^i$ by changing the unitary circulant matrix, C , used in the center of the circuit. Below we give examples of how to “program” this generic circuit to yield efficient quantum circuits for the quantum Hartley transform (QHT) and quantum fractional Fourier transform (QFFT).

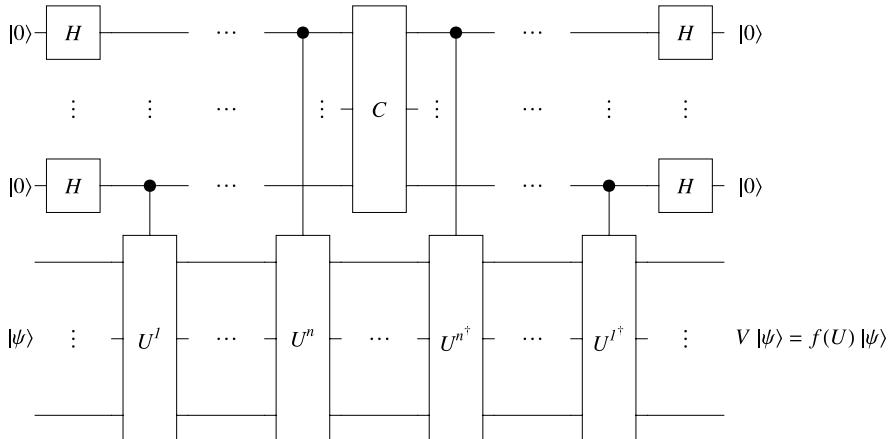


Fig. 3.37 Given efficient quantum circuits for U^i , an efficient quantum circuit for $V = f(U) = \sum_i \alpha_i U^i$ has the form shown. Here C is a unitary circulant matrix whose elements are related to the coefficients α_i

3.7.7.5 Quantum Circuit for QHT via “Design-by-Re-use”

In the case of the quantum Hartley transform (QHT) we have

$$\text{QHT} = \text{QFT}_N \cdot (\alpha \mathbb{1}_N + \beta \text{QFT}_N^2) \quad (3.113)$$

where $\alpha = (\frac{1-i}{2})$ and $\beta = (\frac{1+i}{2})$. The leading QFT is easy so we only need to focus on finding an efficient quantum circuit for the matrix $(\alpha \mathbb{1}_N + \beta \text{QFT}_N^2)$. The circulant matrix in this case is:

$$C = \frac{1}{2} \begin{pmatrix} 1-i & 1+i \\ 1+i & 1-i \end{pmatrix} \quad (3.114)$$

and therefore using the “design-by-re-use” template circuit of Fig. 3.37 an efficient quantum circuit for the QHT will have the form shown in Fig. 3.38.

3.7.7.6 Quantum Circuit for QFFT via “Design-by-Re-Use”

In the case of the quantum fractional Fourier transform (QFFT) we have

$$\text{QFFT}_{N;\alpha} = \text{QFT}_N^{2\alpha/\pi} = a_0(\alpha) \text{QFT}_N^0 + a_1(\alpha) \text{QFT}_N^1 + a_2(\alpha) \text{QFT}_N^2 + a_3(\alpha) \text{QFT}_N^3 \quad (3.115)$$

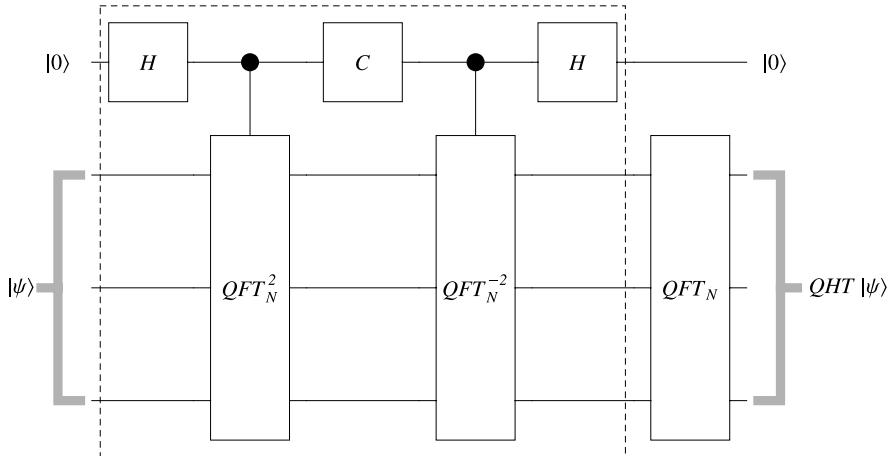


Fig. 3.38 Given an efficient quantum circuit for QFT, an efficient quantum circuit for $\text{QHT} = \text{QFT}_N \cdot (\alpha \mathbb{1}_N + \beta \text{QFT}_N^2)$ has the form shown. Here the circulant matrix is $C = \frac{1}{2} \begin{pmatrix} 1-i & 1+i \\ 1+i & 1-i \end{pmatrix}$

where

$$a_0(\alpha) = \frac{1}{2}(1 + e^{i\alpha}) \cos(\alpha)$$

$$a_1(\alpha) = \frac{1}{2}(1 - ie^{i\alpha}) \sin(\alpha)$$

$$a_2(\alpha) = \frac{1}{2}(-1 + e^{i\alpha}) \cos(\alpha)$$

$$a_3(\alpha) = \frac{1}{2}(-1 - ie^{i\alpha}) \sin(\alpha)$$

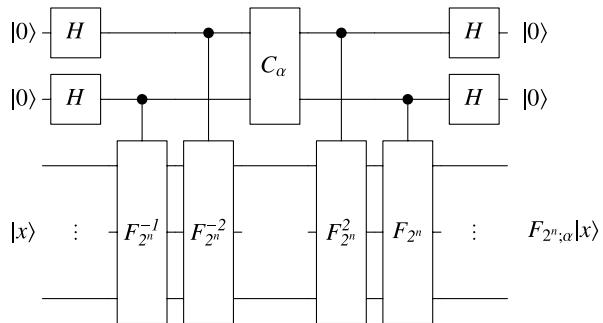
The QFT_N generates a finite group of order four with $\text{QFT}_N^4 = \mathbb{1}_N$. The circulant matrix in this case is:

$$C_\alpha = \text{QFT}_4^{-1} \cdot \text{diag}(1, e^{-i\alpha}, e^{2i\alpha}, e^{-i\alpha}) \cdot \text{QFT}_4 = \begin{pmatrix} a_0(\alpha) & a_3(\alpha) & a_2(\alpha) & a_1(\alpha) \\ a_1(\alpha) & a_0(\alpha) & a_3(\alpha) & a_2(\alpha) \\ a_2(\alpha) & a_1(\alpha) & a_0(\alpha) & a_3(\alpha) \\ a_3(\alpha) & a_2(\alpha) & a_1(\alpha) & a_0(\alpha) \end{pmatrix}$$

Using the template given in Fig. 3.37 an efficient quantum circuit for the QFFT therefore has the form shown in Fig. 3.39.

In truth, Klappenecker and Röettler technique is more general than our description of it here and can be re-cast in sophisticated group-theoretic terms. The more general way to look at the design-by-re-use method is that provided the unitary matrices, U^i , possess a finite dimensional group algebra one can always find an efficient quantum circuit for $V = \sum_i \alpha_i U^i$ having the fixed structure shown in Fig. 3.37, which can be “programmed” to obtain any unitary matrix contained in the

Fig. 3.39 In the figure, F_N represents a quantum Fourier transform QFT_N and $F_{N;\alpha}$ represents a quantum fractional Fourier transform, $\text{QFFT}_{N;\alpha}$ with $N = 2^n$



group algebra. To learn more, the interested reader should consult references [290] and [289].

3.8 Probabilistic Non-unitary Quantum Circuits

So far we have described quantum computations as the application of a sequence of deterministic quantum gates to an input quantum state followed by some non-deterministic measurement. In this picture, we generally view measurements as a necessary evil—the price we must pay to extract an answer from our quantum computation. However, we can use measurements in a more constructive manner, to apply a desired non-unitary transformation to a subset of qubits in a larger quantum system. However, the inherent randomness of the measurement process means that we will lose determinism. That is, we will not be able to achieve desired non-unitary transformations deterministically. However, it is possibly to trade success probability for damage in the sense that we can conceive of scheme for achieving non-unitary transformations of a state probabilistically such that the more likely we are to achieve the desired transform the more damage we do if we don't. The following example, based on a pair of papers by Bob Gingrich and I [199, 535], serves to illustrate this principle, but there are other constructions that could be used e.g., [275, 463, 488].

Suppose we want to construct a quantum circuit that performs the non-unitary transformation:

$$\rho_{\text{in}} \longrightarrow \frac{\mathcal{L} \cdot \rho_{\text{in}} \cdot \mathcal{L}^\dagger}{\text{tr}(\mathcal{L} \cdot \rho_{\text{in}} \cdot \mathcal{L}^\dagger)} \quad (3.116)$$

where \mathcal{L} is an $M \times N$ dimensional non-unitary matrix, and ρ_{in} is an arbitrary n -qubit density operator. The trace in the denominator guarantees that the output will be properly normalized. To ensure the transformation is well-defined, we also require $\det(\mathcal{L}) \neq 0$. If this condition is not met, we must explicitly exclude input states, ρ_{in} , such that $\mathcal{L} \cdot \rho_{\text{in}} \cdot \mathcal{L}^\dagger$ is the zero matrix.

Without loss of generality, we may assume the non-unitary matrix \mathcal{L} is of dimension $2^n \times 2^n$ such that $\max(M, N) \leq 2^n$. If, initially, \mathcal{L} has fewer than 2^n rows or

columns, we must pad \mathcal{L} with zeroes to the right of the columns, and/or below the rows, sufficient to make \mathcal{L} a $2^n \times 2^n$ dimensional matrix.

Given such padding, \mathcal{L} now has the right shape to be an n -qubit quantum gate. Unfortunately, it is still not unitary, and so cannot serve as a quantum gate directly. We need, therefore, to find a larger $(n+1)$ -qubit unitary matrix that contains \mathcal{L} within it in some computationally useful way. One route to creating such a enveloping unitary is to begin by first creating a specially crafted Hamiltonian.

3.8.1 Hamiltonian Built from Non-unitary Operator

Let us define a Hamiltonian to be of the form:

$$\mathcal{H} = -\epsilon \begin{pmatrix} 0 & -i\mathcal{L} \\ i\mathcal{L}^\dagger & 0 \end{pmatrix} \quad (3.117)$$

Such an \mathcal{H} is an hermitian matrix that contains the non-unitary matrix in anti-diagonal block form. Here ϵ is a constant that may be chosen freely. The value of ϵ will affect the fidelity with which we will be able to achieve our target non-unitary transformation and also the probability with which it can be made to occur.

3.8.2 Unitary Embedding of the Non-unitary Operator

Given such a Hamiltonian, we next determine what unitary evolution it implies. An $(n+1)$ -qubit quantum system with a Hamiltonian \mathcal{H} as defined above can achieve the following unitary gate:

$$\mathcal{Q} = \exp(-i\mathcal{H}) = \exp\left(i\epsilon \begin{pmatrix} 0 & -i\mathcal{L} \\ i\mathcal{L}^\dagger & 0 \end{pmatrix}\right) \quad (3.118)$$

3.8.3 Non-unitarily Transformed Density Matrix

If we augment the input state ρ_{in} with a single ancilla prepared initially in state $|1\rangle\langle 1|$, and evolve the expanded system under the action of \mathcal{Q} we can predict, as illustrated in Fig. 3.40, the final density matrix we will obtain, namely:

$$\rho_{out} = \mathcal{Q} \cdot (|1\rangle\langle 1| \otimes \rho_{in}) \cdot \mathcal{Q}^\dagger \quad (3.119)$$

If we then measure the ancilla in the computational basis, we will obtain either $|0\rangle$ or $|1\rangle$, and a certain transformation will be applied to the unmeasured qubits. But what

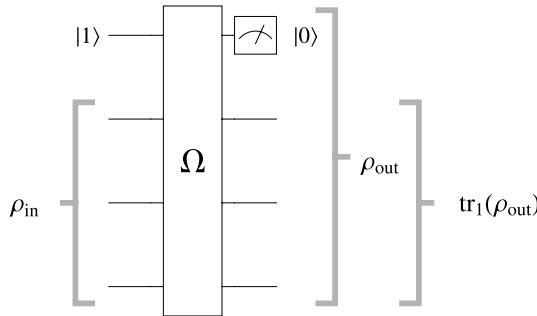


Fig. 3.40 Quantum circuit for achieving a non-unitary transformation probabilistically. Under the construction given in the text, when the output state of the ancilla is found to be $|0\rangle$ the reduced density matrix of the remaining unmeasured qubits contains a good approximation to the desired non-unitary transform, \mathcal{L} , of the input state ρ_{in} . The unitary operator Ω is defined via a designer Hamiltonian that contains the non-unitary operator \mathcal{L} in block anti-diagonal form

exactly will these transformations be? To answer this, we re-write the non-unitary operator \mathcal{L} in terms of its singular value decomposition:

$$\begin{aligned} \mathcal{L} &= U^\dagger \cdot \Sigma \cdot V \\ &= \begin{pmatrix} U^\dagger \cdot \cos(\epsilon \Sigma) \cdot U & 0 \\ 0 & V^\dagger \cdot \cos(\epsilon \Sigma) \cdot V \end{pmatrix} \\ &\quad + \begin{pmatrix} 0 & U^\dagger \cdot \sin(\epsilon \Sigma) \cdot V \\ -V^\dagger \cdot \sin(\epsilon \Sigma) \cdot U & 0 \end{pmatrix} \end{aligned} \quad (3.120)$$

In this form, we can read off what transformations are effected when we measure the ancilla and find it to be in either state $|0\rangle$ or $|1\rangle$. If the ancilla qubit in the output state, ρ_{out} , is measured and found to be in state $|0\rangle$, this constitutes the “success” scenario, and the transformation effected on the remaining n unmeasured quits is approximately

$$\mathcal{L}_{\text{succ}}^{\text{eff}} = U^\dagger \cdot \sin(\epsilon \Sigma) \cdot V \quad (3.121)$$

If we pick ϵ small and as Σ is a diagonal matrix, $\sin(\epsilon \Sigma) \approx \epsilon \Sigma$, and so the effective transformation on the remaining n unmeasured quits is approximately $\epsilon U^\dagger \cdot \Sigma \cdot V$, which is close to \mathcal{L} .

Conversely, if the ancilla qubit in the output state, ρ_{out} , is measured and found to be in state $|1\rangle$, this constitutes the “failure” scenario, and the transformation effected on the remaining n unmeasured quits is approximately

$$\mathcal{L}_{\text{fail}}^{\text{eff}} = V^\dagger \cdot \cos(\epsilon \Sigma) \cdot V \quad (3.122)$$

As ϵ is small and Σ is a diagonal matrix, $\cos(\epsilon \Sigma)$ is close to the identity operator, and so the transformation is approximately $V^\dagger \cdot V = \mathbb{1}$.

Thus, applying \mathcal{Q} to $(|1\rangle\langle 1| \otimes \rho_{in})$ and measuring the ancilla qubit performs (almost) the desired non-unitary transformation when the ancilla is found to be $|0\rangle$ and almost the identity operator when the ancilla is found to be $|1\rangle$.

3.8.4 Success Probability

With what success probability can these outcomes be accomplished? To answer this, we define the measurement operators on the ancilla qubit to be M_0 and M_1 as:

$$M_0 = (|0\rangle\langle 0|) \otimes \mathbb{1} \quad (3.123)$$

$$M_1 = (|1\rangle\langle 1|) \otimes \mathbb{1} \quad (3.124)$$

then the probabilities of the two outcomes for the ancilla measurement can be computed as:

$$p_0 = \text{tr}(M_0^\dagger \cdot M_0 \cdot \rho_{out})$$

$$p_1 = \text{tr}(M_1^\dagger \cdot M_1 \cdot \rho_{out})$$

3.8.5 Fidelity when Successful

Similarly, we can ask, we are “successful” with what fidelity to we accomplish the desired non-unitary state transformation of ρ_{in} ? The density matrices conditioned on these two measurement outcomes are:

$$\rho_0 = \frac{M_0^\dagger \cdot M_0 \cdot \rho_{out}}{p_0}$$

$$\rho_1 = \frac{M_1^\dagger \cdot M_1 \cdot \rho_{out}}{p_1}$$

and the part of the state that contains the desired output is the reduced density matrix of the unmeasured qubits, i.e.,

$$\rho_{out}^{\text{actual}} = \text{tr}_1(\rho_0) \quad (3.125)$$

This should be compared to the desired density matrix:

$$\rho_{out}^{\text{desired}} = \frac{\mathcal{L} \cdot \rho_{in} \cdot \mathcal{L}^\dagger}{\text{tr}(\mathcal{L} \cdot \rho_{in} \cdot \mathcal{L}^\dagger)} \quad (3.126)$$

and the fidelity is given by:

$$F(\rho_{out}^{\text{actual}}, \rho_{out}^{\text{desired}}) = \text{tr} \left(\sqrt{\sqrt{\rho_{out}^{\text{actual}}} \cdot \rho_{out}^{\text{desired}} \cdot \sqrt{\rho_{out}^{\text{actual}}}} \right) \quad (3.127)$$

3.9 Summary

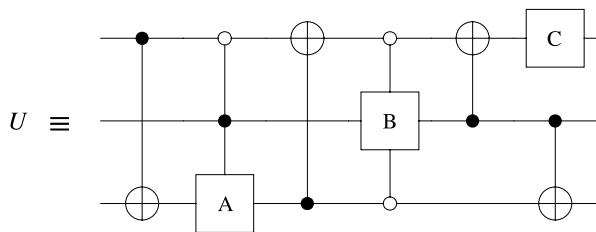
In this chapter we described how to compute the unitary matrix corresponding to a given quantum circuit, and how to compute a quantum circuit that achieves a given unitary matrix. In the forward direction (circuit to matrix) three matrix products turn out to be useful. The *direct product* (also known as the tensor or Kroenecker product) is used to describe quantum gates that act in parallel. Such gates are drawn vertically aligned over distinct subsets of qubits in quantum circuits. Similarly, the *dot product* is used to describe quantum gates that act sequentially. Sequential gates are drawn one after the other from left to right in a quantum circuit. When mapping from a sequential quantum circuit to its corresponding unitary matrix remember that if the circuit shows gate A acting first, then gate B , then gate C , the corresponding dot product describing these steps is $C \cdot B \cdot A$, where the ordering is reversed. Finally, we introduced the *direct sum*, which describes controlled quantum gates. These apply a quantum gate to some “target” subset of qubits depending on the qubit values on another set of “control” qubits. The controlling values can be 0 (white circles) or 1 (black circles), and combinations of control values are allowed. We remind you that in controlled quantum gates we do not have to read the control value in order to determine the action. Instead, the controlled quantum gates apply *all* the control actions consistent with the quantum state of the control qubits.

Certain types of quantum gates are considered as important primitives in quantum computing. The standard 1-qubit and 2-qubit elementary gates were introduced in Chap. 2. In Chap. 3 we built upon these to create more sophisticated n -qubit gates, such as quantum permutations (for qubit reversal, qubit cyclic left shift, and amplitude downshift), and quantum Fourier transform (QFT), quantum wavelet transform (QWT), and quantum cosine transform (QCT). These all admit anomalously compact, polynomially-sized, quantum circuits. Of these the QFT is the most important being at the heart of most quantum algorithms that admit exponential speedups. However, our hope is that by collecting together so many useful transforms and giving explicit quantum circuits for them, we might inspire the reader to compose them in novel ways to achieve new and useful quantum algorithms.

We also showed several techniques for decomposing a given unitary matrix into an equivalent quantum circuit, i.e., a sequence of 1-qubit and 2-qubit quantum logic gates. In so doing, we can use the gate identities of Chap. 2 to choose a particular family of quantum gates that is easiest to implement within some preferred quantum hardware scheme, because we believe it makes sense to tailor the decomposition of a unitary transformation to fit the chosen physical hardware, rather than to wrestle the physics to fit an ad hoc model of computation.

In particular, we presented numerical, algebraic, and re-use methods for quantum circuit design and gave examples of the use of each. A completely arbitrary $2^n \times 2^n$ unitary matrix does not admit an efficient (polynomially-sized) quantum circuit decomposition. However, the types of unitary matrices that arise in practically useful quantum algorithms are often not maximally general and in fact do admit efficient quantum circuits. If the circuits are small enough numerical template minimization can often find them. If the unitary matrices bear a special relationship to previously

Fig. 3.41 Quantum circuit implementing U



known unitary matrices, e.g., if they can be written as low order polynomials in known unitary matrices, we can sometimes apply the re-use method of circuit design. However, only the algebraic method of circuit design is guaranteed to work in every case. However, to achieve efficient quantum circuits using the algebraic design method one must apply circuit compactification rules during and after the design process to identify and strip away unnecessary gate inefficiencies.

Finally, we departed somewhat from the standard quantum circuit model by showing how to harness measurement operations in a useful way to achieve certain non-unitary quantum computations probabilistically. This scheme had the interesting feature that failed attempts to project the computation into the desired output are not totally destructive and one can use these outputs again to attempt to achieve the desired computation, albeit with a degraded fidelity.

3.10 Exercises

3.1 Decompose a general $R_y(\theta)$ rotation in terms of only R_x gates and R_z gates.

3.2 Draw a picture to show what the $R_y(\theta)$ gate does to the state $|1\rangle$ on the Bloch sphere.

3.3 Show how the decomposition of the Hadamard gate into R_y and R_z rotations allows us to predict how the Hadamard gate will move a state $|0\rangle$ on the Bloch sphere.

3.4 Quantum Circuit for Inverse Operation Look at the quantum circuit shown in Fig. 3.41 that implements a unitary operation U . Sketch the quantum circuits for the operations

- (a) U^2
- (b) U^{-1}
- (c) U^\dagger

3.5 Quantum Circuit for the FREDKIN Gate In Chap. 2, you saw the FREDKIN gate, which is a (3-bit)-to-(3-bit) universal gate for classical reversible computing.

1. Prove that matrix describing the action of the FREDKIN gate is unitary, and hence admits a quantum circuit decomposition.

2. By regarding the FREDKIN gate as a controlled-SWAP gate, write down a quantum circuit for the FREDKIN gate in terms of TOFFOLI gates, and hence, controlled 2-qubit gates.
3. Is the quantum circuit you found in part (2), the most compact quantum circuit for the FREDKIN gate? Exhibit a more compact quantum circuit for the FREDKIN gate or explain why the quantum circuit you found in part (2) is already optimal.

3.6 Alternative Factorization of an Arbitrary Unitary Matrix There are other procedures for factorizing an arbitrary $2^n \times 2^n$ dimensional unitary matrix, U other than the scheme based on the GSVD presented in this chapter. One method, is to multiply U on the left (say) by matrices that systematically zero out chosen elements of U until only the identity matrix is left. Specifically, show that:

1. You can zero the i -th element of the leftmost column of U , by multiplying U with the matrix V_1 given by:

$$V_1 = \begin{pmatrix} a^\dagger & 0 & \cdots & b^\dagger & \cdots \\ 0 & 1 & \cdots & & \\ \vdots & & \ddots & & \\ b & \cdots & \cdots & -a & \cdots \\ \vdots & & & \vdots & \ddots \end{pmatrix} \quad (3.128)$$

where

$$a = \frac{u_{11}}{\sqrt{|u_{11}|^2 + |u_{i1}|^2}} \quad (3.129)$$

$$b = \frac{u_{i1}}{\sqrt{|u_{11}|^2 + |u_{i1}|^2}} \quad (3.130)$$

2. That V_1 , and hence $V_1 \cdot U$ are unitary.
3. Show that repeating the process of zeroing out the i -th element that in $2^n - 1$ steps you can obtain the unitary matrix $V_{2^n-1} \cdot V_2 \cdot V_1 \cdot U$ of the form:

$$V_{2^n-1} \cdot V_2 \cdot V_1 \cdot U = \begin{pmatrix} 1 & 0 & 0 & \cdots \\ 0 & u'_{22} & u'_{23} & \cdots \\ 0 & u'_{32} & \ddots & \\ \vdots & \vdots & & \end{pmatrix} \quad (3.131)$$

4. Repeat this procedure on the inner $(2^n - 1) \times (2^n - 1)$ dimensional unitary matrix, using a matrix of the form V'_1 where

$$V'_1 = \begin{pmatrix} 1 & 0 & \cdots & & & \\ 0 & a^\dagger & 0 & \cdots & b^\dagger & \cdots \\ 0 & 0 & 1 & \cdots & & \\ \vdots & \vdots & & \ddots & & \\ 0 & b & \cdots & \cdots & -a & \cdots \\ \vdots & \vdots & & & \vdots & \ddots \end{pmatrix}. \quad (3.132)$$

and so on, eventually bottoms out at the identity matrix.

- 3.7** In the text we claimed that the classical Type II Discrete Cosine Transform (DCT-II), as given by equation 3.58, of a signal $S = \{f_0, f_1, \dots, f_{N-1}\}$, having N sample values, is related to the Discrete Fourier Transform (DFT) of a signal $S' = \{0, f_0, 0, f_1, 0, \dots, f_{N-1}, 0, f_{N-1}, 0, f_{N-2}, 0, \dots, f_1, 0, f_0\}$, having $4N$ values. Verify this claim for the signal consisting of the eight discrete values, $S = \{\frac{1}{\sqrt{74}}, 2\sqrt{\frac{2}{37}}, \frac{3}{\sqrt{74}}, -\frac{1}{\sqrt{74}}, -\sqrt{\frac{2}{37}}, -\frac{3}{\sqrt{74}}, \frac{5}{\sqrt{74}}, \frac{3}{\sqrt{74}}\}$. Note that the unitary matrix defining the DFT is the same as that defining the QFT. Show further that this relationship breaks down if we use instead the orthogonalized version of the classical DCT-II transform as defined by (3.59).

- 3.8** Using $N = 4$, write down the definitions of the matrices V_{2N} and U_{2N} given by (3.64) and (3.67) respectively. By computing $U_{2N}^\dagger \cdot \text{QFT}_{2N} \cdot V_{2N}$ verify that the result contains the Type II Quantum Cosine Transform in its upper left quadrant. Interpret what this means as a conditional logic gate.

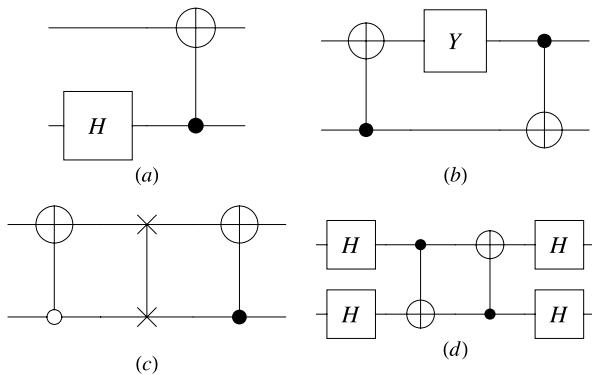
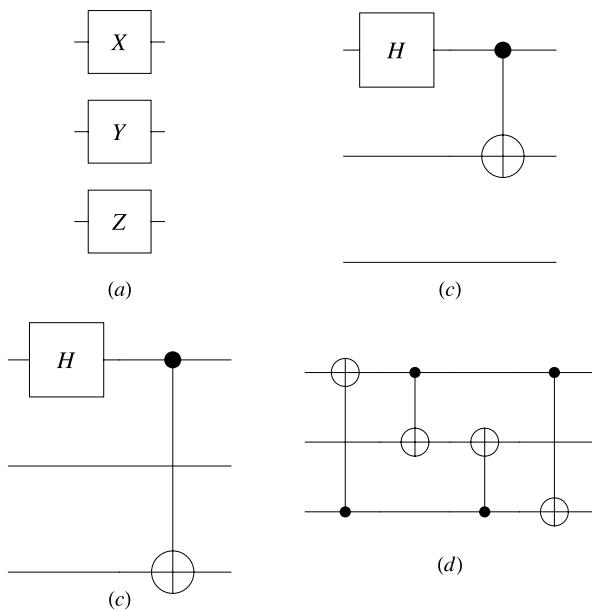
- 3.9** Verify that the permutation matrix Q_{2^n} defined in (3.17) can be factored in accordance with $Q_{2^n} = P_{2^n} [\bigodot_{i=1}^n (X \otimes \mathbb{1}_{2^{n-i}}) \oplus \mathbb{1}_{2^n - 2^{n-i+1}}] \cdot P_{2^n}$.

- 3.10** What are the unitary matrices implied by the circuits shown in Fig. 3.42?

- 3.11** What are the unitary matrices implied by the circuits shown in Fig. 3.43?

- 3.12** It is often useful to represent a given unitary operator in a different basis:

- (a) Write the CNOT gate in the $\{|\bar{0}\rangle, |\bar{1}\rangle\}$ -basis where $|\bar{0}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$ and $|\bar{1}\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$.
- (b) Write the iSWAP gate in the Bell basis $\{|\beta_{00}\rangle, |\beta_{01}\rangle, |\beta_{10}\rangle, |\beta_{11}\rangle\}$.

Fig. 3.42 Some 2-qubit gates**Fig. 3.43** In (a) the gates act in parallel. In (b) the gates act on only a subset of the qubits. In (c) and (d) the CNOT gates act on non-adjacent qubits

3.13 Given the 1-qubit gates C_0 and C_1 defined by:

$$C_0 = 2 \begin{pmatrix} c_3 & -c_2 \\ c_2 & c_3 \end{pmatrix}$$

$$C_1 = \frac{1}{2} \begin{pmatrix} \frac{c_0}{c_3} & 1 \\ 1 & \frac{c_1}{c_2} \end{pmatrix}$$

where $c_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}$, $c_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}$, $c_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}$, and $c_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}$,

- (a) prove that C_0 and C_1 are unitary,
- (b) factor them in terms of R_z , R_y , and Ph gates, and

- (c) verify the factorization of the Daubechies $D_{2^n}^{(4)}$ wavelet transform given by (3.54).

3.14 The following quantum arithmetic operations arose in the quantum circuit decomposition of the Type II Quantum Cosine Transform: the controlled-One's Complement operation (C-OC defined in (3.72)), the controlled-Two's Complement operation (C-TC defined in (3.74)), and the Controlled-Modular-Add-One operation (C-MAO defined in (3.80)). Find quantum circuits for $C-OC_{2^n}$, $C-TC_{2^n}$, and $C-MAO_{2^n}$ where the subscript indicates the dimension of the associate unitary matrix. Note that these quantum arithmetic operations are anomalously easy compared to a general unitary transformation of the same size.

3.15 The Type II Quantum Cosine Transform can be obtained from the factorization given by (3.81), i.e.,

$$U_{2N}^\dagger \cdot \text{QFT}_{2N} \cdot V_{2N} = \text{QCT}_N^{\text{II}} \oplus -\text{QST}_N^{\text{II}}$$

In this factorization, QFT_{2N} and V_{2N} are straightforward, but the factorization of U_{2N}^\dagger is more involved. Verify the factorization of U_{2N}^\dagger given in (3.84) and sketch a quantum circuit reflecting this factorization.

3.16 Verify the “design-by-re-use” method of quantum circuit design by checking the quantum circuit for QFFT can be written in terms of controlled-powers-of-QFT by checking the factorization for QFFT given by:

$$\begin{aligned} U_1 &= H \otimes H \otimes \mathbb{1}_N \\ U_2 &= \mathbf{1}_2 \otimes (\mathbb{1}_N \oplus \text{QFT}_N^{-1}) \\ U_3 &= \text{SWAP}_{1,2;n+2} \cdot \mathbf{1}_2 \otimes (\mathbf{1}_N \oplus \text{QFT}_N^{-2}) \cdot \text{SWAP}_{1,2;n+2} \\ U_4 &= C_\alpha \otimes \mathbb{1}_N \\ U_5 &= \text{SWAP}_{1,2;n+2} \cdot \mathbf{1}_2 \otimes (\mathbb{1}_N \oplus \text{QFT}_N^2) \cdot \text{SWAP}_{1,2;n+2} \\ U_6 &= \mathbf{1}_2 \otimes (\mathbb{1}_N \oplus \text{QFT}_N) \\ U_7 &= H \otimes H \otimes \mathbb{1}_N \\ \text{QFFT}_{N;\alpha} &= U_7 \cdot U_6 \cdot U_5 \cdot U_4 \cdot U_3 \cdot U_2 \cdot U_1 \end{aligned} \tag{3.134}$$

Chapter 4

Quantum Universality, Computability, & Complexity

“[...] Turing’s theory is not entirely mathematical [...]. It makes hidden assumptions about physics which are not quite true. Turing and other physicists who constructed universal models for classical computation tried hard not to make any assumptions about the underlying physics [...]. But their intuition did not encompass quantum theory, and their imaginary paper did not exhibit quantum coherence.”

– David Deutsch¹

Once while visiting Stephen Hawking in Cambridge, England, Stephen asked me what I was working on. At the time I was a research scientist at Xerox PARC developing what later became called the theory of computational phase transitions, which is a view of computation inspired by statistical physics that I will describe in Chap. 7. However, since the term “computational phase transition” was generally unknown at that time, I replied by saying I was working on “computational complexity theory”. I distinctly recall an expression of disdain sweep across Stephen’s face, and the conversation quickly switching to something else. In retrospect, Stephen’s pained expression turned out to be prophetic for many subsequent conversations I have had with other physicists. It appears physicists are not generally enamored with computational complexity theory!

Why is this? In part, I believe it is a cultural difference. I have found that physicists tend to embrace simplified approximate models that encourage comprehension, whereas computer scientists tend to prefer detailed exact models about which strong theorems can be proved. Neither style is right nor wrong—just different. Moreover, physicists have an uncanny knack for picking terminology that is vivid, and alluring, e.g., “Big Bang”, “dark matter”, “black hole”, “twin-paradox”, “strange attractor” etc., whereas theoretical computer science is replete with the most über-geeky nomenclature imaginable as exemplified by the byzantine names of computational complexity classes. My complaint is not so much about the archaic names theoretical computer scientists have chosen, but the ad hoc ways in which the system of names has been expanded. Had we done the same with organic chemistry key

¹Source: in David Deutsch, “Quantum Computation,” Physics World, June (1992) pp. 57–61.

insights and generalizations might have been missed. Had we picked a more systematic naming convention that aids comprehension of the concepts underpinning the complexity classes and how they differ from one another, then perhaps greater insights, or more useful classes, might have been discovered. The current nomenclature does not, in my opinion, assist comprehension of the underlying complexity class distinctions and their interrelationships.

Despite these differences, both fields have revealed extremely counter-intuitive, intriguing, and profound results. In this chapter, we highlight some of these amazing results from theoretical computer science and ask whether or not they still hold true in the quantum domain.

First, there is the question of *complexity*: Can a quantum computer perform the same tasks as a classical computer, but in significantly fewer steps? Second, there is the question of *computability*; Can a quantum computer *perform* computations that a classical computer cannot? And finally there is the question of *universality*; Is there a specialized quantum computer that can simulate any other quantum computer, and classical computer, *efficiently*? A difference between the capabilities of a quantum computer and those of a classical computer on any one of these criteria would be significant.

4.1 Models of Computation

To answer questions about complexity, universality, and computability, one must have a model of computation in mind. In the 1930's three superficially different models of computation were invented by Alan Turing, Emil Post, Kurt Gödel and Alonzo Church.

4.1.1 *The Inspiration Behind Turing's Model of Computation: The Entscheidungsproblem*

In 1900, Hilbert gave an address at the International Congress of Mathematics held in Paris concerning what he believed to be the 23 most challenging mathematical problems of his day. The last problem on his list asked whether there was a mechanical procedure by which the truth or falsity of any mathematical conjecture could be decided. In German, the word for "decision" is "entscheidung," so Hilbert's 23rd problem became known as the "*Entscheidungsproblem*". Turing's abstract model of computation grew out of his attempt to answer the Entscheidungsproblem.

Hilbert's motivation for asking this question arose from the trend towards abstraction in mathematics. Throughout the 19th century, mathematics was largely a practical matter, concerned with making statements about real-world objects. In the late 1800s mathematicians began to invent, and then reason about, imaginary objects to which they ascribed properties that were not necessarily compatible with

“common sense.” Thus the truth or falsity of statements made about such imaginary objects could not be determined by appealing to the real world. In an attempt to put mathematical reasoning on secure logical foundations, Hilbert advocated a “formalist” approach to proofs. To a formalist, symbols cease to have any meaning other than that implied by their relationships to one another. No inference is permitted unless there is an explicit rule that sanctions it, and no information about the meaning of any symbol enters into a proof from outside itself. Thus the very philosophy of mathematics that Hilbert advocated seemed very machine-like, and hence Hilbert proposed the *Entscheidungsproblem*.

Turing heard about Hilbert’s *Entscheidungsproblem* during a course of lectures, given by Max Newman, which he attended at Cambridge University. In his lecture Newman had described the *Entscheidungsproblem* as asking whether there was be a “mechanical” means of deciding the truth or falsity of a mathematical proposition. Although Newman probably meant “mechanical” figuratively, Turing interpreted it literally. Turing wondered whether a machine could exist that would be able to decide the truth or falsity of any mathematical proposition. Thus, in order to address the *Entscheidungsproblem*, Turing realized that he needed to model the process in which a human mathematician engages when attempting to prove some mathematical conjecture.

Mathematical reasoning is an enigmatic activity. We do not really know what goes on inside a mathematician’s head, but we can examine the result of his thought processes in the form of the notes he creates whilst developing a proof. Mathematical reasoning consists of combining axioms (statements taken to be true without proof) with rules of logical inference, to infer consequents, which themselves become additional nuggets of information upon which further inferences may be drawn. So the reasoning process builds on itself and will result in valid conclusions provided the starting axioms are correct and the rules of inference are valid.

Turing abstracted the process followed by the mathematician into four principal ingredients: a set of transformation rules that allowed one mathematical statement to be transformed into another; a method for recording each step in the proof, an ability to go back and forth over the proof to combine earlier inferences with later ones, and a mechanism for deciding which rule to apply at any given moment. This is the essence of the proof process (at least its visible part). Next, Turing sought to simplify these steps in such a way that a machine could be made to imitate them. Mathematical statements are built up out of a mixture of ordinary letters, numbers, parentheses, operators (e.g., plus, “+” and times “ \times ”) and special mathematical symbols (e.g., \forall , \exists , \neg , \wedge , \vee). Turing realized that the symbols themselves were of no particular significance. All that mattered was that they were used consistently and that their number was finite. Moreover, once you know you are dealing with a finite alphabet, you can place each symbol in one-to-one correspondence with a unique pattern of any two symbols (such as 0 and 1). Hence, rather than deal with a rich array of esoteric symbols, Turing realized that a machine only needed to be able to read and write two kinds of symbol, 0 and 1, say, with blank spaces or some other convention to identify the boundaries between the distinct symbols. Similarly, the fact that the scratch pad on which the mathematician writes intermediate results

is two-dimensional is of no particular importance. You could imagine attaching the beginning of one line of a proof to end of the previous line, making one long continuous strip of paper. So, for simplicity, Turing assumed that the proof could be written out on a long strip of paper or a “tape.” Moreover, rather than allowing freeform handwriting, it would clearly be easier for a machine to deal with a tape marked off into a sequence of identical cells and only permitting one symbol to be written inside each cell, or the cell to be left blank.

Finally, the process of the mathematician going back and forth over previous conclusions in order to draw new ones could be captured by imagining that there is a “read/write” head going back and forth along the tape. When a mathematician views an earlier result it is usually in some context. A mathematician might read a set of symbols, write something, but come back to read those same symbols again later, and write something else. Thus, the context in which a set of symbols is read can affect the subsequent actions. Turing captured this idea by defining the “head” of his Turing machine to be in certain “states,” corresponding to particular contexts. The combination of the symbol being read under the head and the state of the machine determined what symbol to write on the tape, which direction to move the head, and which state to enter next.

This is clearly a crude model of the proof process. Nevertheless it turned out to be surprisingly powerful. No matter what embellishments people dreamed up, Turing could always argue that they merely were refinements to some existing part of the model rather than being fundamentally new features. Consequently the Turing machine model was indeed the essence of the proof process. By putting the aforementioned mechanistic analogues of human behavior into a mathematical form, Turing was led to the idea of a “deterministic Turing machine”.

4.1.2 Deterministic Turing Machines

The most influential model of computation was invented by Alan Turing in 1936 [501]. A Turing machine is an idealized mathematical model of a computer that can be used to understand the limits of what computers can do [237]. It is not meant to be a practical design for any actual machine but rather a simplified abstraction that, nevertheless, captures the essential features of any real computer. A Turing machine’s usefulness stems from being sufficiently simple to allow mathematicians to prove theorems about its computational capabilities and yet sufficiently complex to accommodate any actual classical digital computer, no matter how it is implemented in the physical world.

A deterministic Turing machine is illustrated in Fig. 4.1. Its components are inspired by Turing’s abstract view mathematical reasoning. A deterministic Turing machine consists of an infinitely long tape that is marked off into a sequence of cells on which may be written a 0 or a 1, and a read/write head that can move back and forth along the tape scanning the contents of each cell. The head can exist in one of a finite set of internal “states” and contains a set of instructions (constituting

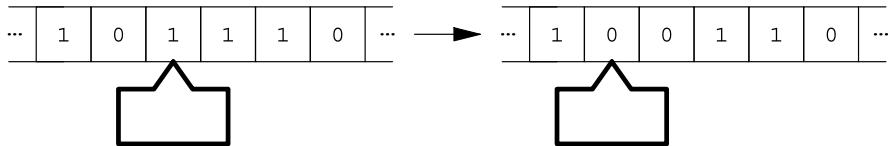


Fig. 4.1 A deterministic Turing machine

the “program”) that specifies, given the current internal state, how the state must change given the bit (i.e., the binary digit 0 or 1) currently being read under the head, whether that bit should be changed, and in which direction the head should then be advanced.

The tape is initially set up in some standardized state such as all cells containing 0 except for a few that hold the program and any initial data. Thereafter the tape serves as the scratch pad on which all intermediate results and the final answer (if any) are written.

Despite its simplicity, the Turing Machine model has proven to be remarkably durable. In the 70-odd years since its inception, computer technology has advanced considerably. Nevertheless, the Turing machine model remains as applicable today as it was back in 1936. Although we are apt to think of multimillion dollar supercomputers as being more powerful than humble desktop machines, the Turing machine model proves otherwise. Given enough time and memory capacity there is not a single computation that a supercomputer can perform that a personal computer cannot also perform. In the strict theoretical sense, they are equivalent. Thus the Turing machine is the foundational upon which much of current computer science rests. It has enabled computer scientists to prove many theorems that bound the capabilities of computing machinery.

More recently, however, a new idea has emerged that adds a slight twist to the deterministic Turing machine. Deterministic Turing machines, which follow rigid pre-defined rules, are susceptible to systematic biases that can cause them to take a very long time to solve certain problems. These are the problems for which the particular set of deterministic rules happen to make the Turing machine examine almost all the potential solutions before discover an actual solution. For example, if an adversary knew the rules by which a give DTM operated they could devise a problem that was guarantee to tax the machine to its maximum before finding a true solution. To avoid such pitfalls, a new type of Turing machine was invented that employs randomness, this is called a probabilistic, or non-deterministic, Turing machine.

4.1.3 Probabilistic Turing Machines

An alternative model of classical computation is to equip a deterministic Turing machine with the ability to make a random choice, such as flipping a coin. The result is a probabilistic Turing machine. Surprisingly, many problems that take a

long time to solve on a deterministic Turing machine (DTM) can often be solved very quickly on a probabilistic Turing machine (PTM).

In the probabilistic model of computation there are often tradeoffs between the time it takes to return an answer to a computation and the probability that the answer returned is correct. For example, suppose you wanted to plan a round the world trip that visited 100 cities, but you wanted to minimize the distance you have to travel between cities and you only wanted to visit each city once. The problem of computing the optimal (shortest path) route for your trip is extremely demanding computationally. However, if you were prepared to accept a route that was guaranteed to be only a little bit longer than the optimal route, and could in fact be the optimal route, then this problem is very easy to solve computationally. For example, the Euclidean TSP is known to be an **NP-Complete** problem [377], which means that, to the best knowledge of computer scientists at the present time, the computational cost of finding the *optimal* tour scales exponentially with the number of cities to be visited, N , making the problem intractable for sufficiently large N . Nevertheless, there is a randomized algorithm that can find a tour to within $\mathcal{O}(1 + 1/c)$ of the optimal tour (for any constant c) in a time that scales only as $\mathcal{O}(N(\log(N))^{O(c)})$ [20], which is worse than linear but better than exponential scaling. Thus, randomization can be a powerful tool for rendering intractable problems tractable provided we are content with finding a good approximation to the optimal or exact solution.

An alternative tradeoff, if you *require* a correct answer, is to allow uncertainty in the length of time the probabilistic algorithm must run before it returns an answer. Consequently, a new issue enters the computational theory, namely, the correctness of an answer and its relationship to the running time of an algorithm.

Whereas a deterministic Turing Machine, in a certain state, reading a certain symbol, has precisely one successor state available to it, the probabilistic Turing machine has multiple legitimate successor states available, as shown in Fig. 4.2. The choice of which state is the one ultimately explored is determined by the outcome of a random choice (possibly with a bias in favor of some states over others). In all other respects the PTM is just like a DTM. Despite the superficial difference between PTMs and DTMs, computer scientists have proved that anything computable by a probabilistic Turing machine can also be computed by a deterministic Turing machine, although in such cases the probabilistic machine is often more efficient [198]. The basic reason for the success of probabilistic approach is that a probabilistic algorithm can be thought of as swapping between a collection of deterministic algorithms. Whereas it is fairly easy to design a problem so that it will mislead a particular deterministic algorithm, it is much harder to do so for a probabilistic algorithm because it keeps on changing its “identity.” Indeed the latest algorithms for solving hard computational problems now interleave deterministic, with probabilistic steps. The exact proportions of each strategy can have a huge impact on the overall efficiency of problem solving.

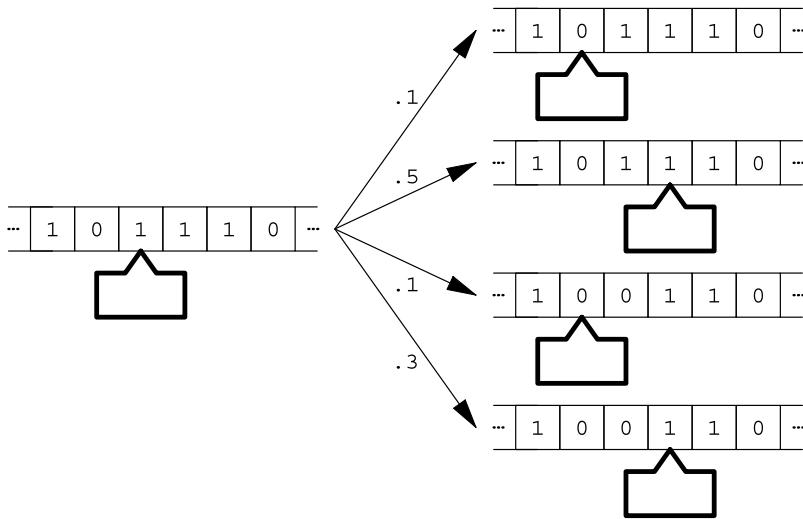


Fig. 4.2 In a probabilistic classical Turing machine there are multiple possible successor states, only one of which is actually selected and pursued at any one time

4.1.4 The Alternative Gödel, Church, and Post Models

Kurt Gödel invented a very different model of computation than that formulated by Turing. Gödel identified the tasks that a computer can perform with a class of *recursive functions*, i.e., functions that refer to themselves. For example, the function $\text{fib}(x) = \text{fib}(x - 1) + \text{fib}(x - 2)$ such that $\text{fib}(1) = \text{fib}(2) = 1$ defines a recursive function that generates the Fibonacci sequence, i.e., as x takes on integer values $x = 1, 2, 3, 4, 5, 6, \dots$, then $f(x)$ generates the Fibonacci numbers $1, 1, 2, 3, 5, 8, \dots$. The function $\text{fib}(\cdot)$ is defined in terms of itself, and is therefore a recursive function.

Yet another model of computation was formulated by Alonzo Church. Church equated the tasks that a computer can perform with the so-called λ -definable functions (which you will have encountered if you have ever used the LISP programming language). This viewed computation as a nesting of function evaluations. The simplicity of the λ -calculus made it possible to prove various properties of computations.

Hence both Gödel's and Church's formulations of computation viewed it as an elaborate mathematical function evaluation in which simpler functions were composed to make more elaborate ones.

Emil Post anticipated many of the results of Gödel, Turing, and Church but chose not publish them. His “finite combinatory processes—Formulation I” [397] is similar in spirit to the idea of a Turing machine. Post did not ever speak overtly of computing machines, but he did invent (independently of Turing) the idea of a human worker moving along a two way infinite “workspace” of boxes each of which could be marked or unmarked, and following a set of directions: a conditional jump, “Stop”, move left, move right, mark box or unmark box.

4.1.5 Equivalence of the Models of Computation

Thus, Turing identified the tasks a computer can perform with the class of functions computable by a hypothetical computing device called a *Turing Machine*. This viewed computation a rather imperative or “procedural” style. Slightly later Emil Post also formalized computation in a similar machine model, which he asserted was “logically equivalent to recursiveness”. Kurt Gödel equated computation with recursive functions and Alonzo Church with λ -definable functions.

Although, superficially, the models of computation advanced by Turing, Gödel, Church, and Post look different, it turns out that they are *equivalent* to one another. This was something of a surprise as there was no reason to expect their equivalence *a priori*.

Moreover, any one of the models alone might be open to the criticism that it provided an incomplete account of computation. But the fact that three radically different views of computation all turned out to be equivalent was a clear indication that the most important aspects of computation had been characterized correctly.

4.2 Universality

In the 1930s computer science was a rather fledgling field. People dabbled with building computers but very few machines actually existed. Those that did had been tailor-made for specific applications. However, the concept of a Turing machine raised new possibilities. Turing realized that one could encode the transformation rules of any particular Turing machine, T say, as some pattern of 0s and 1s on the tape that is fed into some special Turing machine, called U . U had the effect of reading in the pattern specifying the transformation rules for T and thereafter treated any further input bits exactly as T would have done. Thus U was a universal mimic of T and hence was called the *Universal Turing Machine*. Thus, one Turing machine could mimic the behavior of another.

4.2.1 The Strong Church-Turing Thesis

The ability to prove that all the competing models of classical computation were equivalent led Church to propose the following principle, which has subsequently become known as the Church-Turing thesis [450]:

Strong Church-Turing Thesis *Any process that is effective or algorithmic in nature defines a mathematical function belonging to a specific well-defined class, known variously as the recursive, the λ -definable, or the Turing computable functions. Of, in Turing’s words, every function which would naturally be regarded as computable can be computed by the universal Turing machine.*

Thus a model of computation is deemed *universal*, with respect to a family of alternative models of computation, if it can compute any function computable by those other models either directly or via emulation.

4.2.2 Quantum Challenge to the Strong Church-Turing Thesis

Notwithstanding these successes, in the early 1980s a few maverick scientists began to question the correctness of the classical models of computation. The deterministic Turing machine and probabilistic Turing machine models are certainly fine as *mathematical abstractions* but are they consistent with known *physics*? This question was irrelevant in Turing's era because computers operated at a scale well above that of quantum systems. However, as miniaturization progresses, it is reasonable, in fact, necessary, to re-consider the foundations of computer science in the light of our improved understanding of the microscopic world.

Unfortunately, we now know that although these models were intended to be mathematical abstractions of computation that were free of physical assumptions, they do, in fact, harbor implicit assumptions about the physical phenomena available to a computer. These assumptions appear to be perfectly valid in the world we see around us, but they cease to be valid on sufficiently small scales.

We now know that the Turing Machine model contains a fatal flaw. In spite of Turing's best efforts, some remnants of classical physics, such as the assumption that a bit must be either a 0 *or* a 1, crept into the Turing machine models. The obvious advances in technology, such as more memory, more instructions per second, greater energy efficiency have all been merely quantitative in nature. The underlying foundations of computer science have not changed. Similarly, although certainly having a huge social impact, apparent revolutions, such as the explosion of the Internet, have merely provided new conduits for information to be exchanged. They have not altered the fundamental capabilities of computers in any way whatsoever. However, as computers become smaller, eventually their behavior *must* be described using the physics appropriate for small scales, that is, quantum physics.

The apparent discrepancy between Feynman's observation that classical computers cannot simulate quantum system efficiently and the Church-Turing thesis means that the Strong Church-Turing Thesis may be flawed for there is no known way to simulate quantum physics efficiently on any kind of classical Turing machine. This realization led David Deutsch in 1985 to propose reformulating the Church-Turing thesis in physical terms. Thus Deutsch prefers:

Deutsch's Thesis *Every finitely realizable physical system can be perfectly simulated by a universal model computing machine operating by finite means.*

This can only be made compatible with Feynman's observation on the efficiency of simulating quantum systems by basing the universal model computing machine on quantum mechanics itself. This insight was the inspiration that allowed David

Deutsch to prove that it was possible to devise a “Universal Quantum Turing Machine”, i.e., a quantum Turing machine that could simulate any other quantum Turing machine. The efficiency of Deutsch’s Universal Quantum Turing Machine has since been improved upon by several other scientists.

We don’t yet know how history will rate the relative contributions of various scientists to the field of quantum computing. Curiously though, if you search for “quantum computing” at www.wikiquote.com you will discover “David Deutsch, Relevance: 4.2%; Richard Feynman, Relevance: 2.2% and (my personal favorite) God, Relevance: 0.9%”. I have to say that I think wikiquote has it about right! I certainly concur with the relative ratings of Deutsch’s and Feynman’s contributions, but I will leave it to each author (one living, one dead) to argue with the Almighty Himself, the merits of their ranking with respect to God.

4.2.3 Quantum Turing Machines

The first quantum mechanical description of a Turing machine was given by Paul Benioff in 1980 [43]. Benioff was building on earlier work carried out by Charles Bennett who had shown that a reversible Turing machine was a theoretical possibility [44].

A reversible Turing machine is a special version of a deterministic Turing machine that never erases any information. This is important because physicists had shown that, in principle, all of the energy expended in performing a computation can be recovered provided that the computer does not throw any information away. The notion of “throwing information away” means that the output from each step of the machine must contain within it enough information that the step can be undone without ambiguity. Thus, if you think of a reversible Turing machine as a dynamical system, then given knowledge of its state at any one moment would allow you to predict its state at all future and all past times. No information was ever lost and the entire computation could be run forwards or backwards.

This fact struck a chord with Benioff, for he realized that any isolated quantum system had a dynamical evolution that was reversible in exactly this sense. Thus it ought to be possible to devise a quantum system whose evolution over time mimicked the actions of a classical reversible Turing machine. This is exactly what Benioff did. Unfortunately, Benioff’s machine is not a true quantum computer. Although between computational steps the machine exists in an intrinsically quantum state (in fact a “superposition,” of computational basis states, at the end of each step the “tape” of the machine was always back in one of its classical states: a sequence of classical bits. Thus, Benioff’s design could do no more than a classical reversible Turing machine.

The possibility that quantum mechanical effects might offer something genuinely new was first hinted at by Richard Feynman of Caltech in 1982, when he showed that no classical Turing machine could simulate certain quantum phenomena without incurring an unacceptably large slowdown but that a “universal quantum simulator”

could do so. Unfortunately, Feynman did not provide a design for such a simulator, so his idea had little immediate impact. Nor did he did not prove, conclusively, that a universal quantum simulator was possible. However, indeed it is. The question was answered in the affirmative by Seth Lloyd in 1996 [321].

The key step in making it possible to study the computational power of quantum computers came in 1985, when David Deutsch of Oxford University, described the first true quantum Turing machine (QTM) [136]. A QTM is a Turing machine whose read, write, and shift operations are accomplished by quantum mechanical interactions and whose “tape” can exist in states that are highly nonclassical. In particular, whereas a conventional classical Turing machine can only encode a 0, 1, or blank in each cell of the tape, the QTM can exist in a blend, or “superposition” of 0 and 1 simultaneously. Thus the QTM has the potential for encoding many inputs to a problem simultaneously on the same tape, and performing a calculation on all the inputs in the time it takes to do just one of the calculations classically. This results in a superposition of all the classical results and, with the appropriate measurement, you can extract information about certain joint properties of all these classical results. This technique is called “quantum parallelism.” We saw an example of quantum parallelism when we solved Deutsch’s problem in Chap. 1.

Moreover, the superposition state representing the tape of the QTM can correspond to an entanglement of several classical bit string configurations. Entanglement means that the quantum state of the entire tape is well-defined but the state of the individual qubits is not. For example, a 3-qubit tape in the state $\frac{1}{\sqrt{2}}(|010\rangle + |101\rangle)$ represents an entanglement of the two configurations $|010\rangle$ and $|101\rangle$. It is entangled in the sense that if you were to measure any one of these qubits, the quantum state of the other two qubits would become definite instantaneously. Thus, if you read out the bit values from a part of the tape of the QTM when it is in an entangled state, your actions will have a side effect on the state of the other (unmeasured) qubits. In fact it is the existence of such “entangled” qubits that is the fundamental reason QTMs are different from classical deterministic and probabilistic TMs.

A graphical representation of a QTM is shown in Fig. 4.3. There is a *single* physical tape running from left to right in the figure. However, this single tape is drawn as if it were several tapes in parallel to convey the idea that the single quantum tape can hold a superposition of many different bit strings simultaneously.

As we saw in Chap. 1, each qubit in a QTM, when considered in isolation from other qubits, can be visualized as a small arrow contained in a sphere. “Straight up” represents the (classical) binary value 0 and “straight down” represents the (classical) binary value 1. When the arrow is at any other orientation, the angle the arrow makes with the horizontal axis is a measure of the ratio of 0-ness to 1-ness in the qubit. Likewise, the angle through which the arrow is rotated about the vertical axis is a measure of the “phase”. Thus, drawing qubits as arrows contained in spheres we can depict a typical superposition state of Deutsch’s quantum Turing machine as shown in Fig. 4.3. The possible successor states of the tape are indicated by edges between different possible tape configurations.

Quantum Turing machines (QTMs) are best thought of as quantum mechanical generalizations of probabilistic Turing machines (PTMs). In a PTM, if you initialize

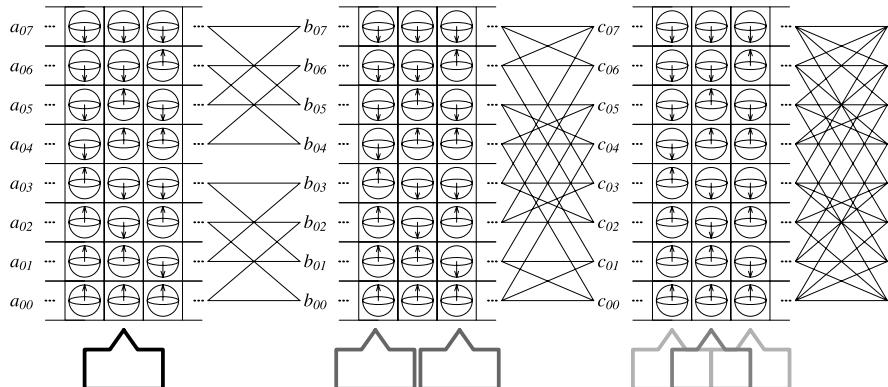


Fig. 4.3 In the quantum Turing machine, each cell on the tape can hold a qubit. In this figure there is *one* physical tape but it is drawn as multiple tapes corresponding to a different bit pattern for each component of the net superposition state

the tape in some starting configuration and run the machine without inspecting its state for t steps, then its final state will be uncertain and can only be described using a probability distribution over all the possible states accessible in t steps.

Likewise, in a QTM if you start the machine off in some initial configuration, and allow it to evolve for t steps, then its state will be described by a superposition of all states reachable in t steps. The key difference is that in a classical PTM only one particular computational trajectory is followed, but in the QTM *all* computational trajectories are followed and the resulting superposition is the sum over all possible states reachable in t steps. This makes the calculation of the net probability of a particular computational outcome different for a PTM than a QTM.

In the PTM if a particular answer can be reached independently, in more than one way, the net *probability* of that answer is given by the sum of each probability that leads to that answer. However, in the QTM if a given answer can be reached in more than one way the net probability of obtaining that answer is given by summing the *amplitudes* of all trajectories that lead to that answer and then computing their *absolute value squared* to obtain the corresponding probabilities.

If the quantum state of the QTM in Fig. 4.3 is the superposition $c_0|00000\rangle + c_1|00001\rangle + c_2|00010\rangle + \dots + c_{31}|11111\rangle$ the coefficients c_0, c_1, \dots, c_{31} are the amplitudes, and probability of finding the tape of the QTM in the bit configuration $|00010\rangle$, say, when you read each of the bits is equal to $|c_2|^2$. If an event occurs with a probability of 0 this means that there is a 0% chance, i.e., utter impossibility, of that event occurring. Conversely, if an event occurs with a probability of 1 this means that there is a 100% chance, i.e., absolutely certainty, that the event will occur.

Whereas classical probabilities are real numbers between zero and one, “amplitudes” are complex numbers (i.e. numbers of the form $x + iy$ where x and y are real numbers). When you add two probabilities you always get a bigger or equal probability. However, when you add two complex amplitudes together they do not always result in a number that has a bigger absolute value. Some pairs of amplitudes

tend to cancel each other out resulting in a net reduction in the probability of seeing a particular outcome. Other pairs of amplitudes tend to reinforce one another and thereby enhance the probability of a particular outcome. This is the phenomenon of quantum interference.

Quantum interference is a very important mechanism in quantum computing. Typically, when designing a quantum computer to solve a hard computational problem, you have to devise a method (in the form of a quantum algorithm) to evolve a superposition of all the valid inputs to the problem into a superposition of all the valid solutions to that problem. If you can do so, when you read the final state of your memory register you will be guaranteed to obtain one of the valid solutions. Understanding how to achieve your desired evolution invariably entails arranging for the computational pathways that lead to non-solutions to interfere destructively with one another and hence cancel out, and arranging for the computational pathways that lead to solutions to interfere constructively and hence reinforce one another.

Armed with this model of an abstract quantum Turing machine, several researchers have been able to prove theorems about the capabilities of quantum computers [58]. This effort has focused primarily on *universality* (whether one machine can simulate all others efficiently), *computability* (what problems the machines can do), and *complexity* (how the memory, time and communication resources scale with problem size). Let us take a look at each of these concepts and compare the perspective given to us by classical computing and quantum computing.

4.3 Computability

Computability theory is concerned with which computational tasks, for a particular model of computation, can and cannot be accomplished within a finite length of time. If there is no algorithm, with respect to a particular model of computation, that can guarantee to find an answer to a given problem in a finite amount of time, that answer is said to be *uncomputable* with respect to that model of computation. One of the great breakthroughs in classical computer science was the recognition that all of the candidate models for computers, Turing machines, recursive functions, and λ -definable functions were equivalent in terms of what they could and could not compute. It is natural to wonder whether this equivalence extends to quantum computation too.

If you ask a young child what a computer can do you might be told, “They let me learn letters and numbers and play games.” Ask a teenager and you might hear, “They let me surf the Web and meet online in chat rooms with my friends.” Ask an adult and you might discover, “They’re great for email, word processing and keeping track of my finances.” What is remarkable is that the toddler, the teenager, the parent might all be talking about the *same* machine! By running the appropriate software it seems we can make the computer perform almost any task.

The possibility of one machine simulating another gave a theoretical justification for pursuing the idea of a programmable computer. In 1982, Richard Feynman observed that it did not appear possible for a Turing machine to simulate certain quan-

tum physical processes without incurring an exponential slowdown [181]. Here is an example.

Suppose you want to use a classical computer to simulate a quantum computer. Let's assume that the quantum computer is to contain n qubits and that each qubit is initially in a superposition state, $c_0|0\rangle + c_1|1\rangle$. Each such superposition is described by two complex numbers, c_0 and c_1 , so we need a total of $2n$ complex numbers to describe the initial state of all n qubits when they are in this product state form.

Now what happens if we want to simulate a joint operation on all n qubits? Well, you'll find that the cost of the simulation skyrockets. Once we perform a joint operation on all n qubits, i.e., once we evolve them under the action of some quantum algorithm, they will most likely become entangled with one another. Whereas the initial state that we started with could be factored into a product of a state for each qubit, an entangled state cannot be factored in this manner. In fact, to even write down an arbitrary entangled state of n qubits requires 2^n complex numbers. Thus, as a classical computer must keep track of all these complex numbers explicitly, the cost of a classical simulation of a quantum system requires a huge amount of memory and computer time.

What about a quantum computer? Could a quantum computer simulate any quantum system efficiently? There is a good chance that it could because the quantum computer would have access to exactly the same physical phenomena as the system it is simulating. This result poses something of a problem for traditional (classical) computer science.

4.3.1 Does Quantum Computability Offer Anything New?

Is it possible to make more pointed statements about computability and quantum computers?

The first work in this area appeared in David Deutsch's original paper on quantum Turing machines [136]. Deutsch argued that quantum computers could compute certain outputs, such as true random numbers, that are not computable by any deterministic Turing machine. Classical deterministic Turing machines can only compute functions, that is, mathematical procedures that return a single, reproducible, answer. However, there are certain computational tasks that cannot be performed by evaluating any function. For example, there is no *function* that generates a *true* random number. Consequently, a Turing machine can only feign the generation of random numbers.

In the same paper, Deutsch introduced the idea of quantum parallelism. Quantum parallelism refers to the process of evaluating a function once on a blend or "superposition" of all possible inputs to the function to produce a superposition of outputs. Thus all the outputs are computed in the time taken to evaluate just one output classically. Unfortunately, you cannot obtain all of these outputs explicitly because a measurement of the final superposed state would yield only one output. Nevertheless, it is possible to obtain certain joint properties of all of the outputs.

In 1991 Richard Jozsa gave a mathematical characterization of the class of functions (i.e., joint properties) that were computable by quantum parallelism [261]. He discovered that if f is some function that takes integer arguments in the range 1 to m and returns a binary value, and if the joint property function J that defines some collective attribute of all the outputs of f , takes m binary values and returns a single binary value, then only a fraction $(2^{2^m} - 2^{m+1})/(2^{2^m})$ of all possible joint property functions are computable by quantum parallelism.

Thus quantum parallelism alone is not going to be sufficient to solve all the joint property questions we might wish to ask. Of course, you could always make a QTM simulate a classical TM and compute a particular joint property in that way. Although this is feasible, it is not desirable, because the resulting computation would be no more efficient on the quantum computer than on the classical machine. However, the ability of a QTM to simulate a TM means that the class of functions computable on QTMs exactly matches the class of functions computable on classical TMs.

4.3.2 Decidability: Resolution of the *Entscheidungsproblem*

It was, you will recall, a particular question regarding computability that was the impetus behind the Turing machine idea. Hilbert's *Entscheidungsproblem* had asked whether there was a mechanical procedure for deciding the truth or falsity of any mathematical conjecture, and the Turing machine model was invented to prove that there was no such procedure.

To construct this proof, Turing used a technique called *reductio ad absurdum*, in which you begin by assuming the truth of the opposite of what you want to prove and then derive a logical contradiction. The fact that your one assumption coupled with purely logical reasoning leads to a contradiction proves that the assumption must be faulty. In this case the assumption is that there *is* a procedure for deciding the truth or falsity of any mathematical proposition and so showing that this leads to a contradiction allows you to infer that there *is*, in fact, no such procedure.

The proof goes as follows: if there *were* such a procedure, and it were truly mechanical, it could be executed by some Turing machine with an appropriate table of instructions. But a “table of instructions” could always be converted into some finite sequence of 1s and 0s. Consequently, such tables can be placed in an order, which meant that the things these tables represented (i.e., the Turing machines) could also be placed in an order.

Similarly, the statement of any mathematical proposition could also be converted into a finite sequence of 1s and 0s; so they too could be placed in an order. Hence Turing conceived of building a table whose vertical axis enumerated every possible Turing machine and whose horizontal axis, every possible input to a Turing machine.

But how would a machine convey its decision on the veracity of a particular input, that is, a particular mathematical proposition? You could simply have the machine

Table 4.1 Turing's Table. The i -th row is the sequence of outputs of the i -th Turing machine acting on inputs 0, 1, 2, 3, ...

i -th DTM	j -th Input							
	0	1	2	3	4	5	6	...
0	\otimes	\otimes	\otimes	\otimes	\otimes	\otimes	\otimes	...
1	0	0	0	0	0	0	0	...
2	1	2	1	\otimes	3	0	\otimes	...
3	2	0	0	1	5	7	\otimes	...
4	3	\otimes	1	8	1	6	9	...
5	7	1	\otimes	\otimes	5	0	0	...
6	\otimes	2	4	1	7	3	4	...
:	:	:	:	:	:	:	:	...

Table 4.2 Turing's Table after diagonal slash

i -th DTM	j -th Input							
	0	1	2	3	4	5	6	...
0	0	0	0	0	0	0	0	...
1	0	0	0	0	0	0	0	...
2	1	2	1	0	3	0	0	...
3	2	0	0	1	5	7	0	...
4	3	0	1	8	1	6	9	...
5	7	1	0	0	5	0	0	...
6	0	2	4	1	7	3	4	...
:	:	:	:	:	:	:	:	...

print out the result and halt. Hence the *Entscheidungsproblem* could be couched as the problem of deciding whether the i -th Turing machine acting on the j -th input would ever halt. Thus Hilbert's *Entscheidungsproblem* had been refashioned into Turing's Halting Problem.

Turing wanted to prove that there was no procedure by which the truth or falsity of a mathematical proposition could be decided; thus his proof begins by assuming the opposite, namely, that there *is* such a procedure. Under this assumption, Turing constructed a table whose (i, j) -th entry was the output of the i -th Turing machine on the j -th input, if and only if the machine halted on that input, or else some special symbol, such as \otimes , signifying that the corresponding Turing machine did not halt on that input. Such a table would resemble that shown in Table 4.1.

Next Turing replaced each symbol \otimes with the bit “0”. The result is shown in Table 4.2: Now because the rows enumerate all possible Turing machines and the columns enumerate all possible inputs (or, equivalently, mathematical propositions) all possible sequences of outputs, that is, all computable sequences of 1s and 0s,

Table 4.3 Turing's Table with 1 added to each element on the diagonal slash

<i>i</i> -th DTM	<i>j</i> -th Input							
	0	1	2	3	4	5	6	...
0	1	0	0	0	0	0	0	...
1	0	1	0	0	0	0	0	...
2	1	2	2	0	3	0	0	...
3	2	0	0	2	5	7	0	...
4	3	0	1	8	2	6	9	...
5	7	1	0	0	5	1	0	...
6	0	2	4	1	7	3	5	...
:	:	:	:	:	:	:	:	:

must be contained somewhere in this table. However, since any particular output is merely some sequence of 1s and 0s it is possible to change each one in some systematic way, for example by flipping one of the bits in the sequence. Consider incrementing each element on a diagonal slash through the table as shown in Table 4.3. The sequence of outputs along the diagonal differs in the *i*-th position from the sequence generated by the *i*-th Turing machine acting on the *i*-th input. Hence this sequence cannot appear in any of the rows in the table. However, by construction, the infinite table is supposed to contain *all* computable sequences and yet here is a sequence that we can clearly compute and yet cannot appear in any one row! Hence Turing established a contradiction and the assumption underpinning the argument must be wrong. That assumption was “there exists a procedure that can decide whether a given Turing machine acting on a given input will halt.” As Turing showed that the Halting problem was equivalent to the Entscheidungsproblem, the impossibility of determining whether a given Turing machine will halt before running it shows that the Entscheidungsproblem must be answered in the negative too. In other words, there is no procedure for deciding the truth or falsity of all mathematical conjectures.

4.3.3 Proof Versus Truth: Gödel's Incompleteness Theorem

In 1936 Kurt Gödel proved two important theorems that illustrated the limitations of formal systems. A formal system \mathcal{L} is called “consistent” if you can never prove both a proposition P and its negation $\neg P$ within the system. Gödel showed that “Any sufficiently strong formal system of arithmetic is incomplete if it is consistent.” In other words there are sentences P and $\neg P$ such that neither P nor $\neg P$ is provable using the rules of the formal system \mathcal{L} . As P and $\neg P$ express contradictory sentences, one of them must be true. So there must be true statements of the formal system \mathcal{L} that can never be proved. Hence Gödel showed that truth and theoremhood (or provability) are distinct concepts.

In a second theorem, Gödel showed that the simple consistency of \mathcal{L} cannot be proved in \mathcal{L} . Thus a formal system might be harboring deep-seated contradictions.

The results of Turing and Gödel are startling. They reveal that our commonsense intuitions regarding logical and mathematical theorem proving are not reliable. They are no less startling than the phenomena of entanglement, non-locality, etc in quantum physics.

In the 1980s some scientists began to think about the possible connections between physics and computability [320]. To do so, we must distinguish between Nature, which does what it does, and physics, which provides models of Nature expressed in mathematical form. The fact that physics is a mathematical science means that it is ultimately a formal system. Asher Peres and Wojciech Zurek have articulated three reasonable desiderata of a physical theory [390], namely, determinism, verifiability, and universality (i.e., the theory can describe anything). They conclude that:

“Although quantum theory is universal, it is not closed. Anything can be described by it, but something must remain unanalyzed. This may not be a flaw of quantum theory: It is likely to emerge as a logical necessity in any theory which is self-referential, as it attempts to describe its own means of verification.”

“In this sense it is analogous to Gödel’s undecidability theorem of formal number theory: the consistency of the system of axioms cannot be verified because there are mathematical statements which can neither be proved nor disproved by the use of the formal rules of the theory, although their truth may be verified by metamathematical reasoning.”

In a later paper Peres points out a “logico-physical paradox” [385]. He shows that it is possible to set up three quantum observables such that two of the observables have to obey the Heisenberg Uncertainty Principle. This Principle, says that certain pairs of observables, such as the position and momentum of a particle, cannot be measured simultaneously. Measuring one such observable necessarily disturbs the complementary observable, so you can never measure both observable together. Nevertheless, Peres arranges things so that he can use the rules of quantum mechanics to predict, with certainty, the value of both these variables individually. Hence we arrive at an example system that we can say things about but which we can never determine experimentally (a physical analogue of Gödel’s undecidability theorem).

4.3.4 Proving Versus Providing Proof

Many decades have now passed since Turing first dreamt of his machine and in fact today there are a number of programs around that actually perform as artificial mathematicians in exactly the sense Turing anticipated. Current interest in them stems not only from a wish to build machines that can perform mathematical reasoning but also more general kinds of logical inference such as medical diagnosis, dialog management, and even legal reasoning. Typically, these programs consist of three distinct components: a reservoir of knowledge about some topic (in the form

of axioms and rules of inference), an inference engine (which provides instructions on how to pick which rule to apply next), and a specific conjecture to be proved.

In one of the earliest examples, SHRDLU, a one-armed robot, was given a command in English which was converted into its logical equivalent and then used to create a program to orchestrate the motion of the robot arm [542]. So the robot gave the appearance of understanding a command in plain English simply by following rules for manipulating symbols. Nowadays such capabilities are commonplace. For example, many cell phones can understand a limited repertoire of verbal commands to dial telephone numbers, and some companies use automated query-answering systems to field routine customer enquiries.

In a more sophisticated example, the British Nationality Act was encoded in first-order logic and a theorem prover used to uncover logical inconsistencies in the legislation [447]. Similarly, the form of certain legal arguments can be represented in logic which can then be used to find precedents by revealing analogies between the current case and past examples. So although most people would think themselves far removed from the issue of “theorem proving,” they could be in for a surprise if the tax authorities decided to play these games with the tax laws!

Today’s artificial mathematicians are far less ingenious than their human counterparts. On the other hand, they are infinitely more patient and diligent. These qualities can sometimes allow artificial mathematicians to churn through proofs upon which no human would have dared embark. Take, for example, the case of map coloring. Cartographers conjectured that they could color any planar map with just four different colors so that no two adjacent regions had the same color. However, this conjecture resisted all attempts to construct a proof for many years. In 1976 the problem was finally solved with the help of an artificial mathematician. The “proof,” however, was somewhat unusual in that it ran to some 200 pages [541]. For a human to even check it, let alone generate it, would be a mammoth undertaking. Table 4.4 shows a summary of some notable milestones in mathematical proof by humans and machines.

Despite differences in the “naturalness” of the proofs they find, artificial mathematicians are nevertheless similar to real mathematicians in one important respect: their output is an explicit sequence of reasoning steps (i.e., a proof) that, if followed meticulously, would convince a skeptic that the information in the premises combined with the rules of logical inference would be sufficient to deduce the conclusion. Once such a chain were found the theorem would have been proved. The important point is that the proof chain is a tangible object that can be inspected at leisure. Surprisingly, this is not necessarily the case with a QTM. In principle, a QTM could be used to create some proof that relied upon quantum mechanical interference among all the computations going on in superposition. Upon interrogating the QTM for an answer you might be told, “Your conjecture is true,” but there would be no way to exhibit all the computations that had gone on in order to arrive at the conclusion. Thus, for a QTM, the ability to prove something and the ability to provide the proof trace are quite distinct concepts. Worse still, if you tried to peek inside the QTM as it was working, to glean some information about the state of the proof at that time, you would invariably disrupt the future course of the proof.

Table 4.4 Some impressive mathematical proofs created by humans and machines. In some cases simple proofs of long-standing mathematical conjectures have only recently been discovered. In other cases, the shortest known proofs are extremely long, and arguably too complex to be grasped by any single human

Mathematician	Proof feat	Notable features
Daniel Gorenstein	Classification of finite simple groups	Created by human. 15,000 pages long
Kenneth Appel and Wolfgang Haken	Proved the Four Color Theorem	Created by computer. Reduced all planar maps to combinations of 1,936 special cases and then exhaustively checked each case using ad hoc programs. Human mathematicians dislike this proof on the grounds that these ad hoc checking programs may contain bugs and the proof is too hard to verify by hand
Andrew Wiles	Proved Fermat's Last Theorem	Created by human. 200 pages long. Only 0.1% of all mathematicians are competent to judge its veracity
Laszlo Babai and colleagues	Invented probabilistic proof checking	Able to verify that a complex proof is “probably correct” by replicating any error in the proof in many places in the proof, thereby amplifying the chances of the error being detected
Thomas Hales	Proved Kepler’s conjecture on the densest way to pack spheres again using ad hoc programs to check a large number of test cases	In reaction to complaints by mathematicians, this proof is now being re-done using automated theorem provers instead of ad hoc checking programs since automated theorem provers, which have been tested extensively, have a higher assurance of being correct
Manindra Agrawal, Neeraj Kayal, and Nitin Saxena	On August 6, 2002 they proved primality testing can done deterministically in polynomial time	Created by humans. Took centuries to find this proof

4.4 Complexity

Complexity theory is concerned with how the inherent cost required to solve a computational problem scales up as larger instances of the problem are considered. It is possible to define many different resources by which the difficulty of performing a computation can be assessed. These include the time needed to perform the computation, the number of elementary steps, the amount of memory used, the number of calls to an oracle or black-box function, and the number of communicative acts. These lead to the notions of computational, query, and communication complexity. Specifically,

- *Computational complexity* measures the number of steps (which is proportional to time) or the minimum amount of memory required (which is proportional to space) needed to solve the problem.
- *Query complexity* measures the number of times a certain sub-routine must be called, or “queried”, in order to solve the problem.
- *Communication complexity* measures the volume of data that must be sent back and forth between parties collaborating to solve the problem.

Thus, whereas computability is concerned with which computational tasks computers can and cannot do, *complexity* is concerned with the efficiency with which they can do them. Efficiency is an important consideration for real-world computing. The fact that a computer can solve a particular kind of problem, in principle, does not guarantee that it can solve it in practice. If the running time of the computer is too long, or the memory requirements too great, then an apparently feasible computation can still lay beyond the reach of any practicable computer.

Computer scientists have developed a taxonomy for describing the complexity of various algorithms running on different kinds of computers. The most common measures of efficiency employ the rate of growth of the time or memory needed to solve a problem as the size of the problem increases. Of course “size” is an ambiguous term. Loosely speaking, the “size” of a problem is taken to be the number of bits needed to state the problem to the computer. For example, if an algorithm is being used to factor a large integer N , the “size” of the integer being factored would be roughly $\log_2 N$.

The traditional computational complexity distinction between tractable and intractable problems depends on whether the asymptotic scaling of the algorithm grows polynomially, i.e., $\mathcal{O}(n^k)$, or exponentially, i.e., $\mathcal{O}(k^n)$ with the problem size n .

These notions of tractability and intractability are somewhat imperfect because asymptotic scaling results are unattainable mathematical ideals in a finite Universe. Nor do they take into account the practically interesting range of sizes of problem instances. For example, airline scheduling is an **NP-Complete** problem. In the worst case, the time needed to find the optimal schedule scales exponentially in the number of aircraft to be scheduled. But the number of jetliners with which we are ever likely to have to deal, in practice, is bounded. So if someone invented a scheduling algorithm that scaled as $O(n^{100})$ (where n is the number of jetliners) then, even

though it is polynomial it might not be practically better than an exponential time scheduling algorithm for realistic problems.

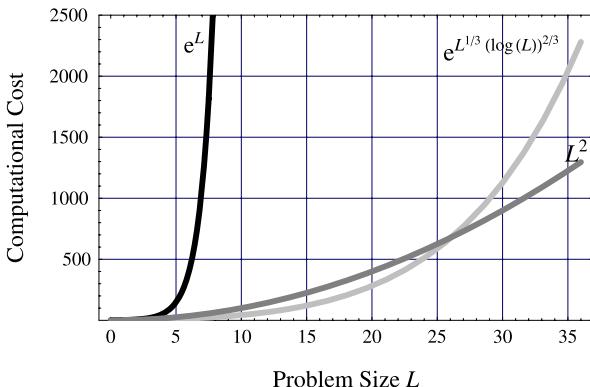
The reason complexity classifications are based on the rates of growth of running times and memory requirements, rather than absolute running times and memory requirements, is to factor out the variations in performance experienced by different makes of computers with different amounts of RAM, swap space, and processor speeds. Using a growth rate-based classification, the complexity of a particular algorithm becomes an intrinsic measure of the difficulty of the *problem* the algorithm addresses.

Although complexity measures are independent of the precise make and configuration of computer, they are related to a particular mathematical model of the computer such as a deterministic Turing machine or a probabilistic Turing machine. It is now known, for example, that many problems that are intractable with respect to a deterministic Turing machine can be solved efficiently, or at least can sometimes have their solutions approximated efficiently, with high probability on a probabilistic Turing machine. The Euclidean Traveling Salesman Problem (Euclidean-TSP), e.g., consists of finding a path having minimum Euclidean distance between a set of points in a plane such that the path visits each point exactly once before returning to its starting point. Euclidean-TSP is known to be **NP-Complete** [377], and therefore rapidly becomes intractable as the number of points to be visited, $N \rightarrow \infty$. Nevertheless, in [20], Arora exhibits a randomized algorithm that can find a tour to within a factor of $\mathcal{O}(1 + 1/c)$ of the optimal tour (for any constant c) in a time that scales only as $\mathcal{O}(N(\log(N))^{\mathcal{O}(c)})$. This is worse than linear scaling but much better than exponential scaling. Other examples include random walk algorithms for approximating the permanent of a matrix with non-zero entries [255], finding satisfying assignments to a Boolean expression (k -SAT with $k > 2$) [439], estimating the volume of a convex body [162], and estimating graph connectivity [364]. Classical random walks also underpin many standard methods in computational physics, such as Monte Carlo simulations. Thus, randomization can be a powerful tool for rendering intractable problems tractable provided we are content with finding a good approximation to a global optimum or exact solution.

There are many criteria by which you could assess how efficiently a given algorithm solves a given type of problem. For the better part of the century, computer scientists focused on worst-case complexity analyses. These have the advantage that, if you can find an efficient algorithm for solving some problem, in the *worst* case, then you can be sure that you have an efficient algorithm for any instance of such a type of problem.

Worst case analyses can be somewhat misleading however. Recently some computer scientists have developed average case complexity analyses. Moreover, it is possible to understand the finer grain structure of complexity classes and locate regions of especially hard and especially easy problems within a supposedly “hard” class [101, 537, 539]. Nevertheless, one of the key questions is whether some algorithm runs in polynomial time or exponential time.

Fig. 4.4 A comparison of polynomial versus exponential growth rates. Exponential growth will always exceed polynomial growth eventually, regardless of the order of the polynomial



4.4.1 Polynomial Versus Exponential Growth

Computer scientists have developed a rigorous way of quantifying the difficulty of a given type of problem. The classification is based on the mathematical form of the function that describes how the computational cost incurred in solving the problem scales up as larger problems are considered. The most important quantitative distinction is between polynomially growing costs (which are deemed tractable) and exponentially growing costs (which are deemed intractable). Exponential growth will always exceed polynomial growth eventually, regardless of the order of the polynomial. For example, Fig. 4.4 compares the growth of the exponential function $\exp(L)$ with the growth of the polynomials L^2 , L^3 and L^4 . As you can see, eventually, whatever the degree of the polynomial in L , the exponential becomes larger.

A good pair of example problems that illustrate the radical difference between polynomial and exponential growth are multiplication versus factoring. It is relatively easy to multiply two large numbers together to obtain their product, but it is extremely difficult to do the opposite; namely, to find the factors of a composite number:

$$1459 \times 83873 \rightarrow 122370707 \text{ (easy)} \quad (4.1)$$

$$122370707 \rightarrow 1459 \times 83873 \text{ (hard)} \quad (4.2)$$

If, in binary notation, the numbers being multiplied have L bits, then multiplication can be done in a time proportional to L^2 , a polynomial in L .

For factoring, the best known classical algorithms are the Multiple Polynomial Quadratic Sieve [460] for numbers involving roughly 100 to 150 decimal digits, and the Number Field Sieve [309] for numbers involving more than roughly 110 decimal digits. The running time of these algorithms grows subexponentially (but superpolynomially) in L , the number of bits needed to specify the number to be factored N . The best factoring algorithms require a time of the order $\mathcal{O}(\exp(L^{1/3}(\log L)^{2/3}))$ which grows subexponentially (but superpolynomially) in L , the number of bits needed to specify the number being factored.

Table 4.5 Progress in factoring large composite integers. One MIP-Year is the computational effort of a machine running at one million instructions per second for one year

Number	Number of decimal digits	First factored	MIPS years
Typical	20	1964	0.001
Typical	45	1974	0.01
Typical	71	1984	0.1
RSA-100	100	1991	
RSA-110	110	1992	
RSA-120	120	1993	825
RSA-129	129	1994	5000
RSA-130	130	1996	750
RSA-140	140	1999	2000
RSA-150	150	2004	
RSA-155	155	1999	8000

Richard Crandall charted the progress in factoring feats from the 1970s to the 1990s [118]. In Table 4.5 we extend his data to more modern times. In the early 1960s computers and algorithms were only good enough to factor numbers with 20 decimal digits, but by 1999 that number had risen to a 155 decimal digit numbers, but only after a Herculean effort. Many of the numbers used in these tests were issued as grand challenge factoring problems by RSA Data Securities, Inc., and hence bear their name. Curiously, RSA-155 was factored prior to RSA-150 (a smaller number). The most famous of these factoring challenge problems is RSA-129.

As we show later in the book, the presumed difficulty of factoring large integers is the basis for the security of so-called public key cryptosystems that are in widespread use today. When one of these systems was invented the authors laid down a challenge prize for anyone who could factor the following 129 digit number (called RSA-129) :

$$\begin{aligned}
 \text{RSA-129} = & 114381625757888676692357799761466120102182 \\
 & \dots 9672124236256256184293570693524573389783059 \\
 & \dots 7123563958705058989075147599290026879543541 \quad (4.3)
 \end{aligned}$$

But in 1994 a team of computer scientists using a network of workstations succeeding in factoring $\text{RSA-129} = p \times q$ where the factors p are q are given by:

$$\begin{aligned}
 p = & 34905295108476509491478496199038981334177646384933878 \\
 & \dots 43990820577 \\
 q = & 32769132993266709549961988190834461413177642967992942 \\
 & \dots 539798288533 \quad (4.4)
 \end{aligned}$$

Extrapolating the observed trend in factoring suggests that it would take millions of MIP-Years to factor a 200-digit number using conventional computer hardware.

However, it might be possible to do much better than this using special purposes factoring engines as we discuss in Chap. 13.

Although, the traditional computational complexity distinction between tractable and intractable problems depends on whether the asymptotic scaling of the algorithm grows polynomially, i.e., $O(n^k)$, or exponentially, i.e., $O(k^n)$ with the problem size n , strictly speaking, this distinction is imperfect since it does not take into account the finiteness of the Universe. Asymptotic results are unattainable mathematical ideals in a finite Universe. Nor do they take into account the practically interesting range of sizes of problem instances. For example, airline scheduling is an **NP-Complete** problem. In the worst case, the time needed to find the optimal schedule scales exponentially in the number of aircraft to be scheduled. But the number of jetliners with which we are ever likely to have to deal, in practice, is bounded. So if someone invented a scheduling algorithm that scaled as $O(n^{100})$ (where n is the number of jetliners) then, even though it is polynomial it might not be practically better than an exponential time scheduling algorithm for realistic problems.

4.4.2 Big \mathcal{O} , Θ and Ω Notation

Complexity theory involves making precise statements about the scaling behavior of algorithms in the asymptotic limit. This is usually described by comparing the growth rate of the algorithm to that of a simple mathematical function in the limit that the size of the computational problem goes to infinity. The most common asymptotic scaling relationships, together with their standard notations, are summarized in Table 4.6.

For example, consider the three functions $f(x) = \sqrt{\frac{x}{2}}$, $g(x) = \frac{3}{x} \sin x + \log x$, and $h(x) = \log \frac{3x}{4}$. Their graphs are shown in Fig. 4.5. For small values of x , $g(x)$ can be greater than or less than $f(x)$, and likewise greater than or less than $h(x)$. However, asymptotically, i.e., “eventually”, $g(x)$ is bounded above by $f(x)$ and therefore $g(x) = \mathcal{O}(f(x))$. Similarly, asymptotically, $g(x)$ is bounded below by $h(x)$ and so $g(x) = \Omega(h(x))$. However, as the limit $\lim_{x \rightarrow \infty} |\frac{\frac{3}{x} \sin x + \log x}{\log \frac{3x}{4}} - 1| = 0$, we also have $g(x)$ equals $h(x)$ asymptotically, i.e., $g(x) \sim h(x)$ asymptotically.

We can use the aforementioned notation to characterize the asymptotic behaviors of some well-known algorithms. Table 4.7 shows the asymptotic running times of some famous algorithms.

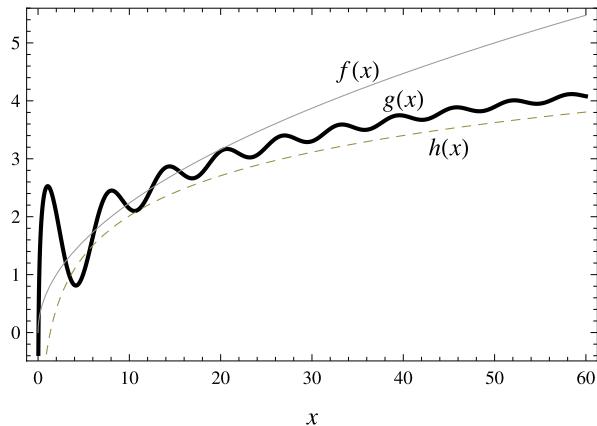
4.4.3 Classical Complexity Zoo

Knowing the exact functional forms for the rates of growth of the number of computational steps for various algorithms allows computer scientists to classify computational problems based on difficulty. The most useful distinctions are based on

Table 4.6 Notation used to characterize the asymptotic scaling behavior of algorithms

Notation	Meaning	Formal definition
$f(x) = \mathcal{O}(g(x))$	$f(x)$ is bounded above by $g(x)$ asymptotically	As $x \rightarrow \infty$, $\exists k$ s.t. $ f(x) \leq kg(x)$
$f(x) = o(g(x))$	$f(x)$ is dominated by $g(x)$ asymptotically	As $x \rightarrow \infty$, $\forall k$ s.t. $ f(x) \leq kg(x)$
$f(x) = \Omega(g(x))$	$f(x)$ is bounded below by $g(x)$ asymptotically	As $x \rightarrow \infty$, $\exists k$ s.t. $ f(x) \geq kg(x)$
$f(x) = \omega(g(x))$	$f(x)$ dominates $g(x)$ asymptotically	As $x \rightarrow \infty$, $\forall k$ s.t. $ f(x) \geq kg(x)$
$f(x) = \Theta(g(x))$	$f(x)$ is bounded above and below by $g(x)$ asymptotically	As $x \rightarrow \infty$, $\exists k_1, k_2$ s.t. $k_1 g(x) \leq f(x) \leq k_2 g(x)$
$f(x) \sim g(x)$	$f(x)$ equals $g(x)$ asymptotically	As $n \rightarrow \infty$, $\forall k$ s.t. $ f(x)/g(x) - 1 \leq k$

Fig. 4.5 Graphs of $f(x) = \sqrt{\frac{x}{2}}$, $g(x) = \frac{3}{x} \sin x + \log x$, and $h(x) = \log \frac{3x}{4}$. As x becomes larger the relative dominance of the functions becomes clear



classes of problems that either can or cannot be solved in polynomial time, in the worst case. Problems that can be solved in polynomial time are usually deemed “tractable” and are lumped together into the class P. Problems that cannot be solved in polynomial time are usually deemed “intractable” and may be in one of several classes. Of course it is possible that the order of the polynomial is large making a supposedly “tractable” problem rather difficult in practice. Fortunately, such large polynomial growth rates do not arise that often, and the polynomial/exponential distinction is a pretty good indicator of difficulty. In Table 4.8 we list some classical complexity classes.

The known inclusion relationships between the more important of these complexity classes are shown in Fig. 4.6.

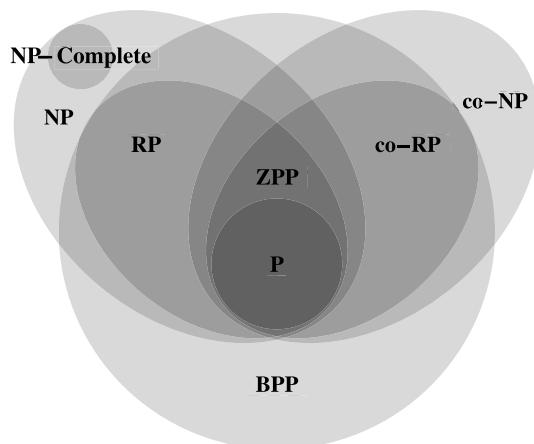
Table 4.7 The asymptotic scaling behavior of some important algorithms

Algorithm	Description	Classical	Quantum	Source
UNSTRUCTURED SEARCH	Given a black box function $f(x)$ that returns 1 iff $x = t$ and 0 otherwise how many calls to $f(x)$ are needed to find the index t ?	$\Omega(N)$	$\mathcal{O}(\sqrt{N})$	See [219]
FACTORING INTEGERS	Given an integer N find factors p and q such that $N = pq$	With $n = \log N$, $\mathcal{O}(e^{n^{1/3}} (\log n)^{2/3})$	$\mathcal{O}((\log N)^3)$	See [458]
VERIFYING MATRIX PRODUCT	Given matrices A , B , and C , verify $A \cdot B = C$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^{5/3})$	See [87]
MINIMUM SPANNING TREE OF WEIGHTED GRAPH	Given an oracle that has knowledge of the adjacency matrix of a graph, G , how many calls to the oracle are required to find a minimum spanning tree?	$\Omega(n^2)$	$\Theta(n^{2/3})$	See [161]
DECIDING GRAPH CONNECTIVITY	Given an oracle that has knowledge of the adjacency matrix of a graph, G , how many calls to the oracle are required to decide if the graph is connected?	$\Omega(n^2)$	$\Theta(n^{2/3})$	See [161]
FINDING LOWEST WEIGHT PATHS	Given an oracle that has knowledge of the adjacency matrix of a graph, G , how many calls to the oracle are required to find a lowest weight path?	$\Omega(n^2)$	$\mathcal{O}(n^{2/3} (\log n)^2)$	See [161]
DECIDING BIPARTITENESS	Given an oracle that has knowledge of the adjacency matrix of a graph, G , how many calls to the oracle are required to decide if the graph is bipartite?	$\Omega(n^2)$	$\mathcal{O}(n^{3/2})$	See [59]

Table 4.8 Some classical complexity classes and example problems within those classes

Classical complexity class	Intuitive meaning	Examples
P or PTIME	Polynomial-Time: the running time of the algorithm is, in the worst case, a polynomial in the size of the input. All problems in P are tractable	Multiplication, linear programming [276], and primality testing (a relatively new addition to this class) [5, 6]. Computing the determinant of a matrix. Deciding if a graph <i>has</i> a perfect matching
ZPP	Zero-Error Probabilistic Polynomial-Time: Can be solved, with certainty, by PTMs in average case polynomial time	Randomized Quicksort
BPP	Bounded-Error Probabilistic Polynomial Time: Decisions problems solvable in polynomial time by PTMs with probability $> 2/3$. Probability of success can be made arbitrarily close to 1 by iterating the algorithm a certain number of times	Decision version of Min-Cut [198]
NP	Nondeterministic Polynomial time: The class of decision problems with the property that if you could magically “guess” a correct solution you could verify this fact in polynomial time	Factoring composite integers: a purported solution can be verified by multiplying the claimed factors and comparing the result to the number being factored. At the present time it is unknown whether or not P = NP but it appears unlikely
NP-Complete	Subset of problems in NP that can be mapped into one another in polynomial time. If just one of the problems in this class is shown to be tractable, then they must all be tractable. Not all problems in NP are NP-Complete	Examples include Scheduling, Satisfiability, Traveling Salesman Problem, 3-Coloring, Subset-Sum, Hamiltonian Cycle, Maximum Clique [115]
NP-Hard	The optimization version of NP-Complete problems, wherein one not only wants to decide if a solution exists but to actually one	Determining the solutions to a SAT problem
#P	Counting version of an NP-Hard problem	Determining the number of satisfying assignments to a SAT problem [507]
#P-Complete	Sharp P Complete	Computing the permanent of an $n \times n$ 0-1 matrix $\{a_{ij}\}$, i.e., $\sum_{\sigma} \prod_{i=1}^n a_{i,\sigma(i)}$ where σ ranges over all permutations of $1, 2, 3, \dots, n$. The number of perfect matchings in a graph

Fig. 4.6 Some known inclusion relationships between classical complexity classes. The most important classes shown are **P**—class of problems that can be solved in polynomial time, and **NP**—the class of problems whose solution can be verified in polynomial time. Of these a special subset—the **NP-Complete** problems—are at least as hard as any other problem in **NP**



4.4.4 Quantum Complexity Zoo

The introduction of quantum considerations turns out to have profound implications for the foundations of computer science and information theory. Decades of old theory must now be taken from the library shelves, dusted off and checked for an implicit reliance upon classical bits and classical physics. By exploiting entirely new kinds of physical phenomena, such as superposition, interference, entanglement, non-determinism and non-clonability, we can suddenly catch a glimpse of a new theoretical landscape before us. This shift from classical to quantum is a qualitative change not merely a quantitative change such as the trends we saw in Chap. 1. It is something entirely new.

Just as there are classical complexity classes, so too are there quantum complexity classes (see Fig. 4.7). As quantum Turing machines are quantum mechanical generalizations of probabilistic Turing machines, the quantum complexity classes resemble the probabilistic complexity classes. There is a tradeoff between the certainty of your answer being correct versus the certainty of the answer being available within a certain time bound. In particular, the classical classes **P**, **ZPP**, and **BPP** become the quantum classes **QP**, **ZQP**, and **BQP**. These mean, respectively, that a problem can be solved with certainty in *worst-case* polynomial time, with certainty in *average-case* polynomial time, and with *probability greater than 2/3* in worst-case polynomial time, by a quantum Turing machine.

Statements about the relative power of one type of computer over another can be couched in the form of subset relationships among complexity classes. Thus **QP** is the class of problems that can be solved, with certainty, in polynomial time, on a quantum computer, and **P** is the set of problems that can be solved, with certainty, in polynomial time on a classical computer. As the class **QP** contains the class **P** (see Table 4.9) this means that there are more problems that can be solved efficiently by a quantum computer than by any classical computer. Similar relationships are now known for some of the other complexity classes too, but there are still many open questions remaining.

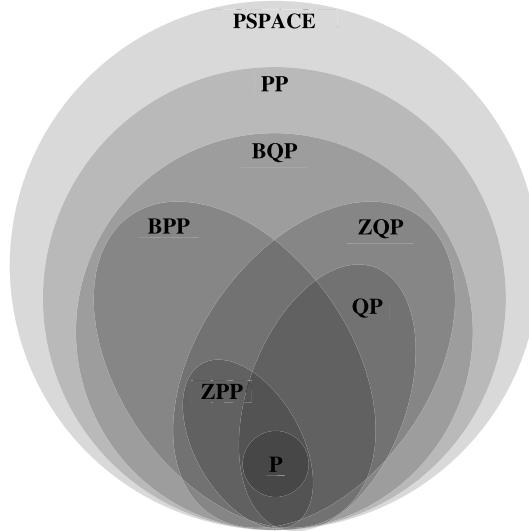


Fig. 4.7 Some known inclusion relationships between classical and quantum complexity classes. Classes correspond to circular and oval shapes and containment is shown by shape inclusion. The most important classes shown are **QP**—the class of problems that can be solved with certainty by a quantum computer in *worst-case* polynomial time; **ZQP**—the class of problems that can be solved with certainty by a quantum computer in *average-case* polynomial time; and **BQP**—the class of problems that can be solved with *probability greater than 2/3* by a quantum computer in *worst-case* polynomial time

The study of quantum complexity classes began with David Deutsch in his original paper on quantum Turing machines (QTMs). The development of the field is summarized in Table 4.10.

In Deutsch's original paper he presented the idea of quantum parallelism. Quantum parallelism allows you to compute an exponential number of function evaluations in the time it takes to do just one function evaluation classically. Unfortunately, the laws of quantum mechanics make it impossible to extract more than one of these answers explicitly. The problem is that although you can indeed calculate all the function values for all possible inputs at once, when you read off the final answer from the tape, you will only obtain one of the many outputs. Worse still, in the process, the information about all the other outputs is lost irretrievably. So the net effect is that you are no better off than had you used a classical Turing machine. So, as far as function evaluation goes, the quantum computer is no better than a classical computer.

Deutsch realized that you could calculate certain joint properties of all of the answers without having to reveal any one answer explicitly. (We explained how this works in Chap. 1). The example Deutsch gave concerned computing the XOR (exclusive-or) of two outputs. Suppose there is a function f that can receive one of two inputs, 0 or 1, and that we are interested in computing the XOR of both function values, i.e., $f(0) \oplus f(1)$ (where \oplus here means “XOR”). The result could,

Table 4.9 Some quantum complexity classes and their relationships to classical complexity classes

Quantum class	Class of computational problems that can...	Relationship to classical complexity classes (if known)
QP	Quantum Polynomial-Time: ... be solved, with certainty, in worst-case polynomial time by a quantum computer. All problems in QP are tractable	P ⊂ QP (The quantum computer can solve more problems in worst case polynomial time than the classical computer)
ZQP	Zero-Error Quantum Polynomial-Time: ... can be solved, with zero error probability, in expected polynomial time by a quantum computer	ZPP ⊂ ZQP
BQP	Bounded-Error Quantum Polynomial Time: ... be solved in worst-case polynomial time by a quantum computer with probability $> \frac{2}{3}$ (thus the probability of error is bounded; hence the B in BQP)	BPP ⊆ BQP ⊆ PSPACE (i.e., the possibility of the equality means it is not known whether QTMs are more powerful than PTMs.) BQP is the class of problems that are easy for a quantum computer, e.g., factoring composite integers, computing discrete logarithms, sampling from a Fourier transform, estimating eigenvalues, and solving Pell's equation [225, 458]

for example, be a decision as to whether to make some stock investment tomorrow based on today's closing prices. Now suppose that, classically, it takes 24 hours to evaluate each f . Thus if we are stuck with a single classical computer, we would never be able to compute the XOR operation in time to make the investment the next day. On the other hand, using quantum parallelism, Deutsch showed that half the time we would get no answer at all, and half the time we would get the guaranteed correct value of $f(0) \oplus f(1)$. Thus the quantum computer would give useful advice half the time and never give wrong advice.

Richard Jozsa refined Deutsch's ideas about quantum parallelism by showing that many functions—for example, SAT (the propositional satisfiability problem)—cannot be computed by quantum parallelism at all [261]. Nevertheless, the question about the utility of quantum parallelism for tackling computational tasks that were not function calculations remained open.

In 1992 Deutsch and Jozsa exhibited a problem, that was not equivalent to a function evaluation, for which a quantum Turning machine (QTM) was exponentially faster than a classical deterministic Turing Machine (DTM). The problem was rather contrived, and consisted of finding a true statement in a list of two statements. It was possible that both statements were true, in which case either statement would be acceptable as the answer. This potential multiplicity of solutions meant that the problem could not be reformulated as a function evaluation. The upshot was that the QTM could solve the problem in a “polynomial in the logarithm of the prob-

Table 4.10 Historical development of quantum complexity theory

Year	Advance in quantum complexity theory
Benioff (1980)	Shows how to use quantum mechanics to implement a Turing Machine (TM)
Feynman (1982)	Shows that TMs cannot simulate quantum mechanics without exponential slowdown
Deutsch (1985)	Proposes first universal QTM and the method of quantum parallelism. Proves that QTMs are in the same complexity with respect to function evaluation as TMs. Remarks that some computational tasks (e.g., random number generation) do not require function evaluation. Exhibits a contrived decision problem that can be solved faster on a QTM than on a TM
Jozsa (1991)	Describes classes of functions that can and cannot be computed efficiently by quantum parallelism
Deutsch & Jozsa (1992)	Exhibit a contrived problem that the QTM solves with certainty in poly-log time, but that requires linear time on a DTM. Thus, the QTM is exponentially faster than the DTM. Unfortunately, the problem is also easy for a PTM so this is not a complete victory over classical machines
Berthiaume & Brassard (1992)	Prove $P \subset QP$ (strict inclusion). The first definitive complexity separation between classical and quantum computers
Bernstein & Vazirani (1993)	Describe a universal QTM that can simulate any other QTM efficiently (Deutsch's QTM could simulate other QTMs, but only with an exponential slowdown)
Yao (1993)	Shows that complexity theory for quantum circuits matches that of QTMs. This legitimizes the study of quantum circuits (which are simpler to design and analyze than QTMs)
Berthiaume & Brassard (1994)	Prove that randomness alone is not what gives QTMs the edge over TMs. Prove that there is a decision problem that is solved in polynomial time by a QTM, but requires exponential time, in the worst case, on a DTM and PTM. First time anyone showed a QTM to beat a PTM. Prove there is a decision problem that is solved in exponential time on a QTM but which requires double exponential times on a DTM on all but a few instances
Simon (1994)	Lays foundational work for Shor's algorithm
Shor (1994)	Discovers a polynomial-time quantum algorithm for factoring large composite integers. This is the first <i>significant</i> problem for which a quantum computer is shown to outperform any type of classical computer. Factoring is related to breaking codes in widespread use today
Grover (1996)	Discovers a quantum algorithm for finding a single item in an unsorted database in square root of the time it would take on a classical computer. if the search takes N steps classically, it takes $(\pi/4)\sqrt{N}$ quantum-mechanically

lem size” time (poly-log time), but that the DTM required linear time. Thus the QTM was exponentially faster than the DTM. The result was only a partial success,

however, as a probabilistic Turing machine (PTM) could solve it as efficiently as could the QTM. But this did show that a quantum computer at least could beat a deterministic classical computer.

So now the race was on to find a problem for which the QTM beat a DTM and a PTM. Ethan Bernstein and Umesh Vazirani analyzed the computational power of a QTM and found a problem that did beat both a DTM and a PTM [57]. Given any Boolean function on n -bits Bernstein and Vazirani showed how to sample from the Fourier spectrum of the function in polynomial time on a QTM. It was not known if this were possible on a PTM. This was the first result that hinted that QTMs might be more powerful than PTMs.

The superiority of the QTM was finally clinched by André Berthiaume and Gilles Brassard who constructed an “oracle” relative to which there was a decision problem that could be solved with certainty in worst-case polynomial time on the quantum computer, yet cannot be solved classically in probabilistic expected polynomial time (if errors are not tolerated). Moreover, they also showed that there is a decision problem that can be solved in exponential time on the quantum computer, that requires double exponential time on all but finitely many instances on any classical deterministic computer. This result was proof that a quantum computer could beat both a deterministic and probabilistic classical computer but it was still not headline news because the problems for which the quantum computer was better were all rather contrived.

The situation changed when, in 1994, Peter Shor, building on work by Dan Simon, devised polynomial-time algorithms for factoring composite integers and computing discrete logarithms. The latter two problems are believed to be intractable for any classical computer, deterministic or probabilistic. But more important, the factoring problem is intimately connected with the ability to break the RSA cryptosystem that is in widespread use today. Thus if a quantum computer could break RSA, then a great deal of sensitive information would suddenly become vulnerable, at least in principle. Whether it is vulnerable in practice depends, of course, on the feasibility of designs for actual quantum computers.

4.5 What Are Possible “Killer-Aps” for Quantum Computers?

The discovery of Shor’s and Grover’s algorithms led many people to expect other quantum algorithms would quickly be found. However, this was not the case. It turns out to be quite hard to find new quantum algorithms. So where exactly should we be looking? Currently, there are two broad classes of quantum algorithms. There are those, such as Shor’s algorithm, that exhibit *exponential* improvements over what is possible classically and those, such as Grover’s algorithm, that exhibit *polynomial* speedups over what is possible classically. Shor’s algorithm is arguably the more interesting case since exponential speedups are game-changing. It is natural to wonder whether the other computational problems that lie in the same complexity class as the problems tackled by Shor’s algorithm might be amenable to a similar speedup.

The most likely candidate opportunities are therefore computational problems (like factoring and discrete log) that are believed to be in the **NP-Intermediate** class. These are problems that are certainly in **NP** but neither in **P** nor **NP-Complete**. Some examples of presumed **NP-Intermediate** problems collected by Miklos Santha are as follows [429]:

GRAPH-ISOMORPHISM Given two graphs $G_1 = (V, E_1)$, and $G_2 = (V, E_2)$, is there a mapping between vertices, $f : V \rightarrow V$, such that $\{u, v\} \in E_1 \Leftrightarrow \{f(u), f(v)\} \in E_2$?

HIDDEN-SUBGROUP Let G be a finite group, and let $\gamma : G \rightarrow X$ (X a finite set), such that γ is constant and distinct on cosets of a subgroup H of G . Find a generating set for H .

PIGEONHOLE SUBSET-SUM Given a set of positive integers $s_1, s_2, \dots, s_n \in \mathbb{N}$ such that $\sum_{i=1}^n s_i < 2^n$, are there two subsets of indices, $I_1 \neq I_2 \subseteq \{1, 2, \dots, n\}$ that sum to the same value, i.e., $\sum_{i \in I_1} s_i = \sum_{j \in I_2} s_j$?

With sufficient research, it is conceivable any of the **NP-Intermediate** problems might be re-classified at some point. Nevertheless, today, the **NP-Intermediate** problems are the best prospects for being amenable to an exponential speedup using some as-yet-to-be-discovered quantum algorithm. So far, exponentially faster quantum algorithms have been found for solving the Hidden Subgroup Problem over abelian groups [72, 283, 362, 458] and some non-abelian groups [30, 192]. However, extending these results to other non-abelian groups has proven to be challenging and only limited progress has been made [324]. Researchers are especially interested in extending these results to two families of non-abelian groups—permutation groups and dihedral groups—because doing so will lead immediately to efficient solutions for GRAPH ISOMORPHISM [262] and the SHORTEST LATTICE VECTOR problems [416], which would make quantum computing considerably more interesting.

While progress is therefore being made the exact boundary where quantum algorithms can be found that outperform classical counterparts by an exponential factor is still ill-defined.

4.6 Summary

The most important concept of this chapter is the idea that, as computers are physical objects, their capabilities are constrained exclusively by the laws of *physics* and not pure mathematics. Yet the current (classical) theory of computation had several independent roots, all based on *mathematical* idealizations of the computational process. The fact that these mathematically idealized models turned out to be equivalent to one another led most classical computer scientists to believe that the key elements of computation had been captured correctly, and that it was largely a

matter of taste as to which model of computation to use when assessing the limits of computation.

However, it turns out that the classical models of computation all harbor implicit assumptions about the physical phenomena available to the computer. As Feynman and Deutsch pointed out, models of computation that allow for the exploitation of *quantum* physical effects are qualitatively different from, and potentially more powerful than, those that do not. Which quantum effects really matter the most is still not entirely understood, but the phenomenon of entanglement appears to play a significant role.

In this chapter we surveyed issues of complexity, computability, and universality in the quantum and classical domains. Although there is no *function* a quantum computer can compute that a classical computer cannot also compute, given enough time and memory, there are computational *tasks*, such as generating true random numbers and teleporting information, that quantum computers can do but which classical ones cannot.

A question of some practical importance is to determine the class of computational problems that quantum computers can solve faster than classical ones. To this end, quantum computer scientists have determined the scaling of the “cost” (in terms of space, time, or communications) of certain quantum algorithms (such as factoring integers, and unstructured search, in comparison to that of their best classical counterparts. Some quantum algorithms, such as Shor’s algorithm for factoring composite integers and computing discrete logarithms, Hallgren’s algorithm for solving Pell’s equation, and eigenvalue estimation, show *exponential* speedups, whereas others, such as Grover’s algorithm for unstructured search, show only *polynomial* speedups [55]. The greatest challenge to quantum computer scientists is to systematically expand the repertoire of problems exhibiting exponential speedups. Good candidates for problems that might admit such speedups are the other problems in the same complexity class as FACTORING and DISCRETE-LOG, i.e., **NP-Intermediate**. However, to date, no one has succeeded in showing exponential speedups on these other **NP-Intermediate** problems in their most general form. Other problems admit only a polynomial speedup (e.g., SEARCHING-A-SORTED-LIST) or no speedup whatsoever (e.g., PARITY). So far, no quantum algorithm has been found that can speedup the solution of an **NP-Complete** or **NP-Hard** problem by an exponential factor, and most quantum computer scientists are highly skeptical any such algorithm exists.

4.7 Exercises

4.1 Stirling’s approximation for the factorial function is $n! = \Theta(\sqrt{2\pi n}(\frac{n}{e})^n)$ (for integer values of n). Does this mean that $n!$ grows at a faster, slower, or equal rate to $\sqrt{2\pi n}(\frac{n}{e})^n$? Plot a graph of the ratio of the left and right hand sides of Stirling’s formula for $n = 1, 2, \dots, 20$. How does the percentage error in the approximation change with increasing values of n ?

4.2 Prove, using non-numeric methods,

- (a) The base of natural logarithms, e , and π satisfy $e^\pi > \pi^e$
- (b) The golden ratio $\phi = (1 + \sqrt{5})/2$ is less than $\pi^2/6$. [Hint: $\frac{\pi^2}{6} = \sum_{n=1}^{\infty} \frac{1}{n^2}$]

4.3 Classify the following particular claims involving $\mathcal{O}(\cdot)$ notation as correct or incorrect, and if incorrect, give a corrected version:

- (a) $\mathcal{O}(n^3 + n^5) = \mathcal{O}(n^3) + \mathcal{O}(n^5)$
- (b) $\mathcal{O}(n^2 \times \log n) = \mathcal{O}(n^2) \times \mathcal{O}(\log n)$
- (c) $0.0001n^3 + 1000n^{2.99} + 17 = \mathcal{O}(n^3)$
- (d) $4n^4 + 3n^{3.2} + 13n^{2.1} = \mathcal{O}(n^{7.2})$
- (e) $\log n^{10} = \mathcal{O}(\log n)$
- (f) $(\log n)^{10} = \mathcal{O}(n^{2.1})$
- (g) $3\log_{10} n^2 + 10\log_2 \log_2 n^{10} = \mathcal{O}(\log_e n)$

4.4 Classify the following generic claims regarding $\mathcal{O}(\cdot)$ notation as correct or incorrect, and if incorrect, give a corrected version:

- (a) If $f(n) = \mathcal{O}(g(n))$ then $kf(n) = O(g(n))$ for any k
- (b) If $f(n) = \mathcal{O}(g(n))$ and $h(n) = \mathcal{O}(g'(n))$ then $f(n) + h(n) = \mathcal{O}(g(n) + g'(n))$
- (c) If $f(n) = \mathcal{O}(g(n))$ and $h(n) = \mathcal{O}(g'(n))$ then $f(n)h(n) = \mathcal{O}(g(n)g'(n))$
- (d) If $f(n) = \mathcal{O}(g(n))$ and $g(n) = O(h(n))$ then $f(n) = \mathcal{O}(h(n))$
- (e) If $f(n)$ is a polynomial of degree d , then $f(n) = O(n^d)$
- (f) If $\log n^k = \mathcal{O}(\log n)$ for $k > 0$
- (g) If $(\log n)^k = \mathcal{O}(n^j)$ for $k > 0$ and $j > 0$

4.5 What can be said about the expression $3n^4 + 5n^{2.5} + 14\log n^{1.2}$ in terms of

- (a) $\mathcal{O}(\cdot)$ notation
- (b) $\Theta(\cdot)$ notation
- (c) $\Omega(\cdot)$ notation

4.6 Complexity analyses often involve summing series over finitely many terms. Evaluate the following sums in closed form:

- (a) $\sum_{i=1}^n i^2$
- (b) $\sum_{i=1}^n i^3$
- (c) $\sum_{i=1}^n i^k$
- (d) $\sum_{i=1}^n 2^i$
- (e) $\sum_{i=1}^n k^i$
- (f) $\sum_{i=1}^n i^3 e^i$ where $e \approx 2.71828$

4.7 The following question is aimed at stimulating discussion. It is often said that physicists are searching for a unified theory of physics—an ultimate theory that will explain everything that can be explained. Do you think a unified theory of physics will be expressed mathematically? Will it be a computable axiomatic system pow-

erful enough to describe the arithmetic of the natural numbers? If so, in light of Gödel's Incompleteness theorem, do you think a unified theory of physics is possible? Or will certain truths of the theory be forever beyond proof? That is, if the unified theory of physics is consistent must it be incomplete? And can the consistency of the axioms of the unified theory of physics be proven within the theory?

Part II

What Can You Do with a Quantum Computer?

Chapter 5

Performing Search with a Quantum Computer

“Grover’s quantum searching technique is like cooking a soufflé. You put the state obtained by quantum parallelism in a “quantum oven” and let the desired answer rise slowly. Success is almost guaranteed if you open the oven at just the right time. But the soufflé is very likely to fall—the amplitude of the correct answer drops to zero—if you open the oven too early.”

— Kristen Fuchs¹

“Search” is one of the most pervasive tasks in computer science. Many important problems can be solved by enumerating the possible solutions and then searching amongst them, systematically or randomly, to determine which are correct. In some cases, determining that certain possibilities are incorrect allows you to eliminate others and hence narrow the search for a true solution. These search problems are said to be “structured”. Alternatively, there are other search problems in which you learn nothing useful upon discovering certain possibilities are incorrect, other than the futility of trying those possibilities again. These search problems are said to be “unstructured”. Thus unstructured search is the quintessential “find-the-needle-in-the-haystack” problem.

Grover’s algorithm provides a quantum method for solving *unstructured* search problems in roughly the square root of the number of steps required using a classical computer. This amounts to a *polynomial* speed up over what is possible classically. Although this is not as impressive a speedup as that seen in other quantum algorithms, such as the Deutsch-Jozsa algorithm, for which an exponential speedup is obtained, Grover’s algorithm is applicable to a much wider range of computational problems. Moreover, a quadratic speedup is not bad either. While it won’t tame problems having an exponential complexity scaling it could, nevertheless, allow significantly larger problem instances to be solved than might otherwise be possible. For example, in an airline scheduling problem any given airline only has finitely many aircraft, and finitely many routes. It is quite possible that a quadratic speedup in

¹Source: [71]. Kristen Fuchs is the spouse of quantum computer scientist Chris Fuchs. Her vivid analogy has helped me convey the essence of amplitude amplification to dozens of students in a single sentence.

solving a scheduling problem is sufficient to confer a practical advantage (provided any required quantum error correction overhead is not too great).

5.1 The Unstructured Search Problem

The concept of an unstructured search problem can be demonstrated using a standard telephone directory. A standard telephone directory contains a list of names, ordered alphabetically, together with their associated telephone numbers. To find someone's telephone *number* given knowledge of their *name* you proceed as follows: open the directory at a random page; if the names on the page alphabetically precede the name you want, mark the current page and open the directory again at a later page. If the names alphabetically succeed the name you want, mark the page and open the directory again at an earlier page. For a telephone directory containing N entries, repeating this process, delimited by the marked points, will take you to the sought after entry in roughly $O(\log N)$ steps. Hence, this algorithm is said to have a complexity of $O(\log N)$, which is deemed "efficient" since it is logarithmic in the number of entries in the telephone directory, or equivalently, polynomial in the number of bits, $n = \log_2 N$, needed to assign a unique index to each entry. The fundamental reason that telephone directory look-up can be performed so efficiently is that when you fail to find the sought after name on a given page you nevertheless gain reliable information as to the direction in which to search next. In other words the alphabetically ordered search space is *structured* and you can exploit this structure to narrow the search for a solution.

Now contrast this with the task of using the same telephone directory to find someone's *name* given their telephone *number*. That is, we are now using the telephone directory to do a reverse lookup. In this case, because the telephone directory is unordered with respect to telephone *numbers*, whenever you find a telephone number that is not the given number, you learn nothing useful regarding in which direction to search next, namely, amongst the predecessors or successors of the last telephone number found. In this case, the search you process you are forced to perform is essentially "generate-and-test". This consists of opening the phone book at a random page, if that page contains the given number reading off the corresponding name and stopping. Else marking the page a "dead-end" and picking one of the unread pages at random, repeating this process until the sought after item is found or all the entries have been exhausted. If there are N entries in the telephone directory it would therefore take you, on average, $O(N/2)$ repetitions of the algorithm to find the given telephone number and hence the associated name. In the worst case, it is conceivable a really unlucky person would have to search every entry in the directory only to find the given number at the last trial. So in the worst case it could take $O(N)$ steps.

We can use the aforementioned example of searching a telephone directory to motivate a more formal statement of the unstructured search problem, as follows:

Unstructured Search Consider a search problem that requires us to find a particular target item amongst a set of N candidates. Suppose that these N candidates are

labelled by indices x in the range $0 \leq x \leq N - 1$, and that the index of the sought after target item is $x = t$. Let there be a computational oracle, of “black-box function”, $f_t(x)$ that when presented with an index x can pronounce on whether or not it is the index of the target. Specifically, $f_t(x)$ is defined such that:

$$f_t(x) = \begin{cases} 0 & \text{if } x \neq t \\ 1 & \text{if } x = t \end{cases} \quad (5.1)$$

where 0 stands for “no” and 1 stands for “yes”. The search problem is “unstructured” because there is no discernible pattern to the values of $f_t(x)$ to provide any guidance in finding $x = t$. Our job is to find the index $x = t$, using the fewest calls to the oracle $f_t(x)$.

This formalization of the unstructured search problem will allow us to estimate the computational cost of solving the problem classically versus quantumly. To facilitate this comparison, in the next section we describe the classical generate-and-test algorithm using the language of quantum mechanics. Before we do that it is worth making a few comments about the oracle, or “black-box function” $f_t(x)$.

5.1.1 Meaning of the Oracle

The computational oracle used in Grover’s algorithm has been the source of much confusion to students of quantum computing, because it sounds like the use of the oracle introduces circular reasoning in the search algorithm. You need to know t to build $f_t(x)$ to then use $f_t(x)$ to find t ! Maddening isn’t it?

The basis for the confusion stems from a misunderstanding about the meaning and purpose of the oracle. To computer scientists, an oracle is merely a *fictitious* mathematical device that allows them to estimate the computational cost of some algorithm measured in units of “the number of calls to the oracle”. In the present example, this enables them to compare the *relative* costs of classical unstructured search versus quantum unstructured search in terms how many times each algorithm much call the oracle.

On the other hand, physicists, especially experimental physicists who actually have to *build* quantum computing hardware, cry foul because someone in their lab has to pick t to built a quantum circuit that plays the role of the oracle, $f_t(x)$. So, they cannot see the point of Grover’s algorithm because they already know t to be able to build a contraption that finds t ! All true.

However, this is similar to the situation we encountered when searching a telephone directory for someone’s name given knowledge of their telephone number. When the telephone directory was composed the author must have had knowledge of which telephone number to associate with which name, and vice versa. So the issue is not whether the solution to some search problem is or is not known in advance of the search, but rather how many times we must query the knowledge-holder

before we learn the solution. In the abstract unstructured search problem the knowledge holder is the oracle, or “black-box function” $f_t(x)$. In the example of searching a telephone directory the knowledge holder is the telephone directory itself.

Moreover, when we come to perform unstructured search on real problems, the oracle, which contains explicit foreknowledge of the solution, is replaced, typically, by a polynomial time (or better) testing procedure. This testing procedure only knows the solutions implicitly via the *properties* that a valid solution must possess. A good example is provided by the graph coloring problem, which is an **NP-Complete** problem.

In graph coloring we are required to assign one of k colors to a graph containing n nodes and m edges such that every node is assigned some color, and every pair of nodes that are directly connected by an edge have different colors. As there are n nodes there can be at most $m = \binom{n}{2} = \frac{1}{2}n(n - 1)$ edges, and so we must only check a maximum of $\frac{1}{2}n(n - 1)$ constraints to verify that a given coloring is or is not acceptable. In this case the complexity will be measured in terms of how many times this testing procedure must be called times the cost of running it each time.

These efficient testing procedures could be quite different from problem to problem. So the use of the oracle in Grover’s algorithm is really only a proxy for such a testing procedure in which we assume, arbitrarily, that there is a unit cost per call to the oracle.

We will have more to say about the oracle shortly, but for now we describe a classical algorithm for solving the unstructured search problem in the language of quantum mechanics. Having done so, we will be able to more clearly see how the quantum search algorithm differs from its classical counterpart.

5.2 Classical Solution: Generate-and-Test

As we saw in the telephone directory example, we can solve the unstructured search problem on a classical computer by a procedure known as “generate-and-test”. This can be done with or without replacement of the indices that are tested along the way. The simplest case to analyze is generate-and-test-with-replacement. Here we imagine we have a bag of indices and we repeatedly dip into this bag, pluck out an index, and ask the oracle whether or not this is the target index, $|t\rangle$. If it is, we stop. If not, we put the index back in the bag (this is the “replacement” step), and repeat the process.

This classical procedure can be expressed in quantum mechanical language as follows: a quantum analog of the bag of indices can be regarded as an equally weighted superposition of all the indices in the range $0 \leq x \leq N - 1$, i.e., the state $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$. Similarly, a quantum analog of the act of plucking out an index, at random, can be regarded as the act of reading this superposition (in the index basis). This gives us a particular index, $|x\rangle$ say. Then we ask the oracle whether or not $x = t$.

If there are N indices, these can be expressed in binary notation using $n = \log_2 N$ qubits. Hence, the easiest way to create the equally weighted superposition state

is apply a separate 1-qubit Walsh-Hadamard gate H to each of n qubits prepared initially in the $|0\rangle$ state, i.e., we perform the operation $|00\dots0\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$. When we read this superposition we will obtain a single index non-deterministically. So this simple process mimics the classical generate-and-test procedure.

Generalizing slightly, if we have *partial* information about the identity of the target index we might want to create a superposition that is weighted more towards indices in the vicinity of where we believe the target to lie. We can do so by picking an arbitrary starting state $|s\rangle$ (instead of the state $|0\rangle^{\otimes n}$), and an (almost) arbitrary unitary operator U (instead of $H^{\otimes n}$). We say “almost” arbitrary because if we are to have any hope of finding the target $|t\rangle$ by reading the superposition $U|s\rangle$, we have ensure $U|s\rangle$ has some non-zero component of $|t\rangle$. Otherwise, we would never find $|t\rangle$ no matter how often we prepared and measured $U|s\rangle$.

To recap then, the general set up we have for a quantum description of classical generate-and-test is that we initialize the system to be in some starting state $|s\rangle$ and apply to it an operator U such that $U|s\rangle$ is guaranteed to contain some non-zero component in $|t\rangle$ (for an unknown target index $|t\rangle$). In the absence of any prior information about the target, the most natural choices for $|s\rangle$ and U are $|s\rangle = |00\dots0\rangle$ and $U = H^{\otimes n}$ respectively. These choices *guarantee* that there will be a non-zero overlap between the (unknown) target $|t\rangle$ and $U|s\rangle$, i.e., $\langle t|U|s\rangle \neq 0$, but other choices are possible, and might be advisable, if you have some prior knowledge about the solution.

Each time we re-synthesize and measure $U|s\rangle$ the probability of finding $|t\rangle$ is given by the modulus squared of the overlap between $|t\rangle$ and $U|s\rangle$, i.e., $p_{\text{succ}}^{\text{CLASSICAL}} = |\langle t|U|s\rangle|^2$. Using standard statistical theory, we can infer we will need to repeat this experiment roughly $|\langle t|U|s\rangle|^{-2}$ times to find the solution with probability $O(1)$ (i.e., near certainty). Hence, this is the “classical” complexity for performing an unstructured search for the target using a generate-and-test-with-replacement procedure.

5.3 Quantum Solution: Grover's Algorithm

Can quantum computers do better? We might expect so because quantum computers need not limit themselves to testing each index in succession. Instead, quantum computers can test several indices at once, in superposition, using quantum parallelism. Unfortunately, since we cannot see the results of these tests individually, quantum parallelism *alone* does not confer any advantage whatsoever.

Fortunately, in 1996, Lov Grover, a computer scientist at Lucent Technologies Bell Labs, discovered a new quantum technique called *amplitude amplification*, that can be exploited to make a quantum algorithm for solving the unstructured search problem [217]. The oracle is used to create an amplitude amplification operator that increases the amplitude of the target index within a superposition of indices while decreasing the amplitudes of the non-target indices. Thus by creating a superposition of all the possible indices and then amplitude amplifying the amplitude of the

target index prior to reading this superposition, we can bias the outcome of the measurement in favor of the target index. This is the key idea behind Grover's algorithm.

The amplitude amplification operator has a very simple form. It is built out of three operators related to the starting state $|s\rangle$, the (almost) arbitrary unitary operator U , and the (unknown) target $|t\rangle$. Specifically, the amplitude amplification operator is given by:

$$Q = -U \mathbb{1}_s U^\dagger \mathbb{1}_t \quad (5.2)$$

where $\mathbb{1}_s = \mathbb{1} - 2|s\rangle\langle s|$ is an operator that inverts the phase of the starting state $|s\rangle$, $\mathbb{1}_t = \mathbb{1} - 2|t\rangle\langle t|$ is an operator that inverts the phase of the (unknown) target state $|t\rangle$, and U is the (almost) arbitrary unitary operator that maps the starting state $|s\rangle$ into a superposition that is guaranteed to contain a non-zero component in the target state $|t\rangle$.

As written, it looks like the operator $\mathbb{1}_t$ requires explicit foreknowledge of the target state $|t\rangle$. However, as we will explain in Sect. 5.4.3, the operator $\mathbb{1}_t$ can be created using the oracle, or “black-box” function $f_t(x)$, which in real applications is replaced, typically, by an efficient *testing procedure* that can recognize a target state via its *properties* rather than its *identity*. Thus, in a real application the oracle (and hence $\mathbb{1}_t$) will *not* have explicit foreknowledge of the target state $|t\rangle$. For the moment, however, just assume $\mathbb{1}_t$ is available since this simplifies the discussion of Grover's algorithm.

With these definitions for $|s\rangle$, $|t\rangle$, $\mathbb{1}_s$, $\mathbb{1}_{f_t}$, and U , Grover's algorithm can be described as follows:

Grover's Algorithm

- Given an oracle, or black-box quantum function, $f_t(x)$ that can pronounce on whether or not a given index x is that of a sought after target t construct: an “amplitude amplification” operator $Q = -U \mathbb{1}_s U^\dagger \mathbb{1}_t$ using the black-box function $f_t(x)$ where

$|s\rangle$ = the starting state

$|t\rangle$ = the (unknown) target state

$\mathbb{1}_s = \mathbb{1} - 2|s\rangle\langle s|$

$\mathbb{1}_t = \mathbb{1} - 2|t\rangle\langle t|$ (which is built from $f_t(x)$ without explicit knowledge of $|t\rangle$)

U = any unitary operator such that $\langle t|U|s\rangle \neq 0$

- Compute $|\psi\rangle = Q^k U |s\rangle$, i.e., iterate the operator Q , $k = \frac{\pi}{4} \sqrt{N}$ times on the state $U|s\rangle$.
- Measure each of the n bit values of the state $|\psi\rangle$.
- Result: with high probability, the target state $|t\rangle$

So how exactly does this sequence of operators perform search? And why does the quantum unstructured search algorithm find the target state in just the square root of number of calls to the oracle as does the classical unstructured search algorithm?

5.4 How Does Grover's Algorithm Work?

To understand how Grover's algorithm works let's examine the evolution of the overlap between the (unknown) target state $|t\rangle$ and the amplitude amplified state $Q^k U|s\rangle$.

The amplitude amplification operator $Q = -U \mathbb{1}_s U^\dagger \mathbb{1}_t$ where $|s\rangle$ is the starting state, $\mathbb{1}_s = \mathbb{1} - 2|s\rangle\langle s|$ and $\mathbb{1}_t = \mathbb{1} - 2|t\rangle\langle t|$. The operators $\mathbb{1}_s$ and $\mathbb{1}_t$ both perform controlled-phase flips. Specifically, $\mathbb{1}_s|x\rangle = -|x\rangle$ if and only if $x = s$. Likewise, $\mathbb{1}_t|x\rangle = -|x\rangle$ if and only if $x = t$. We can use these controlled phase flips to build an operator that pumps probability amplitude into the target eigenstate within a superposition at the expense of the amplitude in the non-target eigenstates.

Substituting in the definitions $\mathbb{1}_s = \mathbb{1} - 2|s\rangle\langle s|$ and $\mathbb{1}_t = \mathbb{1} - 2|t\rangle\langle t|$ into $Q = -U \mathbb{1}_s U^\dagger \mathbb{1}_t$ and expanding out the terms we obtain:

$$Q = -\mathbb{1} + 2|t\rangle\langle t| + 2U|s\rangle\langle s|U^\dagger - 4U|s\rangle\langle s|U^\dagger|t\rangle\langle t| \quad (5.3)$$

Next we consider the effect of Q on two states of particular interest, namely, $U|s\rangle$, and $|t\rangle$. A convenient shorthand way to represent Q acting on $U|s\rangle$ and Q acting on $|t\rangle$ is to write the two equations as the following matrix equation:

$$Q \begin{pmatrix} U|s\rangle \\ |t\rangle \end{pmatrix} = \begin{pmatrix} 1 - 4|\langle t|U|s\rangle|^2 & 2\langle t|U|s\rangle \\ -2\langle t|U|s\rangle^* & 1 \end{pmatrix} \begin{pmatrix} U|s\rangle \\ |t\rangle \end{pmatrix} \quad (5.4)$$

When the overlap between $U|s\rangle$ and $|t\rangle$ is very small, i.e., when $u = \langle t|U|s\rangle \ll 1$, the states $U|s\rangle$ and $|t\rangle$ are almost orthogonal to each other, and Q behaves like a 1-qubit rotation gate, in the space spanned by $U|s\rangle$ and $|t\rangle$. In fact, when $\langle t|U|s\rangle \ll 1$ we have $|\langle t|U|s\rangle|^2 \ll |\langle t|U|s\rangle|$, i.e., $|u|^2 \ll |u|$, and so the matrix representation of Q becomes almost the matrix:

$$\begin{aligned} Q \begin{pmatrix} U|s\rangle \\ |t\rangle \end{pmatrix} &\approx \begin{pmatrix} 1 & 2u \\ -2u^* & 1 \end{pmatrix} \begin{pmatrix} U|s\rangle \\ |t\rangle \end{pmatrix} \\ &= \exp \begin{pmatrix} 0 & 2u \\ -2u^* & 0 \end{pmatrix} \begin{pmatrix} U|s\rangle \\ |t\rangle \end{pmatrix} \end{aligned} \quad (5.5)$$

In the matrix exponential form, the k -th power of Q is easy to calculate, and we find:

$$\begin{aligned} Q^k \begin{pmatrix} U|s\rangle \\ |t\rangle \end{pmatrix} &= \exp \begin{pmatrix} 0 & 2ku \\ -2ku^* & 0 \end{pmatrix} \begin{pmatrix} U|s\rangle \\ |t\rangle \end{pmatrix} \\ &\approx \begin{pmatrix} \cos(2k|u|) & \frac{u}{|u|} \sin(2k|u|) \\ -\frac{u^*}{|u|} \sin(2k|u|) & \cos(2k|u|) \end{pmatrix} \begin{pmatrix} U|s\rangle \\ |t\rangle \end{pmatrix} \end{aligned} \quad (5.6)$$

This implies that after k iterations of the amplitude operator Q the overlap between the (unknown) target state $|t\rangle$ and the amplitude amplified state $Q^k U|s\rangle$ will be:

$$\langle t|Q^k U|s\rangle \approx u \cos(2k|u|) + \frac{u}{|u|} \sin(2k|u|) \quad (5.7)$$

Although these expressions are approximate and only valid in the regime where $u = |\langle t|U|s\rangle| \ll 1$, they illustrate the essential feature of amplitude amplification. With u small, and k modest, $u \cos(2k|u|) \approx u$ and $\frac{u}{|u|} \sin(2k|u|) \approx 2ku \equiv 2k\langle t|U|s\rangle$. By (5.7) we then have (crudely) $\langle t|Q^k U|s\rangle \approx (1 + 2k)\langle t|U|s\rangle$, which implies that the overlap grows roughly *linearly* with the number of steps of amplitude amplification, k . Hence the probability, upon reading the quantum memory register, of obtaining the target state grows quadratically with the number of steps of amplitude amplification, i.e.

$$p_{\text{succ}}^{\text{QUANTUM}} \sim k^2 |\langle t|U|s\rangle|^2 \quad (5.8)$$

Compare this to the scaling for the classical generate-and-test algorithm described in Sect. 5.2. There we found the probability of success to scale with the number of repetitions as:

$$p_{\text{succ}}^{\text{CLASSICAL}} \sim k |\langle t|U|s\rangle|^2 \quad (5.9)$$

Thus amplitude amplification has the effect of enhancing the probability of obtaining the solution when the quantum memory register is read after k iterations.

The second major feature of amplitude amplification that is apparent from (5.7) is that the overlap between the target and the amplitude amplified state oscillates. Thus, it is possible to amplitude amplify too far and actually reduce your probability of finding a solution compared to the classical case. This is the reason Fuchs likens amplitude amplification to baking a soufflé.

5.4.1 How Much Amplitude Amplification Is Needed to Ensure Success?

To conclude our analysis, we would like to estimate how many steps of amplitude amplification are required to reach the target state $|t\rangle$ using the amplitude amplification operator Q starting from state $U|s\rangle$. After k rounds of amplitude amplification,

$$Q^k = \begin{pmatrix} \cos(2k|u|) & \frac{u}{|u|} \sin(2k|u|) \\ -\frac{u^*}{|u|} \sin(2k|u|) & \cos(2k|u|) \end{pmatrix} \quad (5.10)$$

which is almost the same as a matrix that rotates a vector through angle θ , i.e.,

$$\begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \quad (5.11)$$

Hence, as $U|s\rangle$ and $|t\rangle$ are almost orthogonal initially, we need to apply Q until we have rotated $U|s\rangle$ by an angle of about $\pi/2$ to reach $|t\rangle$. At that moment if we were to measure the system we would find it in state $|t\rangle$ with probability of order 1. Therefore, the number of iterations of Q that are required to rotate $U|s\rangle$ into $|t\rangle$ is given by $2k|u| = \frac{\pi}{2}$, which implies $k = \frac{\pi}{4u}$. The same result is also evident by

solving for the smallest positive non-zero real value of k such that $u \cos(2k|u|) + \frac{u}{|u|} \sin(2k|u|) = 1$, which also implies $k \rightarrow \frac{\pi}{4} \sqrt{N}$. Either way,

$$k \approx \frac{\pi}{4} |\langle t | U | s \rangle|^{-1} = \frac{\pi}{4} \sqrt{N} \quad (5.12)$$

where $N = 2^n$ is the number of items searched over. Thus the complexity of quantum search scales as the square root of that of classical search. Hence, Grover's algorithm is quadratically faster than the classical algorithm for performing unstructured search.

5.4.2 An Exact Analysis of Amplitude Amplification

The foregoing description of amplitude amplification was motivated from a desire to de-mystify the process of amplitude amplification. However, it is possible to redo the analysis without introducing any approximations whatsoever. When we do so, and as you are asked to do as Exercise 5.2, we find that the exact expression for the net operator that is obtained after k iterations of amplitude amplification is:

$$\mathcal{Q}^k \begin{pmatrix} U|s\rangle \\ |t\rangle \end{pmatrix} = (-1)^k \begin{pmatrix} \mathcal{U}_{2k}(|u|) & -\frac{u}{|u|} \mathcal{U}_{2k-1}(|u|) \\ \frac{u^*}{|u|} \mathcal{U}_{2k-1}(|u|) & -\mathcal{U}_{2k-2}(|u|) \end{pmatrix} \begin{pmatrix} U|s\rangle \\ |t\rangle \end{pmatrix} \quad (5.13)$$

where $u = \langle t | U | s \rangle$ and $\mathcal{U}_\ell(\cos \theta) = \sin((\ell + 1)\theta) / \sin \theta$ is the Chebyshev polynomial of the second kind. This then gives the overlap after k iterations of amplitude amplification between the target state and the amplitude amplified state to be:

$$\langle t | \mathcal{Q}^k U | s \rangle = (-1)^k \left(u \mathcal{U}_{2k}(|u|) - \frac{u}{|u|} \mathcal{U}_{2k-1}(|u|) \right) \quad (5.14)$$

Hence, the probability of success after k iterations of amplitude amplification is:

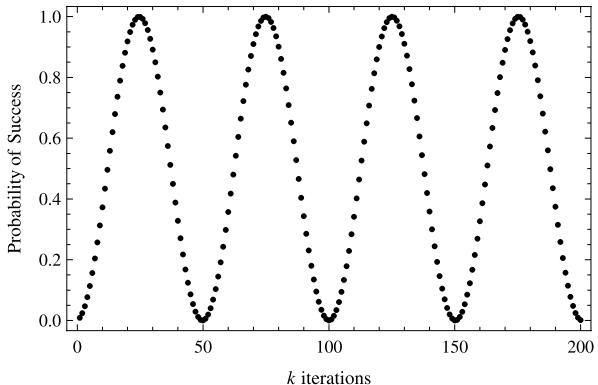
$$\begin{aligned} p_{\text{success}}(k) &= |\langle t | \mathcal{Q}^k U | s \rangle|^2 = \left| (-1)^k \left(u \mathcal{U}_{2k}(|u|) - \frac{u}{|u|} \mathcal{U}_{2k-1}(|u|) \right) \right|^2 = \mathcal{T}_{2k+1}(|u|)^2 \\ &= \cos^2((2k+1) \cos^{-1}(u)) \quad \text{for } k \in \mathbb{Z} \wedge k > 0, \text{ and } u \in \mathbb{R} \wedge 0 < u < 1 \end{aligned} \quad (5.15)$$

where $\mathcal{T}_\ell(|u|)$ is a Chebyshev polynomial of the first kind, $\mathcal{T}_\ell(\cos \theta) = \cos(\ell \theta)$.

Figure 5.1 shows the oscillations in the probability of success of Grover's algorithm with increasing amounts of amplitude amplification. In the figure, there is one solution amongst 2^{10} search items. The maximum probability of success first occurs after $\frac{\pi}{4} \sqrt{2^{10}} \approx 25$ iterations of amplitude amplification, but then declines if one "over-amplifies".

Independent and quite different exact analyses of amplitude amplification are provided in [70] and [61, 200] (for an arbitrary amplitude distribution), but the results are similar.

Fig. 5.1 The probability of success as a function of the number of steps of amplitude amplification for a problem having one solution amongst 2^{10} possibilities. Notice that at first the success probability rises but falls again if one amplitude amplifies too much



5.4.3 The Oracle in Amplitude Amplification

Before we conclude our discussion of Grover's algorithm we need to explain how we can use the oracle, or black-box function $f_t(x)$, to construct the operator $\mathbb{1}_t = \mathbb{1} - 2|t\rangle\langle t|$, which is used within the amplitude amplification procedure.

In mythology an “oracle” is an omniscient person who answers all questions instantly and infallibly. This notion has been borrowed by computer science to conceive of “computational oracles”. These are synonymous with “black-boxes”. You provide an input to the oracle (a “question”) and in one step the oracle responds with the correct answer. The main value of computational oracles is that they allow us to quantify the complexity of complicated algorithms (up to the cost of the oracle) even though parts of those algorithms may be poorly understood. An oracle is a means by which we can compare the relative complexities of two algorithms without necessarily understanding how to implement that oracle. The difference between classical oracles and quantum oracles is in the nature of the questions we can pose and the answers they can give.

As you will recall, the oracle accepts an integer x in the range $0 \leq x \leq N - 1$ and returns 1 or 0 according to whether or not the index is that of the sought after target t , i.e., we have:

$$f_t(x) = \begin{cases} 0 & \text{if } x \neq t \\ 1 & \text{if } x = t \end{cases} \quad (5.16)$$

To create the operator $\mathbb{1}_t$ we introduce a single ancilla to create an $(n + 1)$ -qubit unitary transformation, \mathcal{Q}_t defined as:

$$\mathcal{Q}_f : \mathcal{Q}_f |x\rangle |y\rangle \longrightarrow |x\rangle |y \oplus f_t(x)\rangle \quad (5.17)$$

where $|x\rangle$ one of the indices we want to test, $|y\rangle$ is the ancilla qubit, and $|y \oplus f_t(x)\rangle$ is the exclusive-OR of the bit value of the ancilla and the bit value that is output from our black-box function $f_t(x)$.

Next we prepare the ancilla in the state $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. This can be done easily by applying a 1-qubit Walsh-Hadamard gate, H , to the ancilla prepared initially

in state $|1\rangle$. By the linearity of quantum mechanics, with the ancilla in the state $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ the transformation effected by Ω_t is then:

$$\begin{aligned}\Omega_t|x\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) &= \frac{1}{\sqrt{2}}|x\rangle|0\rangle \oplus f_t(x) - \frac{1}{\sqrt{2}}|x\rangle|1\rangle \oplus f_t(x) \\ &= \frac{1}{\sqrt{2}}|x\rangle|f_t(x)\rangle - \frac{1}{\sqrt{2}}|x\rangle|1\rangle \oplus f_t(x) \\ &= \begin{cases} \frac{1}{\sqrt{2}}|x\rangle(|0\rangle - |1\rangle) & \text{if } x \neq t \text{ and therefore } f_t(x) = 0 \\ \frac{1}{\sqrt{2}}|x\rangle(|1\rangle - |0\rangle) & \text{if } x = t \text{ and therefore } f_t(x) = 1 \end{cases} \\ &= (-1)^{f_t(x)}|x\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \text{ either way} \quad (5.18)\end{aligned}$$

Thus, regardless of whether $x = t$ or $x \neq t$ the transformation performed by Ω_t is:

$$\Omega_t : \Omega_t|x\rangle|y\rangle \longrightarrow (-1)^{f_t(x)}|x\rangle|y\rangle \quad (5.19)$$

when the ancilla $|y\rangle$ is specialized to be in the input state $|y\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

To obtain $\mathbb{1}_t$ from Ω_t we merely ignore the ancilla qubit! Then the transformation we see on the first n qubits is:

$$\mathbb{1}_t : \mathbb{1}_t|x\rangle \longrightarrow (-1)^{f_t(x)}|x\rangle = (\mathbb{1} - 2|t\rangle\langle t|)|x\rangle \quad (5.20)$$

Thus the operator $\mathbb{1}_t$, which appears to require explicit foreknowledge of the state $|t\rangle$ can in fact be obtained from the oracle $f_t(x)$. Again, in practical applications the oracle is replaced by a polynomial time (or better) testing procedure that can recognize the target state via its' properties but does not necessarily know those target states explicitly in advance.

5.5 Quantum Search with Multiple Solutions

Many search problems have multiple, equally acceptable, solutions. In such cases there will be multiple index values of j for which $f(j) = 1$. If there are N items to search amongst, of which exactly t are solutions, we next show that the number of amplitude steps needed to ensure success becomes $\frac{\pi}{4}\sqrt{N/t}$. Each time the Grover search algorithm is run on such a problem, the algorithm will return any *one* of these t solutions with equal probability.

Let us consider the case of an unstructured quantum search problem that has multiple, specifically t , solutions out of a total number of $N = 2^n$ possible index values. That is, there are exactly t solutions to the equation $f(j) = 1$ where j is an n bit index value. How would quantum search work in this case?

The following beautiful approach to analyzing this problem was developed by Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp [78]. One can consider the index values falling naturally into two sets: the set of “good” index values, $j \in \mathcal{G}$ for which $f(j) = 1$ and the set of “bad” index values, $j \in \mathcal{B}$ for which $f(j) = 0$, with the number of solutions being equal to the cardinality of the good set, i.e., $t = |\mathcal{G}|$. Therefore, if we define two superpositions:

$$\begin{aligned} |\psi_{\text{good}}\rangle &= \frac{1}{\sqrt{t}} \sum_{j \in \mathcal{G}} |j\rangle \\ |\psi_{\text{bad}}\rangle &= \frac{1}{\sqrt{N-t}} \sum_{j \in \mathcal{B}} |j\rangle \end{aligned} \quad (5.21)$$

a superposition consisting of all possible indices can be expressed as a combination of $|\psi_{\text{good}}\rangle$ and $|\psi_{\text{bad}}\rangle$, namely:

$$|\psi\rangle = H^{\otimes n} |0\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle = \sqrt{\frac{t}{N}} |\psi_{\text{good}}\rangle + \sqrt{\frac{N-t}{N}} |\psi_{\text{bad}}\rangle \quad (5.22)$$

where H is a Walsh-Hadamard gate. For clarity, we introduce a parameter θ defined via $\sin \theta = \sqrt{\frac{t}{N}}$, and define a state, $|\bar{\psi}\rangle$, orthogonal to $|\psi\rangle$ that will prove to be useful shortly. Thus, we can write:

$$|\psi\rangle = \sin \theta |\psi_{\text{good}}\rangle + \cos \theta |\psi_{\text{bad}}\rangle \quad (5.23)$$

$$|\bar{\psi}\rangle = \cos \theta |\psi_{\text{good}}\rangle - \sin \theta |\psi_{\text{bad}}\rangle \quad (5.24)$$

With these definitions, it is apparent that the $\{|\psi\rangle, |\bar{\psi}\rangle\}$ -basis spans the same space as the $\{|\psi_{\text{good}}\rangle, |\psi_{\text{bad}}\rangle\}$ -basis, and we can flip back and forth between these bases in our analyses. As we will need them later, we note that the basis transformations in the other direction are given by inverting (5.23) and (5.24) to yield:

$$|\psi_{\text{good}}\rangle = \sin \theta |\psi\rangle + \cos \theta |\bar{\psi}\rangle \quad (5.25)$$

$$|\psi_{\text{bad}}\rangle = \cos \theta |\psi\rangle - \sin \theta |\bar{\psi}\rangle \quad (5.26)$$

We can now re-interpret the objective of Grover’s algorithm as being to take an equally weighted superposition of all possible indices, $|\psi\rangle$, into $|\psi_{\text{good}}\rangle$, and then measure this state to reveal one of the index values j that solves $f(j) = 1$.

5.5.1 Amplitude Amplification in the Case of Multiple Solutions

By the above construction, the probability of finding a solution (naively) simply by measuring the equal superposition state, $|\psi\rangle$, is (as seen from (5.22)) $\frac{t}{N}$, which

is exactly what one expects classically by a random generate-and-test approach. However, if we amplitude amplify the equal superposition state *before* making our final measurement then we can boost our chances of success considerably. For this we need the t -solutions analog of the “amplitude amplification” operator, Q , which we built for the single-solution case. We will use the same symbol for this new operator here as it plays the same role although its definition is changed to:

$$Q = \underbrace{-H^{\otimes n} \mathbb{1}_s H^{\otimes n}}_{U_\psi^\perp} \underbrace{\mathbb{1}_t}_{U_f} = U_\psi^\perp U_f \quad (5.27)$$

where U_ψ^\perp and U_f are the unitary matrices needed to perform the following operations:

$$\begin{aligned} U_\psi^\perp |\psi\rangle &= |\psi\rangle \\ U_\psi^\perp |\bar{\psi}\rangle &= -|\bar{\psi}\rangle \end{aligned} \quad (5.28)$$

and

$$\begin{aligned} U_f |\psi_{\text{good}}\rangle &= -|\psi_{\text{good}}\rangle \\ U_f |\psi_{\text{bad}}\rangle &= |\psi_{\text{bad}}\rangle \end{aligned} \quad (5.29)$$

In analogy with the single-solution quantum search, the amplitude amplification operator rotates the state vector being amplitude amplified within a two-dimensional sub-space spanned by the basis vectors $\{|\psi_{\text{bad}}\rangle, |\psi_{\text{good}}\rangle\}$ or, equally, the basis vectors $\{|\psi\rangle, |\bar{\psi}\rangle\}$. The transformations Q performs are as follows:

$$\begin{aligned} Q|\psi\rangle &= U_\psi^\perp U_f |\psi\rangle = U_\psi^\perp U_f (\sin \theta |\psi_{\text{good}}\rangle + \cos \theta |\psi_{\text{bad}}\rangle) \\ &= U_\psi^\perp (-\sin \theta |\psi_{\text{good}}\rangle + \cos \theta |\psi_{\text{bad}}\rangle) = U_\psi^\perp (\cos 2\theta |\psi\rangle - \sin 2\theta |\bar{\psi}\rangle) \\ &= \cos 2\theta |\psi\rangle + \sin 2\theta |\bar{\psi}\rangle = \cos 3\theta |\psi_{\text{good}}\rangle + \sin 3\theta |\psi_{\text{bad}}\rangle \end{aligned} \quad (5.30)$$

where we used (5.25) and (5.26) to switch from the $\{|\psi_{\text{good}}\rangle, |\psi_{\text{bad}}\rangle\}$ -basis to the $\{|\psi\rangle, |\bar{\psi}\rangle\}$ -basis. Likewise, for the orthogonal input state, $|\bar{\psi}\rangle$, we have:

$$\begin{aligned} Q|\bar{\psi}\rangle &= U_\psi^\perp U_f |\bar{\psi}\rangle = U_\psi^\perp U_f (\cos \theta |\psi_{\text{good}}\rangle - \sin \theta |\psi_{\text{bad}}\rangle) \\ &= U_\psi^\perp (-\cos \theta |\psi_{\text{good}}\rangle - \sin \theta |\psi_{\text{bad}}\rangle) = U_\psi^\perp (-\sin 2\theta |\psi\rangle - \cos 2\theta |\bar{\psi}\rangle) \\ &= -\sin 2\theta |\psi\rangle + \cos 2\theta |\bar{\psi}\rangle = -\sin 3\theta |\psi_{\text{good}}\rangle + \cos 3\theta |\psi_{\text{bad}}\rangle \end{aligned} \quad (5.31)$$

Thus the effect of Q is to rotate the initial state, $|\psi\rangle = \sin \theta |\psi_{\text{good}}\rangle + \cos \theta |\psi_{\text{bad}}\rangle$, through an angle of 2θ . Hence, in the $\{|\psi_{\text{good}}\rangle, |\psi_{\text{bad}}\rangle\}$ -basis, Q takes the form:

$$Q = \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ -\sin 2\theta & \cos 2\theta \end{pmatrix} \quad (5.32)$$

where $\sin \theta = \sqrt{\frac{t}{N}}$. When Q is so defined, we have:

$$\begin{aligned} Q|\psi\rangle &= Q(\sin \theta |\psi_{\text{good}}\rangle + \cos \theta |\psi_{\text{bad}}\rangle) \\ &= \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ -\sin 2\theta & \cos 2\theta \end{pmatrix} \cdot \begin{pmatrix} \sin \theta \\ \cos \theta \end{pmatrix} \\ &= \begin{pmatrix} \sin 3\theta \\ \cos 3\theta \end{pmatrix} = \sin 3\theta |\psi_{\text{good}}\rangle + \cos 3\theta |\psi_{\text{bad}}\rangle \end{aligned} \quad (5.33)$$

To predict the affect of k successive applications of Q , we compute:

$$Q^k = \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ -\sin 2\theta & \cos 2\theta \end{pmatrix}^k = \begin{pmatrix} \cos(2k\theta) & \sin(2k\theta) \\ -\sin(2k\theta) & \cos(2k\theta) \end{pmatrix} \quad (5.34)$$

Hence, when applied to the initial state $|\psi\rangle = \sin \theta |\psi_{\text{good}}\rangle + \cos \theta |\psi_{\text{bad}}\rangle$ we obtain

$$\begin{aligned} Q^k |\psi\rangle &= Q^k (\sin \theta |\psi_{\text{good}}\rangle + \cos \theta |\psi_{\text{bad}}\rangle) \\ &= \sin((2k+1)\theta) |\psi_{\text{good}}\rangle + \cos((2k+1)\theta) |\psi_{\text{bad}}\rangle \end{aligned} \quad (5.35)$$

Consequently, to obtain a solution to $f(j) = 1$ by first amplitude amplifying $|\psi\rangle$ a number of times k , and then measuring the resulting state, we will obtain a success probability of $\mathcal{O}(1)$ provided we pick the smallest integer k such that $(2k+1)\theta \approx \frac{\pi}{2}$. As $\theta = \sqrt{\frac{t}{N}}$, this implies $k = \frac{\pi}{4}\sqrt{\frac{N}{t}} - 1/2$, i.e., $\mathcal{O}(\sqrt{\frac{N}{t}})$. Thus, classically a solution can be found in $\mathcal{O}(\frac{N}{t})$ trials, whereas quantumly one can be found in $\mathcal{O}(\sqrt{\frac{N}{t}})$ trials. As in the case of a single solution, we again see a *square root* speedup for the case when there are t solutions out of $N = 2^n$ candidates.

If the number of solutions t to a multi-solution quantum search problem is not known in advance, then quantum search can be combined with another quantum algorithm—called quantum *counting*—to efficiently *count* the number of solutions prior to running the quantum search algorithm. The quantum counting algorithm is described in Chap. 10.

5.6 Can Grover's Algorithm Be Beaten?

It is natural to think that Grover's algorithm is just the first example of a quantum algorithm for solving unstructured search problems and that in time most likely better variants will emerge. Unfortunately, this will not be the case. Remarkably, Christoph Zalka has proved that Grover's algorithm is *optimal* [557]. This means that any other quantum algorithm for performing unstructured quantum search must call the oracle as least as many times as is done by Grover's algorithm. Nor can you parallelize Grover's algorithm to any extent better than merely partitioning the search space amongst multiple quantum computers.

Nevertheless, the fact that there is generally a non-zero probability of success when you terminate Grover's algorithm after exactly k rounds of amplitude amplification allows us to consider an “early-termination” strategy. That is, terminate Grover's algorithm for $k < \frac{\pi}{4}\sqrt{N}$ rounds of amplitude amplification and read the result. If it is the solution stop; if not restart a new Grover search and run it for another k rounds of amplitude amplification. On average the cost of running such an algorithm will be:

$$\begin{aligned} C_{\text{avg}} &= kp_{\text{succ}}(k) + 2kp_{\text{succ}}(k)(1 - p_{\text{succ}}(k)) + 3kp_{\text{succ}}(k)(1 - p_{\text{succ}}(k))^2 + \dots \\ &= \sum_{i=1}^{\infty} ikp_{\text{succ}}(k)(1 - p_{\text{succ}}(k))^{i-1} = \frac{k}{p_{\text{succ}}(k)} \end{aligned} \quad (5.36)$$

where p_{succ} is the probability of success after k rounds of amplitude amplification.

5.7 Some Applications of Quantum Search

Grover's algorithm may lack the impressive exponential speedup seen in the Deutsch-Jozsa, Shor (Quantum Factoring), Eigenvalue Estimation, and Quantum Simulation algorithms, but it has proven to be surprisingly versatile in its own right and as a sub-routine in other quantum algorithms. In Chap. 10 we will give several examples of how quantum search can be used to speed up the solution of various problems in mathematics. Here we focus on how quantum search can be used as a within computer science.

5.7.1 Speeding Up Randomized Algorithms

One of the most effective types of search algorithms for hard computational problems, such as the traveling salesperson problem, are “randomized algorithms” [364]. In a classical randomized algorithm, we use a sequence of pseudo-random numbers to determine a trajectory through the search space. Figure 5.2 shows four runs of a hypothetical randomized algorithm that samples different paths through a search space. At each step the search can go up one step, down one step, or stay at the same level with probabilities reflecting slight local preferences to go up rather than down. After a certain number of steps we assess whether the state reached is deemed a “solution” state. Randomized algorithms usually work such that they either converge on a desired solution after a certain number of steps, or else, they tend to wander aimlessly in the wrong region of the solution space until we give up and run the whole algorithm again using a different seed for the pseudo-random number generator. Quantum search can speed up such classical randomized algorithms [97] by using a superposition of seeds for the pseudo-random number generator to create a superposition of final states that is very likely to contain a solution within it. We can then use quantum search to amplitude amplify this superposition to extract the desired solution in the square root of the number of parallel pseudo-random trials.

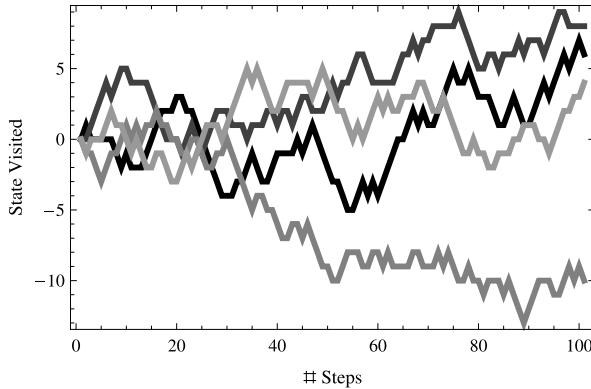


Fig. 5.2 Randomized algorithms use a seed in a pseudo-random number generator to determine a trajectory through the search space. Different seeds lead to different trajectories. If one creates a superposition of seeds, one obtains a superposition of trajectories. If enough seeds are chosen, one or more of these trajectories are likely to terminate in a valid solution. Inspecting this superposition picks out a trajectory at random. But amplitude amplifying the superposition before it is inspected, amplifies the probability of obtaining one of the solution trajectories and suppressed the non-solution trajectories

5.7.2 Synthesizing Arbitrary Superpositions

A final application of quantum search is in the domain of experimental physics to prepare selected superposition states [222]. For example, if we want to create a superposition of indices that correspond to just prime numbers, we could invent an oracle $f(x)$ that returns 1 if x is a prime and 0 otherwise. By amplitude amplifying an equally weighted superposition of indices in some range, we could selectively create a state that is just a superposition of prime numbers within this range. Thus, the quantum search algorithm might find a role in experimental quantum physics as a way of systematically manufacturing desired superposition states.

Quantum State Synthesis Based on Grover's Algorithm

1. Given an n -qubit state $|\psi\rangle = \sum_{i=0}^{2^n-1} c_i |i\rangle$ that we wish to synthesize....
2. Introduce a single extra ancilla qubit prepared initial in state $|0\rangle$ to make a register containing $(1+n)$ qubits initialized to state $|0\rangle|00\dots 0\rangle$.
3. Define $U_1 = (\mathbb{1} \otimes H \otimes H \otimes \dots \otimes H)$ where $\mathbb{1}$ is the 1-qubit identity gate, and H is the 1-qubit Walsh-Hadamard gate.
4. Define U_2 , a matrix implementing $U_2|0\rangle|i\rangle \rightarrow c_i|0\rangle|i\rangle + \sqrt{1 - |c_i|^2}|1\rangle|i\rangle$ plus the remaining orthonormal columns.
5. Define $\mathbb{1}_t = \text{diag}(-1, -1, \dots, -1, +1, +1, \dots, +1)$ (i.e., a sequence of $(2^n) - 1$'s followed by $(2^n) + 1$'s).
6. Define $\mathbb{1}_s = \text{diag}(-1, +1, +1, \dots, +1)$ (i.e., a single -1 followed by $(2^{n+1} - 1) + 1$'s).
7. Define $U = U_2 \cdot U_1$.

8. Define $Q = -(\mathbb{1}_s \cdot U^{-1} \cdot \mathbb{1}_t \cdot U)$.
9. Compute $UQ^m|0\rangle|00\dots0\rangle$ where $m = \frac{\pi}{4}\sqrt{\frac{2^n}{\sum_{i=0}^{2^n-1}|c_i|^2}}$.
10. Measure the ancilla (i.e. the first qubit). If you find the ancilla in state $|0\rangle$, the remaining qubits will be in the state we wish to synthesize.

Let's trace through the execution of this algorithm being used to synthesize the state $|\psi\rangle = \frac{1}{\sqrt{87}}(2|00\rangle - 3|01\rangle + 5i|10\rangle - 7i|11\rangle)$. This is a fairly complicated state: the amplitude of each eigenstate is either positive or negative, real or imaginary, and their magnitudes are relatively prime to one another. Let's see how the quantum state synthesis algorithm works in this case.

First let's re-write the amplitudes in $|\psi\rangle$ using approximate numbers to better visualize what is going on. We have $|\psi\rangle = \frac{1}{\sqrt{87}}(2|00\rangle - 3|01\rangle + 5i|10\rangle - 7i|11\rangle) = c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle$, where $c_0 = 0.214423$, $c_1 = -0.321634$, $c_2 = 0.536056i$, and $c_3 = -0.750479i$. This will help explain the form of U_2 below.

Next we compute the form of $U_1 = (\mathbb{1} \otimes H \otimes H)$ where $\mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. This gives:

$$U_1 = (\mathbb{1} \otimes H \otimes H) = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \quad (5.37)$$

Next we compute a suitable matrix for U_2 . This is more tricky. We seek a U_2 such that $U_2|0\rangle|i\rangle \rightarrow c_i|0\rangle|i\rangle + \sqrt{1 - |c_i|^2}|1\rangle|i\rangle$. Note that, given this definition, U_2 is only partially specified, because we only care about how four of the eight possible basis eigenstates are transformed under the action of U_2 . Specifically, we only care how the basis eigenstates $|0\rangle|00\rangle$, $|0\rangle|01\rangle$, $|0\rangle|10\rangle$, and $|0\rangle|11\rangle$ are transformed under U_2 . The remaining eigenstates can be transformed in any way we pleased so long as the full U_2 matrix is unitary. So the easiest way to build a suitable matrix for U_2 is to start with a “blank” matrix (say all zeroes) and fill in matrix elements to comply with the prescription for how U_2 is to map the four eigenstates $|0\rangle|00\rangle$, $|0\rangle|01\rangle$, $|0\rangle|10\rangle$, and $|0\rangle|11\rangle$. We will then complete U_2 by finding values for the remaining rows and columns sufficient to guarantee that all the rows (and, equivalently, all the columns) are orthonormal and hence U_2 is unitary.

So to fix the first requirement $U_2|0\rangle|00\rangle \rightarrow c_0|0\rangle|00\rangle + \sqrt{1 - |c_0|^2}|1\rangle|00\rangle$ we define the first column of U_2 to be:

$$\begin{pmatrix} c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sqrt{1 - |c_0|^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.38)$$

Next we insert the second column for U_2 which thereby fixes the transformation for both the $|0\rangle|00\rangle$ (first column) and $|0\rangle|01\rangle$ (second column) states:

$$\begin{pmatrix} c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sqrt{1 - |c_0|^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sqrt{1 - |c_1|^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.39)$$

Continuing, in the same manner to insert the third and fourth columns of U_2 then yields:

$$\begin{pmatrix} c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_3 & 0 & 0 & 0 & 0 \\ \sqrt{1 - |c_0|^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sqrt{1 - |c_1|^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{1 - |c_2|^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{1 - |c_3|^2} & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.40)$$

Now we are almost done. All that remains is to find any set of vectors for last four columns (which are currently “blank”) such that they are orthonormal to the first four columns we have just defined. We can do this easily using (say) the Gram-Schmidt orthogonalization procedure. Thus we obtain a suitable form for U_2 as

being:

$$U_2 = \begin{pmatrix} 0.214 & 0 & 0 & 0 & -0.977 & 0 & 0 & 0 \\ 0 & -0.322 & 0 & 0 & 0 & 0.607 & -0.503i & 0.525 \\ 0 & 0 & 0.536i & 0 & 0 & -0.598 & -0.581i & 0.134 \\ 0 & 0 & 0 & -0.751i & 0 & -0.196i & -0.327 & 0.54i \\ 0.977 & 0 & 0 & 0 & 0.214 & 0 & 0 & 0 \\ 0 & 0.947 & 0 & 0 & 0 & 0.206 & -0.171i & 0.178 \\ 0 & 0 & 0.844 & 0 & 0 & -0.379i & 0.369 & 0.085i \\ 0 & 0 & 0 & 0.661 & 0 & -0.223 & 0.371i & 0.613 \end{pmatrix} \quad (5.41)$$

You can check that U_2 is unitary by verifying² $U_2 \cdot U_2^\dagger = \mathbb{1}_8$ where $\mathbb{1}_8$ is an 8×8 identity matrix.

$$\mathbb{1}_t = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.42)$$

$$\mathbb{1}_s = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.43)$$

$$U = \begin{pmatrix} 0.107 & 0.107 & 0.107 & 0.107 & -0.488 & -0.488 & -0.488 & -0.488 \\ -0.161 & 0.161 & -0.161 & 0.161 & 0.566 - 0.252i & -0.566 - 0.252i & 0.041 + 0.252i & -0.041 + 0.252i \\ 0.268i & 0.268i & -0.268i & -0.268i & -0.232 - 0.291i & 0.232 - 0.291i & -0.366 + 0.291i & 0.366 + 0.291i \\ -0.375i & 0.375i & 0.375i & -0.375i & -0.163 + 0.172i & -0.163 - 0.172i & 0.163 - 0.368i & 0.163 + 0.368i \\ 0.488 & 0.488 & 0.488 & 0.488 & 0.107 & 0.107 & 0.107 & 0.107 \\ 0.473 & -0.473 & 0.473 & -0.473 & 0.192 - 0.085i & -0.192 - 0.085i & 0.014 + 0.085i & -0.014 + 0.085i \\ 0.422 & 0.422 & -0.422 & -0.422 & 0.185 - 0.147i & 0.185 + 0.147i & -0.185 - 0.232i & -0.185 + 0.232i \\ 0.33 & -0.33 & -0.33 & 0.33 & 0.195 + 0.185i & -0.195 + 0.185i & -0.418 - 0.185i & 0.418 - 0.185i \end{pmatrix} \quad (5.44)$$

²Note you will get some small round off errors because we have only written the elements of U_2 to four decimal places.

$$Q = \begin{pmatrix} 0.5 & 0.167 & 0.351 & -0.109 & 0.571 - 0.083i & -0.051 + 0.166i & -0.314 - 0.238i & 0.212 + 0.155i \\ -0.167 & -0.5 & 0.109 & -0.351 & 0.051 + 0.166i & -0.571 - 0.083i & -0.212 + 0.155i & 0.314 - 0.238i \\ -0.351 & 0.109 & -0.5 & -0.167 & -0.002 + 0.079i & 0.104 + 0.328i & -0.55 - 0.4i & 0.029 - 0.007i \\ 0.109 & -0.351 & -0.167 & -0.5 & 0.104 - 0.328i & -0.002 - 0.079i & 0.029 + 0.007i & -0.55 + 0.4i \\ -0.571 - 0.083i & 0.051 - 0.166i & -0.002 - 0.079i & 0.104 + 0.328i & 0.632 & 0.019 - 0.188i & 0.218 + 0.022i & 0.039 + 0.166i \\ 0.051 + 0.166i & -0.571 + 0.083i & 0.104 - 0.328i & -0.002 + 0.079i & 0.019 + 0.188i & 0.632 & 0.039 - 0.166i & 0.218 - 0.022i \\ 0.314 - 0.238i & -0.212 - 0.155i & -0.55 + 0.4i & 0.029 - 0.007i & 0.218 - 0.022i & 0.039 + 0.166i & 0.368 & 0.284 - 0.143i \\ -0.212 + 0.155i & 0.314 + 0.238i & 0.029 + 0.007i & -0.55 - 0.4i & 0.039 - 0.166i & 0.218 + 0.022i & 0.284 + 0.143i & 0.368 \end{pmatrix} \quad (5.45)$$

The operation $U \cdot Q^m |0\rangle |00\rangle$ produces the state:

$$\begin{aligned} 0.177244|000\rangle - 0.265866|001\rangle + 0.44311i|010\rangle - 0.620354i|011\rangle \\ - 0.31736|100\rangle - 0.307652|101\rangle - 0.274289|110\rangle - 0.214736|111\rangle \end{aligned} \quad (5.46)$$

We now read the first qubit (the ancilla). It is found to be $|0\rangle$ with probability $|0.177244|^2 + |-0.265866|^2 + |0.44311i|^2 + |-0.620354i|^2 = 0.683287$. In this case the superposition is projected into the state:

$$0.214423|000\rangle - 0.321634|001\rangle + 0.536056i|010\rangle - 0.750479i|011\rangle \quad (5.47)$$

showing that the second and third qubits now correspond to the state we wished to synthesize namely $|\psi\rangle = \frac{1}{\sqrt{87}}(2|00\rangle - 3|01\rangle + 5i|10\rangle - 7i|11\rangle) = 0.214423|00\rangle - 0.321634|01\rangle + 0.536056i|10\rangle - 0.750479i|11\rangle$.

5.8 Quantum Searching of Real Databases

Shortly after Grover's algorithm was published I received a telephone call from someone at Oracle asking me if I thought Grover's algorithm could be used to search a real database. I answered that it could provided the database was encoded in a quantum state, and the oracle (with a little "o") was replaced with a testing procedure.

Clearly, to obtain a *practically useful* algorithm, and to build a practically useful quantum computer capable of running that algorithm, we cannot rely upon foreknowledge of the solution to the problem we are trying to solve. Therefore, to obtain a practically useful quantum algorithm, we must always replace the use of an oracle defined via the black box *function* $f_t(x)$ (which contains explicit knowledge of the solution in advance) with an efficient (i.e., polynomially bounded) *procedure* that applies a test to the index being queried sufficient to decide whether or not that index meets the criteria for being the target t . Typically, this testing procedure will involve checking that a purported solution exhibits all the required properties that an acceptable solution must possess.

For example, consider the graph coloring problem. Here we have to assign one of k colors to a graph having n nodes and m edges such that every node has some color and no two nodes connected directly by an edge share the same color. These constraints express what it means to be a solution, but do not require foreknowledge of the solutions explicitly. In this example, the graph coloring problem is **NP-Complete**, and can be very challenging. Nevertheless, for any given graph, and any

given value of k , we have at most $\binom{n}{2}$ pairs of nodes we must check to ensure they are colored acceptably. Hence, the test is at most quadratic in the number of nodes and hence we can easily devise an efficient testing procedure that can check whether a proposed coloring satisfies all the requirements to be a true solution. When we swap out the oracle for one of these efficient testing procedures instead, the cost of running the algorithm becomes measured in terms of how many times we must call this testing procedure times the cost per run.

By using an efficient *procedure* to test whether an index is or is not the target, one can avoid having to know the identity of that target in advance). Fortunately, there are many important computational problems, such as all **NP-Complete** problems, which admit such an efficient testing procedures. So the oracular quantum algorithms are not generally directly useful algorithms for solving real problems. But they do simplify the assessment of the relative costs of quantum and classical algorithms solving the same problem with access to the same (fictional) oracle.

5.9 Summary

Although Grover's algorithm offers only a polynomial speedup over what we can do classically, it is nevertheless extremely versatile and has inspired several other quantum algorithms. Moreover, by nesting one quantum search algorithm within another, even more impressive speedups appear to be possible, and a better-than-classical exponential time quantum algorithm for **NP-Hard** problems appears to be within reach.

Originally, Grover's algorithm was called the database search algorithm, but this name was dropped because it misled people into thinking that it could be used to search real databases when, in fact, it cannot, at least not without first encoding the database in the quantum state to be amplitude amplified [558]. If this encoding is done naively, the cost of creating the database would be linear in its size—that is, $\mathcal{O}(N)$. Thus, the cost of encoding followed by quantum search would be $\mathcal{O}(N + \sqrt{N})$, whereas the cost of a classical search alone would be just $\mathcal{O}(N)$ —beating the quantum scheme. More clever (parallel) encoding schemes might be feasible but they would seem to necessitate trading time complexity for space complexity. Nevertheless, in some applications, this might be acceptable. For Grover's algorithm to be of practical use, we must avoid creating the database explicitly and work instead with a set of indices that enumerate distinct candidate solutions to some problem. Provided we can map an index to a particular candidate solution efficiently and then test it for correctness in polynomial time, we would have a quantum search procedure that could work on interesting problems such as the **NP-Hard** and **NP-Complete** problems. However, if there is a systematic mapping between an index and the candidate solution, the problem must have some internal structure, and is not therefore truly an unstructured search problem.

Grover's algorithm is can often be used as a sub-routine in more sophisticated quantum algorithms. In this chapter we looked at applications in physics, e.g., synthesizing arbitrary superposition states, and in computer science, e.g., speeding up

randomized algorithms. In the next chapter we shall find it useful for speeding up the breaking of the classical cryptosystem called DES [71]. In Chap. 10 we will also find it useful in speeding up mean estimation and counting problems.

5.10 Exercises

5.1 Describe how you would use Grover's algorithm to synthesize the states:

- (a) $|\psi\rangle = \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$
- (b) $|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$
- (c) $|\psi\rangle = \frac{1}{\sqrt{3}}|001\rangle + \frac{1}{\sqrt{3}}|010\rangle + \frac{1}{\sqrt{3}}|100\rangle$

5.2 The analysis of Grover's algorithm required us to compute the k -th power of the matrix Q where:

$$Q = \begin{pmatrix} 1 - 4|u|^2 & 2u \\ -2u^* & 1 \end{pmatrix} \quad (5.48)$$

where u is an arbitrary complex number. In the chapter we deduced that:

$$Q^k \approx \begin{pmatrix} \cos(2k|u|) & \frac{u}{|u|} \sin(2k|u|) \\ -\frac{u^*}{|u|} \sin(2k|u|) & \cos(2k|u|) \end{pmatrix} \quad (5.49)$$

However, Q^k can be computed exactly in terms of Chebyshev polynomials. We find that:

$$Q^k = (-1)^k \begin{pmatrix} \mathcal{U}_{2k}(|u|) & -\frac{u}{|u|} \mathcal{U}_{2k-1}(|u|) \\ \frac{u^*}{|u|} \mathcal{U}_{2k-1}(|u|) & -\mathcal{U}_{2k-2}(|u|) \end{pmatrix} \quad (5.50)$$

where $\mathcal{U}_k(\cos \theta) = \sin((k+1)\theta)/\sin \theta$ is the Chebyshev polynomial of the second kind. Use a proof by induction to show that this form is correct. You will find the following facts to be useful: $\mathcal{U}_0(x) = 1$, $\mathcal{U}_1(x) = 2x$, $\mathcal{U}_2(x) = 4x^2 - 1$ and the Chebyshev polynomials are related to one another via the recursion formula $\mathcal{U}_{k+1}(x) - 2x\mathcal{U}_k(x) + \mathcal{U}_{k-1}(x) = 0$.

5.3 The success probability of Grover's algorithm after k rounds of amplitude amplification is given elsewhere as $p_{\text{succ}} = \sin^2((2k+1)\arcsin(u))$ where u is the overlap between the target state and the starting state, i.e., $u = \langle t|U|s\rangle$. Yet in this chapter we derive the same result as $p_{\text{succ}} = \mathcal{T}_{2k+1}(|u|)^2 = \cos^2((2k+1)\arccos(u))$ for $k \in \mathbb{Z} \wedge k > 0$ and $u \in \mathbb{R} \wedge 0 < u < 1$. Show that there is no discrepancy by proving $\cos^2((2k+1)\arccos(u)) = \sin^2((2k+1)\arcsin(u))$ when $k \in \mathbb{Z} \wedge k > 0$ and $u \in \mathbb{R} \wedge 0 < u < 1$.

Chapter 6

Code Breaking with a Quantum Computer

“There are three names in quantum computing: Shor, Alice and Bob”

– Michail Zak

The next quantum algorithm we will consider is the most famous of all—Peter Shor’s quantum algorithm for factoring composite integers [455]. The discovery of this algorithm played a pivotal role in transforming the field of quantum computing from an esoteric backwater of computer science to the mainstream.

How this algorithm came to have such importance is interesting story. Every few months the National Security Agency receives an academic paper from someone claiming to have made a significant breakthrough in cryptography. In most cases, the claim proves to be unfounded, and the purported result can be ruled out quickly by NSA’s expert cryptographers and mathematicians. When Peter Shor’s algorithm arrived at the NSA it was met with the usual degree of skepticism. The state-of-the-art classical algorithm for factoring composite integers was, at that time (and still is), the Number Field Sieve [309]. The running time of this algorithm is super-polynomial in the number of bits needed to represent the integer being factored. So as bigger integers are factored the time needed to do so grows faster than any polynomial. This makes factoring an intractable problem for conventional computers. So a claim, like Shor’s, that there was an algorithm that could factor composite integers in polynomial time seemed very unlikely to be correct at first sight. Worse still, this new algorithm was written in an arcane language of quantum mechanical states and operators—not the terminology NSA-mathematicians were accustomed to seeing.

As the weeks went by, however, and appropriate experts were brought in. It became apparent that Shor’s proposal had substance, and various government workshops were organized to consult with more quantum expertise. Jon Dowling, a charismatic early evangelist quantum computing, says he could identify the “spooks” at one of these early meetings because they were the only people with no affiliation printed on their name tags, listened attentively, said nothing, and scribbled notes in standard government issue green-cover notebooks. Later I was told the NSA had a hard time assigning someone to assess Shor’s algorithm because they had few staff at that time conversant with the quantum mechanical concepts needed

to understand it. Nevertheless, to their credit, they did so, and, moreover, quickly realized that they needed to know whether it was possible for anyone to build such a factoring engine. That single-minded objective became the rallying cry around which most U.S. funding for quantum computing has since been directed.

6.1 Code-Making and Code-Breaking

Cryptography, the science of making and breaking coded messages, is undoubtedly one of the oldest expressions of human mathematical ingenuity. The first known coded text is a small piece of Babylonian cuneiform writing dating from around 1500 BC. It describes a secret method for making a glaze for pottery. By recording the instructions in a code, the potter could preserve his secret recipe whilst denying its details to jealous competitors.

The ancient Greeks used a code-making scheme known as the scytale to send messages via human couriers across insecure territories. The scytale consisted of long thin strip of cloth and a matching pair of irregularly tapered sticks. One stick was in the possession of the sender and the other was in possession of the intended recipient prior to any message being sent. Given this pre-agreed arrangement, to encode a message, a blank cloth strip was wrapped around the senders' stick and the message written in vertical columns. When the cloth was unwound and removed from the stick the letters comprising the message became permuted by an amount determined by the contours of the irregularly tapered stick on which the cloth had been wrapped. Thus, anyone intercepting the cloth, but who did not possess the appropriately tapered stick, would be unable to read the message. But when the intended recipient, who already possessed the appropriately tapered stick, received the cloth from the courier he could decipher the message simply by wrapping it around his stick and reading the letters in vertical columns.

A more algorithmic approach was adopted by the Roman emperor Julius Caesar. Caesar is known to have used a transposition code in which each character in a message was displaced four characters ahead in the alphabet. Thus the message “Brutus might betray me” would have been encrypted as the text string “Fvyxyw qmklx fixvec og”.

With the advent of modern computers, cryptosystems have become significantly more sophisticated and complex. Indeed modern e-commerce routinely uses encryption to protect sensitive information and financial transactions as they fly across the Internet. Nowadays, cryptosystems do more than merely protect our confidential information, however. They are used to authenticate our identity, ensure the integrity of the data we transfer over public channels, and commit us to making our transactions.

Cryptography remains a game of cat and mouse between code-makers and code-breakers. As fast as one group creates codes the other tries to break them. On several occasions cryptographers had thought they had invented unbreakable sometimes turn out to fall short of this, in practice, due to subtle holes in their security proofs, imperfect implementation or enforcement of the ideal protocol, or outright blackmail or intimidation of the users.

6.1.1 Code-Breaking: The Enigma Code and Alan Turing

The modern era of code breaking dates from the Second World War when encrypted messages became widely used for conveying secret message over radio broadcasts. Since such broadcasts could be intercepted it became necessary to encrypt the communications.

In more modern times machines have been developed to encode and decode messages using sophisticated new codes. The Enigma machine was invented around 1918, by Arthur Scherbius who promoted its use for secure banking communications. It was described in a patent in 1919 and was adopted by the German Navy in 1926, the German Army in 1928 and the German Air Force in 1935. It was withdrawn from public use once the German Navy adopted it but it had already become known worldwide, throughout the cryptographic community. It was certainly known to the British government as there is a record of a meeting at which it was determined that the Enigma machine was not suitable for military use.

The Germans thought otherwise, however. In the Second World War, they used the Enigma machine and an even more elaborate Lorenz machine to pass orders between the German High Command and officers in the field. The Enigma code was used for communications with German submarines called U-boats, and the Lorenz code, known as the “Fish” code to the British code-breakers, was used for communications amongst the higher echelons of the German Army. Three Polish mathematicians, working for the Polish government first cracked the Enigma-encoded messages of the German military in 1930. However, the ability was lost again when the Germans used more sophisticated machines.

The breaking of Enigma and Lorenz-encrypted coded messages quickly became a military priority. The British and the Americans both established code-breaking centers staffed by some of the best minds of their generation. In Britain, the center was located within a manor house known as Bletchley Park. These elite teams were not composed exclusively of mathematicians, but rather by bright people with a knack for solving puzzles. For example, at Bletchley Park, Dilly Knox, an eminent Greek scholar, unscrambled Enigma encoded messages of the Italian navy once or twice during the Spanish Civil War. Bill Tute, at the time a biologist, reconstructed a Lorenz machine in 1942/1943, that enabled him to tackle some of the Fish codes. Despite the skill of these individuals, the need to use a machine to break the codes quickly became apparent.

A machine that reads in a coded message and unscrambles it by applying some systematic algorithm is nothing other than a Turing machine. So it is not surprising that Alan Turing was enlisted into the British code breaking effort during the Second World War. Turing joined the Code and Cipher School at Bletchley Park. The first code-breaking machines were called Turing Bombs. The Turing Bombs were electro-mechanical contraptions built to design of Alan Turing with contributions from others such as Gordon Welchman, a Cambridge mathematician. They employed relays, parts of tabulating machines and special rotors and weighed about one ton each. They were programmed by means of plug boards and first started breaking the Enigma codes in the summer 1940. In all, 210 Turing Bombs were

built in UK and 200 in USA by end of War. They were extremely effective, breaking a total of 186 different keys during the War, some virtually every day.

By 1943 onwards, the British and Americans were reading an average of 2000–3000 secret German messages a day. Turing Bombes took from 14 to 55 hours to break each Enigma coded message, less when the Germans were found to be re-using certain keys. Speed was crucial. A message had to be decoded before the event threatened in the message took place. The old Turing Bombes were simply too slow to get the job done in time. Consequently, British Intelligence commissioned the construction of an electronic code-breaking machine that was based on vacuum tubes, which are much faster than electro-mechanical relays. At the time, there was no computer industry as such, so the British had to make do with the best they had—the Post Office, which also controlled the telephone system. They had more experience in the use of vacuum tube technology than anyone else.

The first vacuum tube code breaking machine used at Bletchley Park was called the “Heath-Robinson” because of its outlandish design. The Heath-Robinson used two tape loops that had to be kept synchronized and broke frequently. They were never used operationally but demonstrated that a vacuum tube based “computer” could break the encrypted messages, such as the “Fish codes” produced by the Lorenz machines. The real breakthrough for unscrambling the Fish codes came with the introduction of the “Colossus” computer that employed some 3500 vacuum tubes. In all, 13 Colossus machines were built. Typically, they could crack a Fish code in one or two days but once succeeded in doing so in 20 minutes.

On the 12th November 1940, at the height of the Second World War, British military intelligence intercepted an Enigma-encoded message from the German high command to the commander of the Luftwaffe. The Germans had no idea that the Allies could break the Enigma code and were remarkably explicit in communicating their true intentions in their coded messages. This particular message outlined a plan for a bombing raid on 14th November on the city of Coventry, an industrial center in the middle of England. Unfortunately, the British code-breakers misinterpreted the reference to Coventry and instead warned Winston Churchill, the British Prime Minister, that the raid was probably to be on London but there was a possibility that it could be Coventry or Birmingham. At 3:00 pm on 14th November the Radio Countermeasures team, a group independent of the cryptographic team, detected intersecting directional radio beams over the city of Coventry confirming that Coventry was indeed the target.

It is not entirely clear what happened next. It is most likely that in the fog of war the message was not relayed to a high enough authority to be acted upon effectively. Less likely, some have speculated that Winston Churchill or other senior British officers deliberately failed to alert the people of Coventry of the impending raid so that the Germans would not suspect that the Enigma code had been broken. If Churchill had known the target for the air raid he would not have been likely to announce it publicly but would, instead, have stepped up the anti-aircraft defenses around Coventry. Curiously, the ack-ack defenses were increased around Coventry on the night of the 14th November but this might simply have been in response to the possibility of the raid being on Coventry. The ensuing air raid on Coventry killed

hundreds of people but probably many more lives were saved from the knowledge gleaned from intercepted German messages in the months that followed.

6.2 Public Key Cryptosystems

Although symmetric private key cryptosystems, such as the OTP, are highly secure provided the keys are kept secret and not re-used, they are also highly impractical. This is because all symmetric private key cryptosystems require the sender and receiver to agree upon matching keys prior to the commencement of their secure communications. Worse still, as these symmetric private key cryptosystems are enacted, they typically consume their key material at a voracious rate. These two properties make the symmetric private key systems quite impractical in a world of classical communications.

For decades cryptographers sought a more practical protocol that would enable parties to communicate securely without the need for a pre-arranged shared private key required by the One Time Pad cryptosystem. Breakthroughs were made in 1976 when Diffie and Hellman [144], and in 1978 when Rivest, Shamir, and Adleman [419] and Merkle [349], invented different versions of *public* key cryptosystems. In truth, the breakthrough appears to have occurred even earlier at the British military intelligence establishment GCHQ, but it was hushed-up by the authorities for several decades. Today all secure internet transactions use some form of public key cryptosystem to ensure the confidentiality and integrity of sensitive data as it zips across the internet for all to see. These public schemes work along the same lines as a safe with two keys—one public and the other private. The public key can be given out freely, and anyone wanting to send a message locks it (securely) in the safe using this public key. But only the legitimate recipient can open the locked safe using the matching private key. Rather than these keys being physical hardware, in the public key cryptography schemes they are the products of mathematical operations that are easy to compute in one direction, but intractably hard to compute in the other. In particular, the Diffie-Hellman scheme relies upon the exponentiation (easy)/discrete-logarithm (hard) problems to make the key pairs, whereas RSA relies upon multiplication (easy)/factoring (hard) problems to make its key pairs. So long as the operations needed to obtain the private key from purely publicly available information are computationally intractable, these public key schemes remain secure.

6.2.1 The RSA Public-Key Cryptosystem

One of the public key cryptosystem of most interest to us is the so-called RSA system invented by Ronald Rivest, Adi Shamir, and Leonard Adleman in 1978 [419].

The RSA cryptosystem solves the key distribution problem. Unlike the one-time pad scheme, in RSA the sender and recipient do not need to meet beforehand to

exchange secret keys. Instead, the sender and receiver use *different* keys to encrypt and decrypt a message. This makes it significantly more practical than the one-time pad scheme.

The basic idea is as follows. A person wishing to receive secret messages, using RSA, creates his own pair of keys, consisting of a *public* key and a *private* key. He makes the public key known but keeps the private key hidden. When someone wants to send him a secret message, the sender obtains the public key of the intended recipient and uses it to encrypt his message. Upon receiving the scrambled message, the recipient uses his private key to decrypt the message. The trick is to understand how the public key and private key need to be related to make the scheme work in an efficient, yet secure, fashion.

To be an *efficient* cryptographic scheme, it must be easy for a sender to compute E , the encryption of the plaintext message M given the public key \$PublicKey. In other words, the computation

$$E = \text{Encrypt} [\text{MessageToIntegers} [M, \$PublicKey]] \quad (6.1)$$

must be simple. Moreover, it must also be easy for the intended recipient to decrypt an encrypted message given the private key \$PrivateKey. That is, the computation

$$M = \text{IntegersToMessage} [\text{Decrypt} [E, \$PrivateKey]] \quad (6.2)$$

must be simple too. Last but not least it must be computationally easy to generate the required public key/private key pairs.

$$\text{Alice encrypts: } \text{Encrypt} [\text{plaintext}, k_{\text{public}}] = \text{ciphertext} \quad (6.3)$$

$$\text{Bob decrypts: } \text{Decrypt} [\text{ciphertext}, k_{\text{private}}] = \text{plaintext} \quad (6.4)$$

To be a *secure* cryptographic scheme, it must be extremely difficult to determine the message M given only knowledge of E and the public key \$PublicKey. Also, it must be extremely difficult to guess the correct key pair. Such a dual-key scheme is called a *public key cryptosystem* [528]. It is possible to have different cryptosystems by choosing different mathematical functions for creating the key pairs or by choosing different encryption or decryption procedures, e.g., elliptic curve cryptosystems.¹

The RSA system is just such a cryptosystem. It relies on the *presumed* difficulty of factoring large integers on a classical computer. In an effort to assure people that factoring was indeed a sound basis on which to risk the security of a cryptosystem, in 1991 RSA Laboratories announced set of grand challenge factoring problems that were believed to be difficult to solve using the computers of the day. The results of this competition are summarized in Table 6.1. In the early years the integers to be factored were given names such as “RSA-100” in which the number in the name

¹Curiously, the quantum algorithm that breaks the RSA public key cryptosystem can be modified slightly to break the elliptic curve cryptosystem too. We will have more to say on this later.

Table 6.1 Table showing the number of MIPS-years of effort needed to factor progressively larger integers. A “MIPS-year” is the number of steps processed in one year at a rate of one million instructions per second. The table shows the name of the integer factored, the size of that integer in base-10 and base-2 notation, the magnitude of the cash prize offered for factoring it, the date it was factored (if ever) and the MIPS-years of computational effort that were required

Number	Number of decimal digits	Number of binary digits	Cash prize	First factored	MIPS years
Typical	45	—	—	1974	0.01
Typical	71	—	—	1984	0.1
RSA-100	100	—	—	Apr. 1991	
RSA-110	110	—	—	Apr. 1992	
RSA-120	120	399	—	Jun. 1993	825
RSA-129	129	429	\$100	Apr. 1994	5000
RSA-130	130	432	—	Apr. 1996	750
RSA-140	140	466	—	Feb. 1999	2000
RSA-150	150	—	—	Apr. 2004	
RSA-155	155	512	—	Aug. 1999	8000
RSA-160	160	—	—	Apr. 2003	
RSA-576	174	—	\$10,000	Dec. 2003	
RSA-640	193	640	\$20,000	Nov. 2005	
RSA-200	200	—	—	May 2005	
RSA-704	212	704	\$30,000	Unsolved	
RSA-768	232	768	\$50,000	Unsolved	
RSA-896	270	896	\$75,000	Unsolved	
RSA-1024	309	1024	\$100,000	Unsolved	
RSA-1536	463	1536	\$150,000	Unsolved	
RSA-2048	617	2048	\$200,000	Unsolved	

referred to how many *decimal* digits they contained. This convention was changed with the introduction of “RSA-576” and thereafter the number in the name referred to how many *binary* digits were in the integer to be factored. The last cash prizes paid out were for factoring RSA-576 and RSA-640. The latter required approximately 30 CPU-years of computational effort (for a 2.2 GHz Opteron processor) over 5 months of calendar time [445]. This is actually less than the 55 CPU-years (on a 2.2 GHz Opteron processor) and 3 months of calendar time that was required to factor RSA-200 [444]. These results tend to support the belief that factoring is indeed a hard computational problem even with modern computer hardware. However, the RSA factoring competition was closed in 2007, and the outstanding prizes may no longer be claimed, leaving several of the grand challenge factoring problems remaining unsolved to this day.

A cryptosystem whose security is based on the presumption that factoring is hard works as follows. Suppose Alice wants to receive secret messages from other peo-

ple. To create a public key/private key pair, Alice picks two large prime numbers, p and q , and computes their product, $n = pq$. She then finds two special integers, d and e , that are related to p and q . The integer d can be chosen to be any integer such that the largest integer that divides both d and $(p - 1)(q - 1)$ exactly (i.e. with zero remainder) is 1. When this is the case d is said to be “co-prime” to $(p - 1)(q - 1)$. The integer e is picked in such a way that the remainder after dividing ed by $(p - 1)(q - 1)$ is 1. When this relationship holds, e is said to be the modular inverse of d . Alice uses these special integers to create a public key consisting of the pair of numbers (e, n) and a private key consisting of the pair of numbers (d, n) . Alice broadcasts her public key but keeps her private key hidden.

Now suppose Bob wishes to send Alice a secret message. Even though Bob and Alice have not conspired beforehand to exchange key pads, Bob can still send a message to Alice that only she can unscramble. To do so, Bob looks up the public key that Alice has advertised and represents his text message M_{text} as a sequence of integers in the range 1 to n . Let us call these message integers M_{integers} . Now Bob creates his encrypted message E by applying the rule:

$$E_i = M_i^e \bmod n \quad (6.5)$$

(i.e., raise the i -th message integer to the power e , divide the result by n and keep the remainder) for each of the integers M_i in the list of message integers M_{integers} .

Upon receipt of these integers, Alice decrypts the message using the rule:

$$M_i = E_i^d \bmod n \quad (6.6)$$

The final step is then to reconvert the message integers to the corresponding text characters. Thus the RSA cryptosystem can be summarized as follows:

Algorithm for the RSA Public Key Cryptosystem

1. Find two large primes p and q and compute their product $n = pq$.
2. Find an integer d that is co-prime to $(p - 1)(q - 1)$.
3. Compute e from $ed \equiv 1 \pmod{(p - 1)(q - 1)}$.
4. Broadcast the public key, i.e., the pair of numbers (e, n) .
5. Represent the message to be transmitted, M_{text} , say, as a sequence of integers, $\{M_i\}_{i=1}^n$.
6. Encrypt each M_i using the public key by applying the rule

$$E_i = M_i^e \bmod n$$

7. The receiver decrypts the message using the rule

$$M_i = E_i^d \bmod n$$

8. Reconvert the $\{M_i\}_{i=1}^n$ into the original message M_{text} .

6.2.2 Example of the RSA Cryptosystem

It is instructive to follow through the steps of the RSA algorithm. We will use atypically small numbers to make it easier to verify the steps in the algorithm. In practice, to have a secure system one would need to use numbers containing hundreds of digits.

Let's suppose Alice wishes to tell Bob her PIN number for her bank ATM machine. So Alice's plaintext message is "My PIN number is 1234". She trusts Bob with her PIN number but not other people, so she wants to maintain the confidentiality of her message to Bob by encrypting it using the RSA public-key cryptosystem.

Her first task is to convert the message (a string of characters) into an equivalent sequence of "message integers." The standard ASCII encoding associates, for each character, an integer in the range 0 to 255. Alice can use the ASCII codes plus 100 to guarantee that each encoded character has a 3-digit code. Note that such an encoding amounts to nothing more than a simple substitution cipher for the plaintext. This in itself confers little security. The main purpose of converting the plaintext into corresponding sequence of integers is to prepare the way for encrypting the plaintext by performing mathematical operations on these message integers. Here, then, is one way Alice can map her plaintext into a sequence of message integers:

Example: Prelude to Encryption—Converting a Message to Integers

1. Partition the string "My PIN number is 1234" into its individual characters giving "M, y, , P, I, N, , n, u, m, b, e, r, , i, s, , 1, 2, 3, 4". Note that blank spaces and punctuation marks are also regarded as characters and they have unique ASCII codes too.
2. Map each character into its ASCII equivalent giving "77, 121, 32, 80, 73, 78, 32, 110, 117, 109, 98, 101, 114, 32, 105, 115, 32, 49, 50, 51, 52".
3. Increment each such integer by 100 to ensure all the integers have exactly three digits, giving "177, 221, 132, 180, 173, 178, 132, 210, 217, 209, 198, 201, 214, 132, 205, 215, 132, 149, 150, 151, 152".
4. We can regard these integers as our "message integers" or we can re-group contiguous sequences of them into larger blocks, and treat those blocks as our message integers. For example, in groups of 6 they would be "177221, 132180, 173178, 132210, 217209, 198201, 214132, 205215, 132149, 150151, 152" and we could just as well call these our message integers. The blocking does not really matter. So long as our cryptosystem can reproduce the digits in each block, we can reproduce the digit triplets, and hence the ASCII integers, and hence the plaintext message.

This concludes the discussion of how message strings can be converted into integers. Now let's run the RSA algorithm on these message integers.

Example of How to Use the RSA Cryptosystem

1. Find two "large" primes p and q and compute their product $n = pq$. Alice picks $p = 659$ and $q = 541$ and their product is $n = pq = 356519$.

2. Find an integer d that is co-prime to $(p - 1)(q - 1)$. Alice finds $d = 182257$, which we can verify is co-prime to $(p - 1)(q - 1)$ by noting that $\gcd((659 - 1) \times (541 - 1), 182257) = 1$. Thus Alice's private key is the pair $(d, n) = (182257, 356519)$.
3. Compute e from $ed \equiv 1 \pmod{(p - 1)(q - 1)}$. Alice finds $e = 79033$. Thus her public key is the pair $(e, n) = (79033, 356519)$.
4. Broadcast the public key, i.e., the pair or numbers (e, n) , but keep the private key (d, n) secret.
5. Represent the message to be transmitted, M_{text} , say, as a sequence of integers, $\{M_i\}_{i=1}^n$. As shown above, Alice and Bob agreed upon a character encoding that mapped her plaintext "My PIN number is 1234" into the sequence "177221, 132180, 173178, 132210, 217209, 198201, 214132, 205215, 132149, 150151, 152".
6. Encrypt each M_i using the public key by applying the rule

$$E_i = M_i^e \pmod{n}$$

Applying this rule gives Alice the ciphertext "4253, 222477, 99943, 139852, 141469, 321340, 239261, 307414, 42925, 9406, 8973".

7. The receiver (Bob), looks up Alice's public key (so he learns (e, n)), and decrypts Alice's message using the rule

$$M_i = E_i^d \pmod{n}$$

This allows Bob to recover the message integers "177221, 132180, 173178, 132210, 217209, 198201, 214132, 205215, 132149, 150151, 152".

8. Reconvert the $\{M_i\}_{i=1}^n$ into the original message M_{text} . That is, from the message integers, Bob recovers the integer triplets and hence the plaintext that Alice intended to send. "My PIN number is 1234".

What makes RSA so useful is not merely the fact that there is an algorithm by which messages can be encrypted and decrypted but rather that the algorithm can be computed *efficiently*. Speed is vital if a cryptosystem is to provide a viable means of secure communication. Fortunately, each step in the RSA procedure can be done quickly. It is not immediately obvious that the calculations needed to find the pair of large prime numbers p and q , and the special integers d and e can all be done efficiently. However, it turns out they can [528]. Thus every step that the RSA procedure can be computed efficiently making it a viable cryptosystem overall.

Does the ease of the computations underlying RSA mean that RSA is vulnerable to attack? To answer this, let us take a look at what an adversary would have to do to crack RSA encoded messages.

6.3 Shor's Factoring Algorithm for Breaking RSA Quantumly

The essential trick to breaking the RSA public key cryptosystem is a method for factoring composite integers (i.e., integers that are the product of two large primes)

efficiently. Currently, the Number Field Sieve (NFS) is the preferred algorithm for factoring such composite integers if they have more than 100 digits. It's run time complexity scales as:

$$\mathcal{O}(e^{(c+o(1))(\log n)^{1/3}(\log \log n)^{2/3}}) \quad (6.7)$$

which is superpolynomial. Thus, factoring remains a very difficult computational problem to solve today, although it is not as hard as many other problems in the **NP** class.

Shor's algorithm provides an ingenious new way to factor composite integers in a time that grows only polynomially in the size of the number being factored. The general structure of Shor's algorithm can be inferred from the structure of the quantum circuit which implements it. This circuit is shown in Fig. 6.1.

Shor's Quantum Algorithm for Factoring Composite Integers

1. Pick a number $q = 2^{t_A}$ such that $N^2 \leq q < 2N^2$.
2. Pick a random integer x that is co-prime to N . That is, pick an integer x such that x and N have no common factors other than 1, i.e., $\gcd(x, N) = 1$.
3. Repeat steps (4) through (10) $\mathcal{O}(\log q)$ times, *using the same value for x each time*.
4. Initialize Register A (having t_A qubits) to be $|0\rangle$ and Register B (having t_B qubits) to be $|0\rangle$, i.e.,

$$|\psi_0\rangle = |0\rangle_A |0\rangle_B \quad (6.8)$$

5. Apply a Hadamard gate to each qubit in Register A, i.e.

$$|\psi_1\rangle = (H \otimes H \otimes \cdots \otimes H)|0\rangle_A \otimes |0\rangle_B \quad (6.9)$$

$$= \frac{1}{\sqrt{2^{t_A}}} \sum_{j=0}^{2^{t_A}-1} |j\rangle_A |0\rangle_B \quad (6.10)$$

Think of this as placing Register A in an equally weighted superposition of all the possible integer values it can contain, i.e., all integers in the range 0 to $2^{t_A} - 1$.

6. Now apply the transformation $U_x : |j\rangle_A |0\rangle_B \rightarrow |j\rangle_A |x^j \bmod N\rangle_B$ to state $|\psi_1\rangle$. The state of the complete register becomes:

$$\begin{aligned} |\psi_2\rangle &= U_x |\psi_1\rangle \\ &= \frac{1}{\sqrt{2^{t_A}}} \sum_{j=0}^{2^{t_A}-1} U_x |j\rangle_A |0\rangle_B \\ &= \frac{1}{\sqrt{2^{t_A}}} \sum_{j=0}^{2^{t_A}-1} |j\rangle_A |x^j \bmod N\rangle_B \end{aligned} \quad (6.11)$$

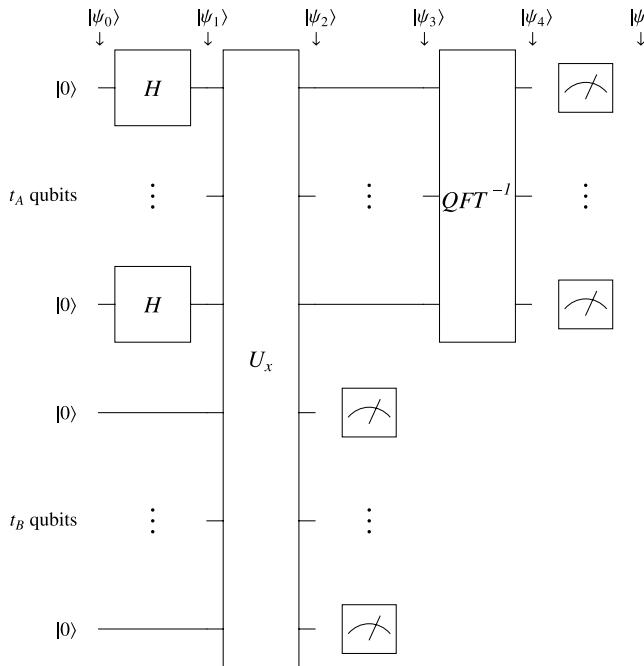


Fig. 6.1 Quantum circuit for Shor’s Algorithm. Register A has t_A qubits. Register B has t_B qubits. The state of the registers after each step of the computation is indicated by $|\psi_0\rangle, |\psi_1\rangle, \dots, |\psi_5\rangle$ left to right across the top of the circuit. Initially, both registers are set to $|0\rangle$, a set of t_A Walsh Hadamard gates are applied to Register A , thereby creating an equally weighted superposition of all the bit strings (i.e., different values of j) Register A can hold. Next a transformation is applied to Register B conditional on the value in Register A . Specifically, if Register A contains integer $|j\rangle$, and Register B contains $|0\rangle$, the transformation is $|j\rangle|0\rangle \rightarrow |j\rangle|x^j \bmod N\rangle$. Then Register B is read, projecting out a particular value of $x^j \bmod N$. Due to entanglement between the registers, Register A is thereby projected into a superposition of values of j consistent with the outcome in Register B . These j values are spaced r apart, where r is the sought after period. Hence, taking the inverse QFT of Register A , gives state that is sharply peaked in the vicinity of multiples of the inverse period. Thus, by reading Register A we obtain a string of bits corresponding to the binary representation of a number $k2^{t_A}/r$ where k, t_A and r are all integers, of which t_A is known. Hence, after a few repetitions we have enough samples of integer multiples of the inverse period to be able to guess r . The two factors of N may then be computed from $\gcd(x^{r/2-1}, N)$ and $\gcd(x^{r/2+1}, N)$.

7. Measure the state of Register B . This reveals a particular integer value for the contents of Register B , e.g., $|x^{b_0} \bmod N\rangle_B$ for some smallest value b_0 , and simultaneously projects the state of Register A into a superposition of just those values of $|j\rangle$ such that $x^j \bmod N = x^{b_0} \bmod N$. As these j values are separated from one another by an integer amount, namely the sought-after period r , they can be written in the form $|ar + b_0\rangle$. If r happens to be a power of 2, the superposition created as a side effect of measuring Register B can be written in

the form:

$$|\psi_3\rangle = \frac{1}{\sqrt{2^{\ell_A}/r}} \left(\sum_{a=0}^{\frac{2^{\ell_A}}{r}-1} |ar + b_0\rangle \right)_A |x^{b_0} \bmod N\rangle_B \quad (6.12)$$

In this case the values in Register A are now strongly peaked at multiples of r offset by the value b_0 , i.e., $|b_0\rangle$, $|r + b_0\rangle$, $|2r + b_0\rangle$, ..., thereby creating a periodically repeating pattern in Register A. However, if r is not a power of 2, we would replace the factor $\frac{2^{\ell_A}}{r}$ with the integer m_{b_0} representing the largest integer for which $(m_{b_0} - 1)r + b_0 \leq 2^{\ell_A} - 1$. In practice this means that the superposition in Register A will still be periodic with period r , but it will not contain a whole number of complete periods. In the remainder of the analysis we will assume r is a power of 2 as it is easier to see how the algorithm works in this case. The complications caused when r is not a power of 2 will be explained by example.

8. Next compute the inverse QFT of the projected state in Register A and do nothing (equivalent to applying the identity operator) to Register B.

$$\begin{aligned} |\psi_4\rangle &= (\text{QFT}^{-1} \otimes \mathbb{1}_{2^{\ell_B}})|\psi_3\rangle \\ &= \frac{1}{\sqrt{2^{\ell_A}/r}} \sum_{a=0}^{\frac{2^{\ell_A}}{r}-1} \left(\frac{1}{\sqrt{2^{\ell_A}}} \sum_{j=0}^{2^{\ell_A}-1} e^{-2\pi ij(ar+b_0)/2^{\ell_A}} |j\rangle \right)_A |x^{b_0} \bmod N\rangle_B \\ &= \frac{1}{\sqrt{r}} \left(\sum_{j=0}^{2^{\ell_A}-1} \left(\frac{1}{2^{\ell_A}/r} \sum_{a=0}^{\frac{2^{\ell_A}}{r}-1} e^{-2\pi i \frac{ja}{2^{\ell_A}/r}} \right) e^{-2\pi ij b_0 / 2^{\ell_A}} |j\rangle \right)_A |x^{b_0} \bmod N\rangle_B \\ &= \frac{1}{\sqrt{r}} \left(\sum_{k=0}^{r-1} e^{-2\pi i \frac{k}{r} b_0} \left| \frac{k 2^{\ell_A}}{r} \right\rangle \right)_A |x^{b_0} \bmod N\rangle_B \end{aligned} \quad (6.13)$$

where we have used the identity

$$\frac{1}{L} \sum_{a=0}^{L-1} \exp\left(-2\pi i \frac{ja}{L}\right) = \begin{cases} 1 & \text{if } j \text{ is an integer multiple of } L \\ 0 & \text{otherwise} \end{cases} \quad (6.14)$$

with $L = 2^{\ell_A}/r$ and $j = k \frac{2^{\ell_A}}{r}$ for $0 \leq k < r$, $k \in \mathbb{Z}$.

9. Measure Register A. This effectively samples from the inverse discrete Fourier transform of the periodic function that was in Register A just before the inverse QFT was applied. As can be seen from the final form for $|\psi_4\rangle$, in measuring Register A we will obtain a result that is of the form $|\frac{k 2^{\ell_A}}{r}\rangle$ for some unknown integer k ($0 \leq k \leq r - 1$).
10. By repeating the steps (4) through (9) $\mathcal{O}(\log q)$ (i.e., “polynomially many”) times, and when r happens to be a power of 2, we obtain a set of samples

from the inverse QFT of the periodic sequence contained in register A . That is, each time we run Shor's algorithm we find Register A in a state such as $|\frac{k_1 2^A}{r}\rangle$, $|\frac{k_2 2^A}{r}\rangle$, or $|\frac{k_3 2^A}{r}\rangle$ etc. where the integers k_i and the period r are all unknown. There are three cases to consider: (a) If we obtain $\frac{k_i}{r} = 0$ (i.e., $k_i = 0$), we cannot learn anything about r so we re-run Shor's algorithm; (b) If $k_i \neq 0$ and k_i is co-prime to r , the fraction $\frac{k_i}{r}$ cannot be further reduced so the denominator is in this case r ; (c) If $k_i \neq 0$ and k_i is not co-prime to r , then the rational number $\frac{k_i}{r}$ has a common factor. Canceling this common factor will yield a rational number having a denominator smaller than r . For example, if the period r happened to be 4 (say) and k_i happened to be 2, then the rational number $\frac{k_i}{r} = \frac{2}{4}$ would reduce to $\frac{1}{2}$ and the denominator would not be r . However, one could easily spot this by re-running Shor's algorithm just a handful of times to find, e.g., multiples of the inverse period to be $\frac{1}{4}, \frac{1}{2} (= \frac{2}{4}), \frac{3}{4}$ etc.

If r is not a power of 2, (6.13) is no longer strictly correct, although it is close to being correct. In this case, the integers that are the outputs from Shor's algorithm are only guaranteed to be *approximations* to integer multiples of $\frac{1}{r}$. That is, we will obtain integers such as $c_1 \approx \frac{k_1 2^A}{r}$, $c_2 \approx \frac{k_2 2^A}{r}$, $c_3 \approx \frac{k_3 2^A}{r}$, etc. To find r in this case we use the continued fraction trick explained in Sect. 6.3.1. In brief, this involves dividing each distinct integer obtained by reading Register A , c_1, c_2, c_3, \dots , by 2^A to obtain the rational approximations $\frac{c_1}{2^A} \approx \frac{k_1}{r}$, $\frac{c_2}{2^A} \approx \frac{k_2}{r}$, $\frac{c_3}{2^A} \approx \frac{k_3}{r}$ etc. Each such number $\frac{c_i}{2^A}$ can be written as a continued fraction. The sequence of rational numbers obtained by truncating this continued fraction expansion after progressively more terms define the convergents of $\frac{c_i}{2^A}$. The convergent of $\frac{c_i}{2^A}$ having the largest denominator less than n , the number being factored, is the exact integer multiple of the inverse period $\frac{k_i}{r}$. By seeing just a handful of examples of multiples of the inverse period $\frac{1}{r}, \frac{2}{r}, \frac{3}{r}$ etc. it is very easy to determine r .

11. Having obtained the period r of the contents of Register A , if r is odd, the algorithm has failed and must be re-run using a different value for x . However, if r is even the factors of n can be obtained from $\gcd(x^{r/2} - 1, N)$ and $\gcd(x^{r/2} + 1, N)$. Occasionally, the algorithm returns only the trivial factors of N , namely 1 and N , and must then be re-run using a different value of x .

6.3.1 The Continued Fraction Trick at the End of Shor's Algorithm

Curiously, I have found that the most puzzling part of Shor's algorithm for many students is the *classical* computation used to extract the period r from the samples of the inverse QFT obtained from Register A on successive runs of the algorithm. So let us take a look at this using a concrete example. To remind you, the basic strategy behind Shor's algorithm is to arrange for Register A to contain a periodic function

whose period is related to the factors of N (the composite number we wish to factor). The algorithm then uses quantum computing techniques to efficiently prepare a superposition containing this periodic function, and the inverse QFT to efficiently extract its period, r . Once r is known the rest of the algorithm proceeds classically by finding the factors of n from $\gcd(x^{r/2} - 1, N)$ and $\gcd(x^{r/2} + 1, N)$. The inverse QFT of a periodic function has sharp narrow spikes in the vicinity of integer multiples of the inverse period $\frac{1}{r}$, i.e. around $\frac{1}{r}, \frac{2}{r}, \frac{3}{r}$ etc. For certain periodic functions the spikes are exactly at integer multiples of the inverse period. But for other periodic functions the spikes have some noticeable (albeit small) width, which means when we sample there is some small probability of obtaining a value near (but not exactly at) the peak of the spike. In this more general case the integers obtained by reading Register A, i.e., the samples c_i of the inverse QFT of the periodic function, are only guaranteed to be *approximations* to integer multiples of the inverse period. That is, we have $c_i \approx \frac{k_i 2^A}{r}$ for unknown integers k_i and r . How then do we find k_i and r given knowledge of only c_i (the samples) and 2^A (the size of Register A)? This is where the continued fraction trick comes in. It relies on the fact that any real number ξ can be written as a *continued fraction expansion* as follows:

$$\xi = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4 + \ddots}}}} \quad (6.15)$$

where a_0, a_1, a_2, \dots are all positive integers. Such a continued fraction expansion is finite (i.e., terminates at some point) if ξ is rational. Otherwise, it is infinite (i.e., never terminates). We can find progressively better rational number approximations to ξ by truncating its continued fraction expansion after progressively more terms. The rational approximations formed by truncating the continued fraction expansion at successive terms are called the *convergents* of ξ . Thus, given the aforementioned continued fraction expansion of ξ its first few convergents will be:

$$\begin{aligned} \text{0th convergent of } \xi &\approx a_0 = \frac{a_0}{1} \\ \text{1st convergent of } \xi &\approx a_0 + \frac{1}{a_1} = \frac{a_1 a_0 + 1}{a_1} \\ \text{2nd convergent of } \xi &\approx a_0 + \frac{1}{a_1 + \frac{1}{a_2}} = \frac{a_2(a_1 a_0 + 1) + a_0}{a_2 a_1 + 1} \end{aligned} \quad (6.16)$$

$$\begin{aligned}
\text{3rd convergent of } \xi &\approx a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3}}} \\
&= \frac{a_3(a_2(a_1a_0 + 1) + a_0) + (a_1a_0 + 1)}{a_3(a_2a_1 + 1) + a_1}
\end{aligned}$$

By induction, and counting from $i = 0$, the i th convergent of ξ can therefore be written as:

$$i\text{th convergent of } \xi = \frac{\alpha_i}{\beta_i} \quad (6.17)$$

where

$$\begin{aligned}
\alpha_i &= a_i\alpha_{i-1} + \alpha_{i-2}, \quad \alpha_{-1} = 1, \quad \alpha_{-2} = 0 \\
\beta_i &= a_i\beta_{i-1} + \beta_{i-2}, \quad \beta_{-1} = 0, \quad \beta_{-2} = 1
\end{aligned} \quad (6.18)$$

Given the convergents of a continued fraction there is a theorem in number theory that states the following:

Convergent Approximation Theorem If there is a rational number $\frac{k}{r}$ (for integers k and r) and a real number ξ such that:

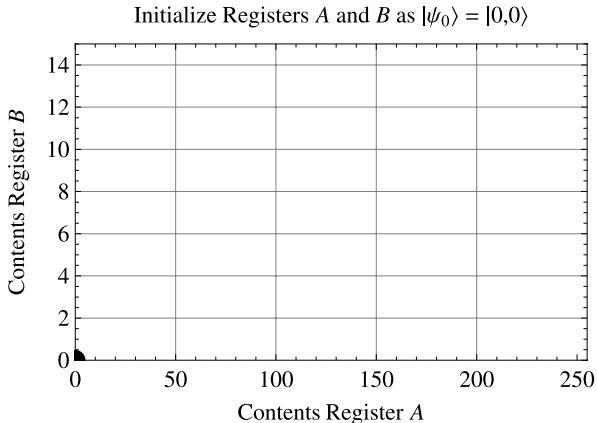
$$\left| \xi - \frac{k}{r} \right| \leq \frac{1}{2r^2} \quad (6.19)$$

then $\frac{k}{r}$ is a convergent of the continued fraction expansion of ξ . In the context of Shor's algorithm, the samples from the inverse QFT give us several distinct integers $c_i \approx \frac{k_i 2^A}{r}$. Knowing the size of Register A, i.e., 2^A , we can therefore form the rational numbers $\frac{c_i}{2^A} \approx \frac{k_i}{r}$. Thus, provided the approximation is close enough, i.e., provided $\left| \frac{c_i}{2^A} - \frac{k_i}{r} \right| \leq \frac{1}{2r^2}$, then the above theorem applies and $\frac{k_i}{r}$ is a convergent of the continued fraction expansion of $\frac{c_i}{2^A}$. Thus, to find k_i and r we compute the convergents of $\frac{c_i}{2^A}$ until we find that convergent having the largest denominator less than N , the integer we wish to factor. This convergent is the sought-after $\frac{k_i}{r}$. As a concrete example, suppose we are trying to factor $N = 21$ and have picked $q = 512 = 2^A = 2^9$ and $x = 10$. After running Shor's algorithm, the distinct nonzero integers we might obtain by sampling from the inverse QFT of Register A might be $c_i \in \{84, 85, 86, 169, 170, 171, 255, 256, 257, 340, 341, 342, 425, 426, 427\}$. We know that these output integers, c_i , are approximately integer multiples of $\frac{1}{r}$, specif-

ically, $\frac{c_i}{2^A} \approx \frac{k_i}{r}$. Computing, for each such c_i , the convergents of $\frac{c_i}{2^A}$ we obtain:

$$\begin{aligned}
 \text{convergents}\left(\frac{84}{2^9}\right) &= \left\{ 0, \left[\frac{1}{6} \right], \frac{10}{61}, \frac{21}{128} \right\} \\
 \text{convergents}\left(\frac{85}{2^9}\right) &= \left\{ 0, \left[\frac{1}{6} \right], \frac{42}{253}, \frac{85}{512} \right\} \\
 \text{convergents}\left(\frac{86}{2^9}\right) &= \left\{ 0, \frac{1}{5}, \left[\frac{1}{6} \right], \frac{21}{125}, \frac{43}{256} \right\} \\
 \text{convergents}\left(\frac{169}{2^9}\right) &= \left\{ 0, \left[\frac{1}{3} \right], \frac{33}{100}, \frac{34}{103}, \frac{169}{512} \right\} \\
 \text{convergents}\left(\frac{170}{2^9}\right) &= \left\{ 0, \left[\frac{1}{3} \right], \frac{85}{256} \right\} \\
 \text{convergents}\left(\frac{171}{2^9}\right) &= \left\{ 0, \frac{1}{2}, \left[\frac{1}{3} \right], \frac{171}{512} \right\} \\
 \text{convergents}\left(\frac{255}{2^9}\right) &= \left\{ 0, \left[\frac{1}{2} \right], \frac{127}{255}, \frac{255}{512} \right\} \\
 \text{convergents}\left(\frac{256}{2^9}\right) &= \left\{ 0, \left[\frac{1}{2} \right] \right\} \tag{6.20} \\
 \text{convergents}\left(\frac{257}{2^9}\right) &= \left\{ 0, 1, \left[\frac{1}{2} \right], \frac{128}{255}, \frac{257}{512} \right\} \\
 \text{convergents}\left(\frac{340}{2^9}\right) &= \left\{ 0, 1, \frac{1}{2}, \left[\frac{2}{3} \right], \frac{85}{128} \right\} \\
 \text{convergents}\left(\frac{341}{2^9}\right) &= \left\{ 0, 1, \frac{1}{2}, \left[\frac{2}{3} \right], \frac{341}{512} \right\} \\
 \text{convergents}\left(\frac{342}{2^9}\right) &= \left\{ 0, 1, \left[\frac{2}{3} \right], \frac{171}{256} \right\} \\
 \text{convergents}\left(\frac{425}{2^9}\right) &= \left\{ 0, 1, \frac{4}{5}, \left[\frac{5}{6} \right], \frac{39}{47}, \frac{44}{53}, \frac{127}{153}, \frac{425}{512} \right\} \\
 \text{convergents}\left(\frac{426}{2^9}\right) &= \left\{ 0, 1, \frac{4}{5}, \left[\frac{5}{6} \right], \frac{104}{125}, \frac{213}{256} \right\} \\
 \text{convergents}\left(\frac{427}{2^9}\right) &= \left\{ 0, 1, \left[\frac{5}{6} \right], \frac{211}{253}, \frac{427}{512} \right\}
 \end{aligned}$$

Fig. 6.2 Step 1: Load Register A and Register B with zeroes



Keeping, in each case, that convergent having the largest denominator less than $n = 21$ gives the integer multiples of the (unknown) inverse period as $\frac{1}{6}, \frac{1}{3} (= \frac{2}{6}), \frac{1}{2} (= \frac{3}{6}), \frac{2}{3} (= \frac{4}{6})$, and $\frac{5}{6}$. Hence, it is easy to see that the sought-after period is, in this case, $r = 6$. Hence, the continued fraction trick allows us to find the appropriate multiple of the inverse period.

6.3.2 Example Trace of Shor's Algorithm

Let's look at a trace of Shor's algorithm when it is being used to factor the number $N = 15$. In the following figures, we represent the contents of Register A in the horizontal direction and the contents of Register B in the vertical direction.

As we wish to factor $N = 15$, we begin by picking a value for x that is coprime to 15. This means we need to choose a value for x such that $\gcd(x, 15) = 1$. The value $x = 13$ works fine. Next we pick a value for q in the region $N^2 \leq q \leq 2N^2$. We pick $q = 256 = 2^8$. Therefore, Register A, which needs to hold all the possible integers from 0 to $q - 1$, need only have $t_A = 8$ qubits. Likewise, Register B, which needs to hold all the values in the periodic sequence $x^j \bmod N$ need only have $t_B = 4$ qubits because $13^j \bmod 15$ generates the sequence 1, 13, 4, 7, 1, 13, 4, 7, 1, 13, 4, 7, 1, 13, ... in which the largest number is 13 and this is expressible in 4 bits.

Initially we load Register A (8-qubits wide) and Register B (4-qubits wide) with zeroes, as indicated by the dot in bottom left corner of Fig. 6.2.

Next we load Register A with a superposition of all the possible integers in the range 0 to $q - 1$. This is represented as a long horizontal stripe in Fig. 6.3.

Next the gate U_x performs a different computation on Register B depending on the value of j in Register A. Specifically, if Register A is in state $|j\rangle$, then the value inserted into Register B is $U_x|j\rangle|0\rangle = |j\rangle|x^j \bmod N\rangle$. As Register A contains a superposition of many different j values, Register B becomes set a superposition of many different values of $x^j \bmod N$ too. For the case of $N = 15$ and

Fig. 6.3 Step 2: Load Register A with an equally weighted superposition of all the integers it can hold

$$\text{Load Register } A: |\psi_1\rangle = \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle |0\rangle$$

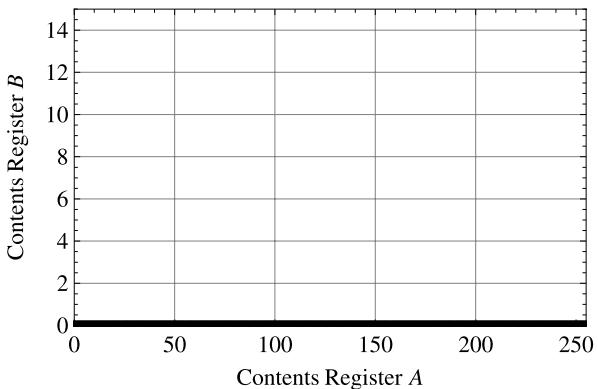
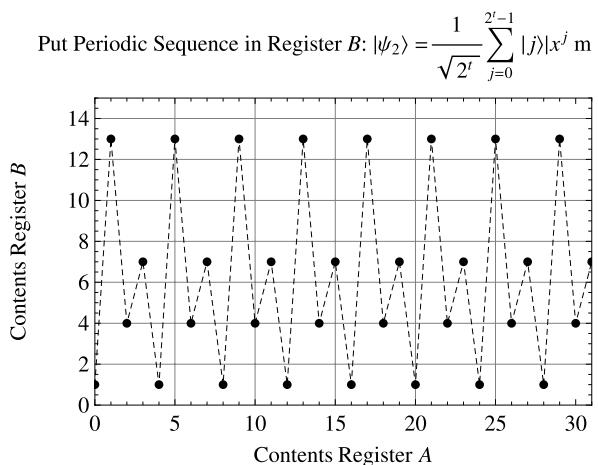


Fig. 6.4 Step 3: Compute, in quantum parallel, the modular exponential of $x^j \bmod N$ for each index j stored in Register A, and put the result in Register B



$x = 13$, the sequence of values in Register B is periodic as the j values increase monotonically. In fact, Register B will, as shown in Fig. 6.4, contain the superposition of states representing the sequence of values 1, 13, 4, 7, 1, 13, 4, 7, 1, 13, 4, 7, 1, 13, 4, 7, 1, 13, 4, 7, 1, 13, 4, 7, 1, 13, 4, 7, ... etc.

Next we read the contents of Register B. That is, we read the bit values of the four qubits in Register B and interpret the result as a base10 number that Register B is “holding”. Suppose the value we find is, as shown in Fig. 6.5, a “1” (corresponding to the four bit values $|0001\rangle$).

Due to there being entanglement between Registers A and B, the measurement we just performed on Register B has an instantaneous side effect on the contents of Register A. It projects Register A into an equally weighted superposition of all the

Fig. 6.5 Step 4: Read the bit values in Register B

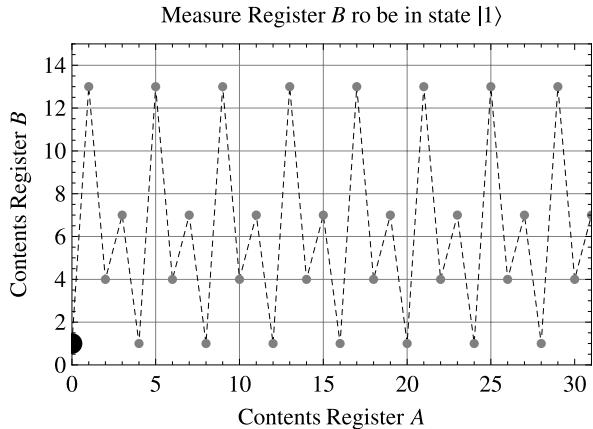
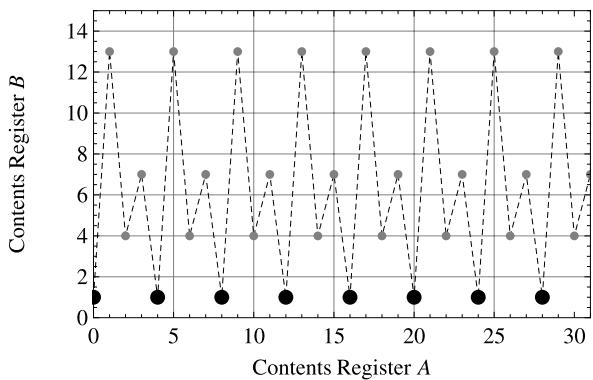


Fig. 6.6 Step 5: Register B is found to hold a particular integer “1”. Register A is projected into a superposition of j values for which $x^j \bmod N = 1$

$$\text{Project Register } A \text{ such that } |\psi_3\rangle = \frac{1}{\sqrt{2^t/r}} \sum_{a=0}^{(2^t/r)-1} |a r + b_0\rangle$$



j values such that $x^j \bmod N = 1$, i.e., the state $|0\rangle + |4\rangle + |8\rangle + |12\rangle + |16\rangle + \dots$. This is illustrated in Fig. 6.6.

Next we compute the inverse quantum Fourier transform (QFT $^{-1}$) of the contents of Register A . This produces a superposition in which, as shown in Fig. 6.7, the amplitudes are no longer equal, but instead are very strongly peaked around certain values. In fact the inverse Fourier transformed state has most of the amplitude concentrated in states that correspond to multiples of $1/r$, where r is the sought after period.

Upon observing the state of the Register A , we essentially sample from the inverse QFT and are likely to find a result corresponding to $k 2^t/r$ for some integer k .

Fig. 6.7 Step 6: Compute the QFT^{-1} of Register A

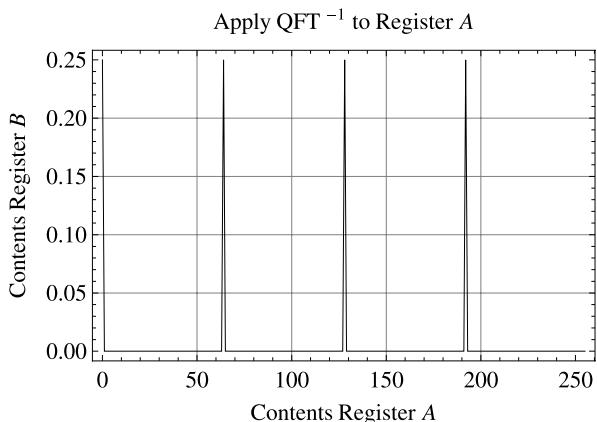
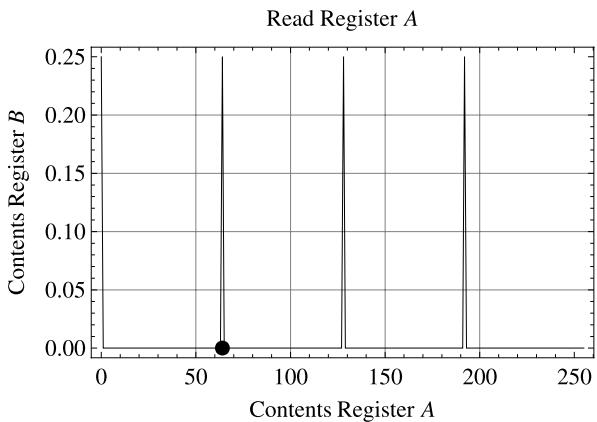


Fig. 6.8 Step 7: Read the bit values in Register A



Suppose we obtain the result “64”, as indicated in Fig. 6.8. We store this result and repeat the procedure all over again.

Multiple repetitions of the preceding steps might give us the following set of samples from the Fourier transform: $\{128, 64, 0, 192, 0, 128, 128, 64, 0, 192, 192, 64\}$. On those occasions when 0 is returned the algorithm yields no useful information and must be repeated. However, when nonzero integers are returned we know that each of these is an approximation to (or equal to) an integer multiple of the inverse period $\frac{1}{r}$. To find this period we follow the prescription given earlier and compute the convergents of $\frac{64}{2^8}$, $\frac{128}{2^8}$, and $\frac{192}{2^8}$, and find the multiples of the inverse period to be $\frac{1}{4}$, $\frac{1}{2} (= \frac{2}{4})$, and $\frac{3}{4}$. Hence, we deduce that the sought after period r is $r = 4$. We can then obtain the factors of 15 by computing $\gcd(x^{r/2} - 1, 15) = \gcd(13^2 - 1, 15) = 3$ and $\gcd(13^2 + 1, 15) = 5$. Thus, the factors of 15 are 3 and 5.

It is important to realize that Shor's algorithm is probabilistic: it does not always return a non-trivial factor. For example, if we wanted to factor $N = 15$, but we picked $x = 14$ (instead of $x = 13$) as the number co-prime to N , then

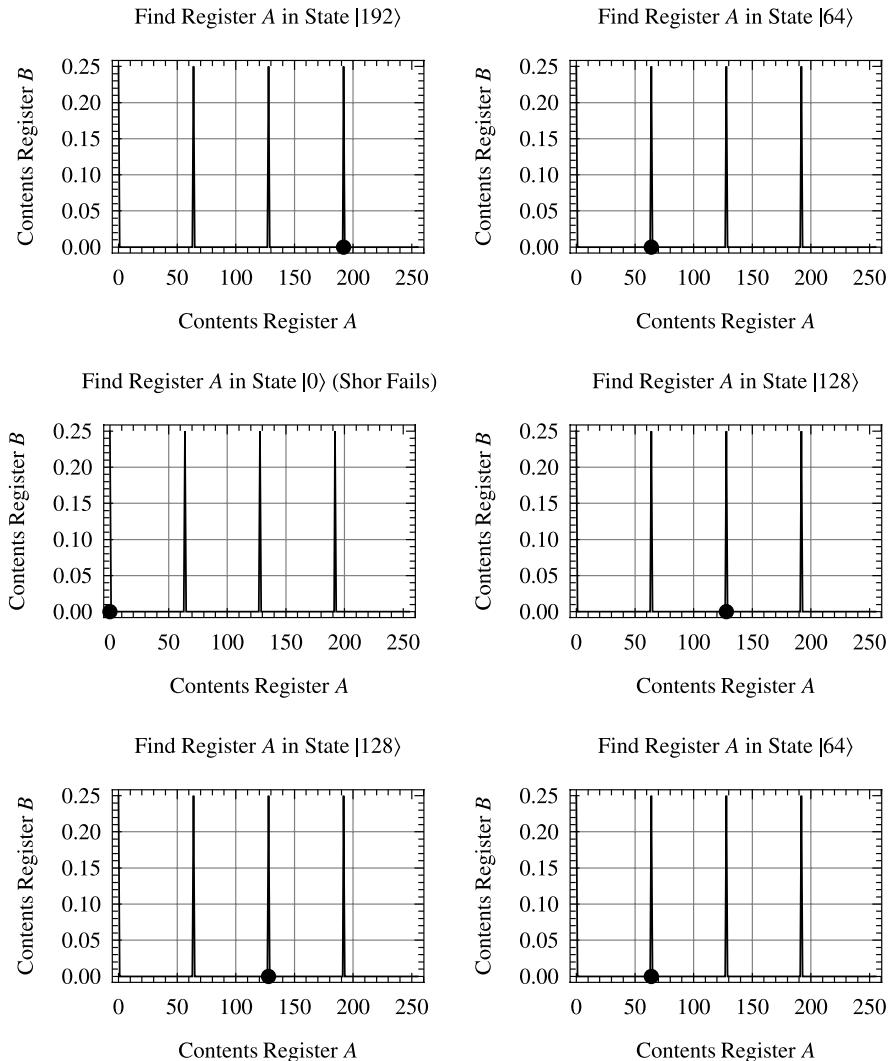


Fig. 6.9 Step 7: Read the bit values in Register A

the periodic sequence in Register B would be 1, 14, 1, 14, 1, 14, 1, 14, 1, 14, Hence we get a period $r = 2$. Unfortunately, when we compute $\gcd(14^{r/2} - 1, 15)$ we obtain $\gcd(13, 15) = 1$, and when we compute $\gcd(14^{r/2} + 1, 15)$ we obtain $\gcd(15, 15) = 15$. Hence, in this case, Shor's algorithm only yields the trivial divisors “1” and “15” and therefore fails. In such a circumstance we run Shor's algorithm again using a different value for x , which can be any number co-prime to N , the number being factored.

6.4 Breaking Elliptic Curve Cryptosystems with a Quantum Computer

Peter Shor's original paper contained quantum algorithms for order finding (and hence factoring composite integers) and computing discrete logarithms over finite groups. So far we have paid exclusive attention to the integer factorization problem, as this is the foundation on which the security of the RSA public-key cryptosystem rests. However, there are other public-key cryptographic protocols, such as the Diffie-Hellman key agreement scheme [143], the ElGamal encryption and signature schemes [170], and the Elliptic Curve Cryptosystem [294, 353] that rely for their security on the presumption that a different mathematical problem, namely *computing discrete logarithms*, is computationally intractable on classical computers. Discrete logarithms are similar to ordinary logarithms except that they work over *finite* fields of numbers instead of over real and complex numbers. Whereas the ordinary logarithm of b to base a , i.e., $\ell = \log_a b$ is the solution of the equation $b = a^\ell$ over the field of real or complex numbers, the discrete logarithm of β to base α is the solution of the equation $\beta = \alpha^\ell$ where α and β are restricted to be elements of a finite cyclic group G . For example, let \mathbb{Z}_p be the set of integers $\{0, 1, 2, \dots, p - 1\}$ where p is a prime number and all arithmetic operations (e.g., addition and multiplication) are performed modulo p . There is a mathematical theorem that *guarantees* that any member of \mathbb{Z}_p can be written as $\alpha^\ell \pmod{p}$. The discrete logarithm problem concerns the finding this power, ℓ . More formally, we have:

The Discrete Logarithm Problem Given a prime p , a generator α of a group, e.g., \mathbb{Z}_p , and a non-zero element $\beta \in \mathbb{Z}_p$, find the unique integer ℓ (for $0 \leq \ell \leq p - 2$), such that $\beta \equiv \alpha^\ell \pmod{p}$. The integer ℓ is the discrete logarithm of β to base α .

One can devise cryptosystems whose routine encoding and decoding steps rely on modular exponentiation (\pmod{p}) (which is easy), but which to break requires the computation of discrete logarithms (which is so hard as to be effectively intractable). Having functions that are easy to compute in one direction but effectively intractable to compute in the inverse direction make them a possible foundation for a public key cryptosystem. However, the actual complexity of solving a discrete logarithm problem depends on the choice of the underlying group over which the problem is defined. In the Diffie-Hellman and ElGamal schemes the groups used allow discrete logarithms to be computed in sub-exponential time, i.e., the same time as required for factoring a composite integer n , namely $\mathcal{O}(e^{c(\log n)^{1/3}(\log \log n)^{2/3}})$. However, if the underlying groups are taken to be elliptic curve groups on finite fields such as $\text{GF}(p)$ (p an odd prime) or $\text{GF}(2^m)$ (m an integer), the discrete logarithm problem is then especially difficult, requiring truly exponential time to find a solution. The essential ideas behind these elliptic curve groups are the following. An elliptic curve provides a geometric way of picturing the elements of a finite field of q elements, F_q , as points on a planar curve. In particular, let F_q denote a finite field of q elements, i.e., $\{0, 1, 2, \dots, q - 1\}$. Operations on the elements of F_q are to be computed modulo q , and always result in an element of F_q . In practical cryptosystems we usually take

$q = 2^m$ or $q = p$ where p is a large prime. An elliptic curve is then defined as the locus of points such that

$$y^2 \pmod{q} = x^3 + ax + b \pmod{q} \quad \text{s.t.} \quad 4a^3 + 27b^2 \pmod{q} \neq 0 \quad (6.21)$$

When a and b satisfy the above specified criterion, the locus of points induced by (6.21) defines an elliptic curve *whose points are closed under modular arithmetic operations*. That is, modular arithmetic operations (\pmod{q}) on the elements of F_q will always return elements in F_q . This means that modular *arithmetic* operations on the elements of F_q correspond to *geometric* operations on the points of the elliptic curve. For example, the “point addition” of point P on the elliptic curve and point Q on the elliptic curve yields another point $R = P \cdot Q$ on the elliptic curve. Geometrically, this point is obtained by projecting a straight line through points P and Q until it intersects the elliptic curve and then reflecting the intersection point across the x -axis. The result is a point R (on the elliptic curve) that is the “point addition” of points P and Q . If a point is added to itself, $R = P \cdot P$, we project the tangent of the point P until it intersects the elliptic curve and then reflect this intersection point across the x axis to yield $R = P \cdot P$. An elliptic curve augmented with such an “addition” operation creates a so-called elliptic curve *group* because it imbues the points on the elliptic curve with all the required characteristics of an abelian group (closure, associativity, zero element, inverse element, and commutativity). Once the addition operation is defined, whatever sequence of operations are performed the result is always one of the finitely many points on the elliptic curve. Thus, given an elliptic curve, a definition for the “addition”, i.e., “.”, of any pair of points on the elliptic curve, and a starting point, it is trivial to follow a prescribed sequence of additions to determine the final point reached. This is analogous to modular exponentiation and is easy. What is much harder is to find the number of iterations needed to carry a given starting point P into a given desired point Q via a sequence of self-additions, i.e., finding the integer ℓ (if it exists) such that $Q = P^\ell = P \cdot P \cdot \dots \cdot P$ (i.e., P added to itself ℓ times). This amounts to solving the elliptic curve discrete logarithm problem defined as follows:

The Elliptic Curve Discrete Logarithm Problem Given an elliptic curve E defined over a group F_q , a point $P \in E(F_q)$ of order n , and a point $Q \in E(F_q)$, determine the integer ℓ with $0 \leq \ell \leq n - 1$, such that $Q = P^\ell$, provided that such an integer exists. (Note: others write this in the notation “ $Q = \ell P$ ” but it means the same).

The realization that the elliptic curve discrete logarithm problem was so difficult led Neal Koblitz [294] and Victor Miller [353] to invent the concept of the Elliptic Curve Cryptosystem (ECC), independently, in the mid to late 1980s. To break ECC one would need to solve the socalled Elliptic Curve Discrete Logarithm Problem (ECDLP). Currently, the best known classical algorithm for computing the elliptic curve discrete logarithms is the Pollard rho-method [195, 395, 530], which has a complexity of $\mathcal{O}(\sqrt{\pi n}/2) \equiv \mathcal{O}(e^{\frac{1}{2}\log n + \frac{1}{2}\log \frac{\pi}{4}})$ serially and $\mathcal{O}(\sqrt{\pi n}/(2r))$ when

parallelized on r processors [489]. Using the elliptic curve groups on finite fields such as $\text{GF}(p)$ (p an odd prime) or $\text{GF}(2^m)$ (m integer) the discrete logarithm problem is especially difficult, requiring truly exponential time to find a solution. Thus, for the same key length the ECC cryptosystem provides a higher level of security than the RSA, Diffie-Helman and ElGamal cryptosystems. In fact, a mere 256-bit ECC key is about as hard to break as a 1024-bit RSA key [102]. Such is the confidence in the security of the Elliptic Curve Cryptosystem that it was recently approved by the U.S. National Security Agency (NSA) for encrypting information up to “Top Secret” level.² backslash before the underscore character Of most significance for quantum computing, as Shor’s algorithms for factoring integers and computing discrete logarithms *both* run in $\mathcal{O}(n^3)$ time, the speedup afforded by Shor’s algorithm in breaking ECC would be considerably *greater* than its speedup in breaking RSA, provided Shor’s discrete logarithm algorithm can be made to work over the groups $\text{GF}(p)$ (p an odd prime) or $\text{GF}(2^m)$ (m integer). This was indeed shown to be possible by John Proos and Christof Zalka for $\text{GF}(p)$ (p an odd prime) in [403] and by Phillip Kaye and Christof Zalka for $\text{GF}(2^m)$ (m integer) in [269]. The latter result was improved by Donny Cheung, Dmitri Maslov, Jimson Mathew, and Dhiraj Pradhan in [102] and [344] by showing that multiplicative inversion need only be used once, at the last stage of the algorithm, rather than at each operation on a point on the elliptic curve. Thus, quantum computers can speed up the breaking of Elliptic Curve Cryptosystems by more than they can speed up the breaking of RSA, Diffie-Helman and ElGamal cryptosystems, even though the ECC cryptosystem is classically “more secure”, for a given key length, than the other cryptosystems mentioned. Specifically, whereas Shor’s quantum algorithm moves a subexponential problem classically to a polynomial one quantumly, Proos and Zalka’s modified version for discrete logarithms, moves a truly exponential problem classically to a polynomial one quantumly. Moreover, a 160-bit ECC key can be broken on a 1000-qubit quantum computer, whereas a 1024-bit RSA key would require about a 2000-qubit quantum computer [403]. Thus, the *threshold* to implementing a quantum algorithm to compute elliptic curve discrete logarithms requires far fewer qubits than that needed to factor integers, and so it might be more feasible to build such a quantum computer.

6.5 Breaking DES with a Quantum Computer

In the 1970’s the dramatic expansion of electronic funds transfers and other e-transactions caused the U.S. government to realize the need for a new cryptographic standard to protect the integrity and confidentiality of such transactions. Accordingly, the U.S. National Bureau of Standards (NBS) solicited proposals for a new data encryption standard. The intent was to embed a dedicated chip within devices

²See NSA web page “The Case for Elliptic Curve Cryptography,” http://www.nsa.gov/business/programs/elliptic_curve.shtml

that exchanged confidential information so that a standard algorithm could be run quickly on data flowing into and out of such devices. Initially, the scheme the NBS selected was a simplification of one proposed by IBM called the Lucifer cipher. However, whereas the Lucifer cipher used 128 bit keys, to achieve acceptable speeds of operation, the NBS wanted to use only 64 bit keys and of these eight were discarded by the encryption algorithm. Thus the Data Encryption Standard (DES) was born using a key consisting of one out of effectively 2^{56} possibilities. Once the key was selected the DES algorithm operated deterministically encoding a given plaintext into a unique ciphertext. At the time the claim was made that “*Anyone buying cryptographic equipment which has been validated against the DES can be assured of a specific level of data security: namely that 2^{55} attempts and the use of the method of exhaustion are required to obtain any one key for the encryption algorithm used in DES*” [132]. Unfortunately, from the 1970’s onwards computer speed rose at such an astonishing rate the 56-bit DES scheme was quickly seen to be uncomfortably close to being breakable by the computers of the day, and the consensus was that, if adopted, it would only have a usable lifetime of about a decade. To break the DES scheme a code-breaker needs to obtain (or hypothesize) matching fragments of plaintext and ciphertext and then test whether different possible cryptographic keys, when plugged into the DES encoding algorithm, would map the plaintext into the ciphertext. Once the code-breaker determines the cryptographic key that works for this fragment of the ciphertext they can use it to unscramble the other parts of the intercepted message, or other messages, that were encrypted using the same key. The computation to try out all these key possibilities and thereby break DES is an unstructured search problem, which is **NP-Hard**. Hence, the DES scheme is secure provided $\mathbf{P} \neq \mathbf{NP}$ (which is still unproven although widely believed to hold) and provided the key length is sufficiently long that this code-breaking computation is effectively intractable for the computers of the day. Using a 56-bit key means there are 2^{56} key possibilities to try, so on average one would expect to have to test about half of these (2^{55} attempts) before hitting on the correct key. With the advent of quantum computing, however, there is an alternative approach to the trial and error search through the cryptographic keys. One could use Grover’s algorithm to find the sought after key in roughly the square root of the number of trials needed classically. The trick, first recognized by Gilles Brassard [71], is to think of the relationship between keys and ciphertexts as a virtual “quantum phone book” in which the keys corresponded to “names” and the ciphertexts corresponded to “phone numbers”. Given a known ciphertext (i.e., a “phone number”) and a known corresponding plaintext, our job is to find the DES key (i.e., the unique “name”) that correctly maps the plaintext into the ciphertext by following the steps of the DES encryption algorithm. Hence, the task of breaking DES corresponds to that of finding someone’s name in this virtual phone book given knowledge of only their telephone number, which is the familiar unstructured search problem. To assess the potential speedup one could obtain, imagine a sub-routine that tests whether a specific number is the secret key. A classical computer will need to call this sub-routine, on average, $\frac{1}{2}2^{56} \approx 3.6 \times 10^{10}$ million times before finding the right key! In contrast a quantum computer need only call the quantum version of this sub-routine, on

average, $\frac{1}{2} \times \frac{\pi}{4} \sqrt{2^{56}} \approx 105$ million times to find the right key. So although Grover's algorithm does not give an exponential speedup it could nevertheless be extremely useful.

6.6 Summary

Modern internet communications and electronic transactions rely heavily on the use of public key cryptosystems. Two famous examples are the RSA cryptosystem (RSA), and the elliptic curve cryptosystem (ECC). Such cryptosystems have the advantage that they do not require the sender and recipient of confidential messages to have met beforehand and exchanged secret key material. Instead, the person wishing to receive confidential messages creates a pair of matching public and private cryptographic keys, posts their public key for all to see, but keeps their private key secret. Anyone wishing to send the author of the public key a confidential message uses the posted public key to encrypt a message, and transmits the encrypted message via a potentially insecure classical communications channel. Upon receipt, the legitimate recipient uses his matching private key to unscramble the encrypted message.

The security of such public key cryptosystems relies upon the *presumption* that certain mathematical problems are intractably hard to solve. For example, the security of the RSA public key cryptosystem relies upon the presumption that factoring composite integers is intractable. Likewise, the security of the ECC public key cryptosystem relies upon the presumption that computing discrete logarithms is intractable. Both these presumptions appear to be valid if the code-breaker is only able to use a classical computer. Specifically, the best known classical algorithm for breaking RSA (the Number Field Sieve), has a running time that scales super-polynomially (but sub-exponentially) with increasing key length. A super-polynomial complexity scaling is regarded as intractable provided the key length is sufficiently great. Similarly, the best known classical algorithm for breaking ECC (the Pollard rho algorithm), has a running time that scales truly exponentially with increasing key length. This means, that for a given key length, an ECC-encrypted message is even harder to break than the corresponding RSA-encrypted message.

The situation changed in 1994, however, when Peter Shor published his polynomial time algorithm for factoring composite integers and computing discrete logarithms [455]. Thus, Shor's algorithm can break both RSA (via efficient integer factorization) and ECC (via efficient computation of discrete logarithms). The fact that quantum computers have the potential to break the types of public key cryptosystems used in email and electronic commerce was the impetus behind much of the funding for quantum computing, at least in the United States. One of the most striking, but under-reported, aspects of Shor's algorithm for factoring integers and solving the discrete logarithm problem, and Proos and Zalka's extension of the latter to the elliptic curve discrete logarithm problem, is that it shows the separation in complexity scaling between classical and quantum breaking of RSA is less than the complexity separation between classical and quantum breaking of

ECC. Whereas breaking RSA requires sub-exponential time classically, and polynomial time quantumly, breaking ECC requires truly exponential time classically and polynomial time quantumly. Hence, the advantage of the quantum computer is *even greater* in breaking ECC than in breaking RSA.

Other cryptosystems, such as a DES, are also impacted by the arrival of quantum computing, but in this case the speedup in code breaking is only polynomial, as it relies upon the use of Grover's algorithm to search through possible DES keys.

It is worth pointing out, however, that just as quantum mechanics takes away our security, by showing how to break supposedly “strong” classical cryptosystems, so too does it restore security by providing us with quantum cryptography—a fundamentally new approach to cryptography that is invulnerable to both quantum and classical attacks. We shall look at quantum cryptography in Chap. 13.

6.7 Exercises

6.1 Multiply the following numbers by hand:

- (a) Let $p = 12$ and $q = 34$. Find $N = p \times q$.
- (b) Let $p = 123$ and $q = 456$. Find $N = p \times q$.
- (c) Let $p = 1234$ and $q = 5678$. Find $N = p \times q$.

Now try factoring the following numbers by hand:

- (d) Let $403 = p \times q$. Find p and q .
- (e) Let $56,089 = p \times q$. Find p and q .
- (f) Let $7,006,619 = p \times q$. Find p and q .

Do you notice any change in difficulty between the multiplication problems and the factoring ones?

6.2 Table 6.1 shows the computational effort needed to factor various composite integers quoted in units of “MIPS-years”. 1 MIPS-year is the number of operations performed in one year by a single computer running at a rate of one million operations per second. If the running time of a Number Field Sieve factoring algorithm is “55 CPU-Years” when the CPU in question is a 2.2 GHz machine, what is the equivalent computational effort measured in MIPS-years? Use your calculations to fill in the missing “MIPS-year” data for factoring RSA-200 given in Table 6.1.

6.3 Given two integers, x and y , having greatest common divisor d , i.e., $d = \gcd(x, y)$, what is the least common multiple of x and y , i.e. $\text{lcm}(x, y)$?

6.4 Suppose you want to factor the number 15 using Shor's algorithm on a quantum computer.

- (a) Generate a random integer, $1 < x < 15$, that is co-prime to 15
- (b) Pick the number of qubits you will need for Register A and Register B. Explain why you picked this number
- (c) What is the period, r , of the sequence $x^0 \bmod 15, x^1 \bmod 15, x^2 \bmod 15, \dots$

- (d) Suppose you pick $x = 7$ and $n = 8$ qubits and you generate the superposition

$$|\psi\rangle = \frac{1}{\sqrt{256}} \sum_{i=0}^{255} |i\rangle |x^i \bmod 15\rangle \quad (6.22)$$

What are the first 15 terms of this superposition?

- (e) If you measure the second register, when the two registers are in state $|\psi\rangle$, and find it to be in state $|1\rangle$, what is the corresponding state of the (unread) first register?
- (f) How is the state of the first register then related to the period r ?
- (g) How is the period r related to the factors of 15?
- (h) What is the state of the first register after you apply the inverse QFT to it?

6.5 In Shor's algorithm the inverse QFT plays a pivotal role. Draw a quantum circuit for the inverse QFT.

6.6 Prove the shift invariance property of the n -qubit quantum Fourier transform, QFT_{2^n} . Specifically, define

$$|\psi\rangle = \sum_{j=0}^{2^n-1} c_j |j\rangle \quad (6.23)$$

and define a “shifted” version of this state as:

$$|\psi'\rangle = \sum_{j=0}^{2^n-1} c_j |j + \ell \bmod 2^n\rangle \quad (6.24)$$

Show that:

- (a) $\text{QFT}_{2^n}|\psi\rangle$ and $\text{QFT}_{2^n}|\psi'\rangle$ are the same up to an overall phase factor.
- (b) What is this phase factor?
- (c) How does this affect the probabilities with which we would obtain the outcome $|j\rangle$ if we measured the output from $\text{QFT}_{2^n}|\psi\rangle$ in comparison to measuring the outcome from $\text{QFT}_{2^n}|\psi'\rangle$?

6.7 Prove the convolution property of the n -qubit quantum Fourier transform, QFT_{2^n} . In classical computing, the “convolution” of two signals quantifies the degree to which they are similar. We can extend this notion to quantum states as follows. Suppose we have two n -qubit quantum states, which we can think of as encoding two “signals”, of the form:

$$|\psi\rangle = \sum_{k=0}^{2^n-1} c_k |k\rangle \quad (6.25)$$

$$|\varphi\rangle = \sum_{k=0}^{2^n-1} d_k |k\rangle \quad (6.26)$$

Following classical signal processing, we define the “convolution” of two such quantum states to be:

$$\text{convolution}(|\psi\rangle, |\varphi\rangle) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} \sum_{\ell=0}^{2^n-1} c_\ell d_{j-\ell} |j\rangle \quad (6.27)$$

and we follow the classical convention that when $j - \ell$ is negative we take $d_{j-\ell} = d_{2^n + j - \ell}$. Your task is to show that the QFT of the convolution of two such states is related to the QFTs of the states themselves. To see this, write the QFTs of $|\psi\rangle$ and $|\varphi\rangle$ in the form:

$$\begin{aligned} \text{QFT}_{2^n} |\psi\rangle &= \sum_{k=0}^{2^n-1} \alpha_k |k\rangle \\ \text{QFT}_{2^n} |\varphi\rangle &= \sum_{k=0}^{2^n-1} \beta_k |k\rangle \end{aligned} \quad (6.28)$$

and, based on these definitions, prove:

$$\text{QFT}_{2^n} \text{convolution}(|\psi\rangle, |\varphi\rangle) = \sum_{j=0}^{2^n-1} \alpha_j \beta_j |j\rangle \quad (6.29)$$

Note that it is not possible to devise a deterministic quantum circuit that actually computes the convolution of two such quantum states even though the mathematical relationship between the QFT of the convolution and the QFTs of the states being convolved still holds (see [323]). But approximate convolution is possible (see [121]).

6.8 What are the convergents of the rational numbers (a) $\frac{291}{2^9}$, (b) $\frac{365}{2^9}$, and (c) $\frac{438}{2^9}$? If the numbers 291, 365, and 438 had arisen as the output samples from Register A having run Shor’s algorithm to factor $n = 39$ what would be the period r ? Would this have been a successful or unsuccessful run of Shor’s algorithm? Explain your answer.

Chapter 7

Solving NP-Complete Problems with a Quantum Computer

*“If quantum states exhibit small nonlinearities during time evolution, then quantum computers can be used to solve **NP-Complete** problems in polynomial time [...] we would like to note that we believe that quantum mechanics is in all likelihood exactly linear, and that the above conclusions might be viewed most profitably as further evidence that this is indeed the case.”*

– Dan Abrams and Seth Lloyd¹

In computer science, a “decision problem” is a problem with a “yes” or “no” answer. Therefore, the question “Are there more than five prime numbers whose values are between 4 and 20?” is an example of a decision problem. In this case, by the way, the answer happens to be “yes”.

A decision problem is in **NP** (which stands for “Non-deterministic Polynomial” time) if a “yes” answer can be verified efficiently, i.e., in a time that grows no faster than a polynomial in the size of the problem. Hence, loosely speaking, the problems in **NP** are those such that if you *happened to guess* the solution correctly (this is the “non-deterministic” aspect) then you could *verify* the solution *efficiently* (this is the “polynomial” aspect). Hence the name “Non-deterministic Polynomial” time.

A decision problem is **NP-Complete** if it lies in the complexity class **NP** and all other problems in **NP** can be reduced to it. Thus the **NP-Complete** problems are the only ones we need to study to understand the computational resources needed to solve *all* of the problems in **NP**. Hence, the **NP-Complete** problems have a special place in complexity theory.

Notice, that a decision problem does not require that the solution on which the decision is based be exhibited, although exhibiting such a solution and then verifying it is certainly one way to arrive at the decision. For example, in the prime number example above, I could have listed out all the primes between 4 and 20, i.e., 5, 7, 11, 13, 17, 19, and then counted them to decide if there were more than five. But the distinction between deciding the answer and exhibiting the answer was

¹Source: in “Nonlinear Quantum Mechanics Implies Polynomial-Time Solution for **NP-Complete** and # **P** Problems, Phys. Rev. Lett., Volume **81** (1998) pp. 3992–3995”.

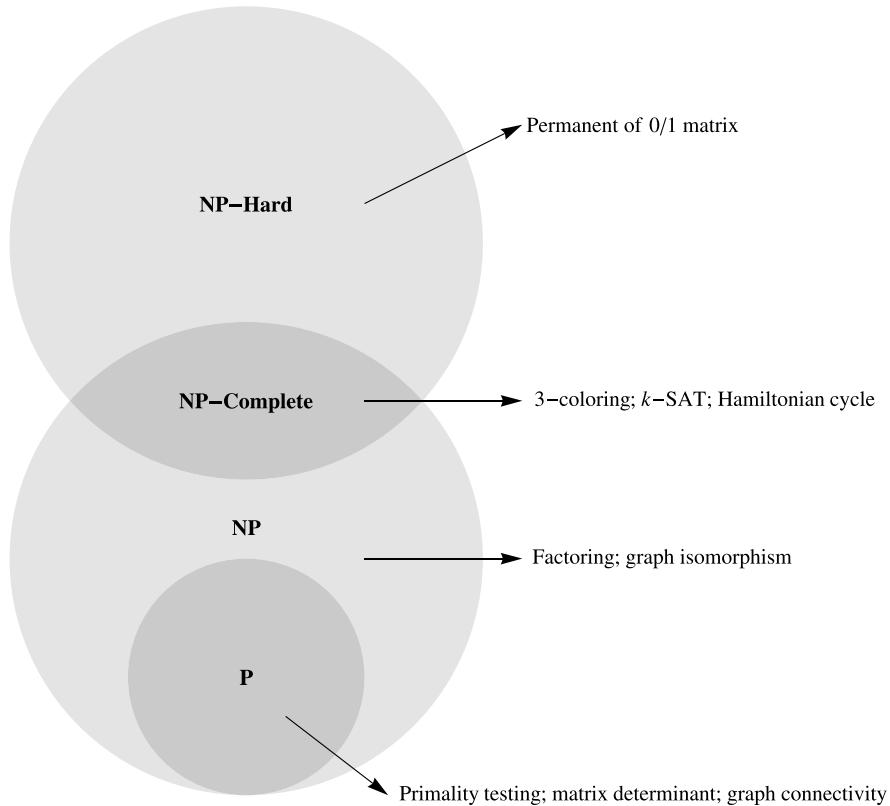


Fig. 7.1 Diagram showing the relationship between the **P**, **NP**, **NP-Complete**, and **NP-Hard** complexity classes

shown very clearly in Chap. 1 when we looked at Deutsch's algorithm for deciding if a function was constant or balanced.

Typically, if the “decision” version of some problem is **NP-Complete**, the related problem of “exhibiting a solution explicitly” is **NP-Hard**. The relationship between these complexity classes (assuming $\mathbf{P} \neq \mathbf{NP}$, which most computer scientists believe) is shown in Fig. 7.1.

The **NP-Complete** problems are amongst the most common computational problems encountered in practice [196]. Unfortunately, **NP-Complete** problems appear to be even harder than the integer factorization problem. Whereas, classically, the best known algorithm for the integer factorization scales only sub-exponentially [308, 309], the best known algorithms for solving **NP-Complete** problems all scale exponentially [196]. Thus, the discovery of Shor's quantum algorithm for factoring composite integers and computing discrete logarithms exponentially faster than is possible classically does not amount to a complete victory over the entire **NP** class. The factoring and discrete logarithm problems are in **NP**

but they are not **NP-Complete**. So Shor's algorithm tells us nothing about how efficiently we can solve **NP-Complete** problems on a quantum computer.

7.1 Importance and Ubiquity of NP-Complete Problems

Computer scientists now know of over 3000 superficially different **NP-Complete** problems. Some of the most famous examples are the following:

1. PROPOSITIONAL SATISFIABILITY (k -SAT): Find an assignment of True or False to the n variables in a Boolean formula, written in conjunctive normal form (CNF), which makes the formula True. CNF means that the formula is a conjunct (i.e., logical “AND”) of m clauses where each clause is the disjunct (i.e., logical “OR”) of k variables or negated variables. Thus a CNF formula for an instance of 3-SAT has a form such as:

$$(x_1 \vee \neg x_3 \vee x_4) \wedge \underbrace{(\neg x_2 \vee x_3 \vee x_4)}_{k \text{ variables per clause}} \wedge \cdots \wedge \underbrace{(x_1 \vee x_5 \vee x_6)}_{m \text{ clauses}}$$

Deciding if a k -SAT formula is satisfiable is **NP-Complete** for $k \geq 3$, and exhibiting a satisfying assignment is **NP-Hard**.

2. GRAPH COLORING (k -COL): Find a coloring of a graph having n nodes and m edges using k colors such that every node has a color and no two nodes that are connected to one another directly have the same color. Deciding if a coloring exists that uses no more than k colors is **NP-Complete** for $k \geq 3$, and exhibiting such a coloring is **NP-Hard**. Although GRAPH-COLORING might sound like a toy problem, it turns out to be equivalent to the SCHEDULING problem, and therefore has immense importance to industry.
3. TRAVELLING SALESMAN: Given an n -node graph with edge weights w_{ij} find a tour of the graph that visits every node once and minimizes the cumulative sum of the weights of the edges traversed. Deciding if the shortest tour has a length less than some threshold is **NP-Complete**. Exhibiting a shortest tour is **NP-Hard**.
4. SUBSET SUM: Given a set of positive and negative integers, is there a subset of those integers that sums exactly to zero? SUBSET SUM is **NP-Complete**.

Notice that there are similarities between these problems: they all involve a set of variables that may take on various allowed values (which may be discrete or continuous) and there exists a set of constraints between the variables that restricts what values the variables may assume simultaneously. Thus they are generically “constraint satisfaction problems” (if the goal is to find a solution that satisfies all constraints), “maximization” problems (if the goal is to find a solution that satisfies as many constraints as possible), or “optimization” problems (if the goal is to find the best solution according to some criterion). All these variants are typically **NP-Complete** or **NP-Hard**, depending on whether or not you simply want to merely

decide the answer or exhibit a solution, again illustrating the ubiquity and importance of these complexity classes.

In addition to their ubiquity, **NP-Complete** problems share a fortuitous kinship: any **NP-Complete** problem can be mapped into any other **NP-Complete** problem using only polynomial resources [56]. Thus, if a quantum algorithm were found that can solve one type of **NP-Complete** problem efficiently, this would immediately lead to efficient quantum algorithms for *all* **NP-Complete** problems (up to the polynomial cost of translation). And, for sure, such a discovery would mark one of the greatest advances in theoretical computer science. So, in some sense, the thousand or so different **NP-Complete** problems are really the same problem in disguise. It is therefore sufficient to focus on any single **NP-Complete** problem, for any progress made in solving that problem is likely applicable to all the other **NP-Complete** problems too, so long as you don't mind paying the polynomial cost of translation.

7.1.1 Worst Case Complexity of Solving NP-Complete Problems

Broadly speaking, algorithms for solving **NP-Complete** problems fall into two categories: “backtracking” and “heuristic”. Backtracking algorithms extend a partial solution towards a complete solution by systematically exploring the *entire* space of possible solutions, and jumping back to an earlier partial solution if a particular variable assignment can be proven to be impossible to extend into a complete solution. Heuristic algorithms make a sequence of local changes iteratively to a complete assignment of values to variables, which *tend* to increase the number of satisfied clauses. Typically the heuristic algorithms are correct (i.e., when they find a solution it is a valid solution) but incomplete (i.e., they do not check all possible value assignments to the variables before giving up, and so can miss solutions even if they do exist). However, in practice, the heuristic methods work surprisingly well often finding solutions faster than the complete search algorithms.

7.1.1.1 The Davis-Putnam-Logemann-Loveland Algorithm

The DPLL algorithm [133, 332] picks a variable in the 3-SAT formula, sets its value to `True` or `False`, simplifies the resulting formula (which now contains one fewer unassigned variable) and then checks recursively if the simplified formula is satisfiable. If the simplified formula is satisfiable, then so is the original. If not, the recursion is repeated using the other value for the truth assignment. If the problem remains unsatisfiable after all possible value assignments to the variables have been tried, the problem instance is proven unsatisfiable.

7.1.1.2 The WalkSAT Algorithm

The WalkSAT algorithm [446] picks a random truth assignment for each variable in the problem, and iteratively improves it using a combination of random and greedy changes to the Boolean values assigned to the variables. In a greedy change, the variable selected for flipping is the one that minimizes the resulting number of unsatisfied clauses. In a random change, a variable is selected at random and its bit value is flipped regardless of whether this increases or decreases the number of clauses that are unsatisfied. The choice of whether to make a random move or a greedy move is set by a user-defined probability p such that WalkSAT makes a random flip with probability p , and a greedy flip with probability $(1 - p)$. On a particular trial, WalkSAT continues until a user-defined limit of *max-flips* flips have been performed. If it has not found a solution by then, WalkSAT restarts the search from a new random assignment. The total number of re-starts allowed is set by another user-defined parameter *max-trials*. There is still some art in picking these user defined parameters to get the best overall problem solving performance.

7.1.1.3 NP-Complete Problems Are Hard in the Worst Case

A given problem instance can be easy for both types, hard for both types, or easy for one type and hard for the other type. Unfortunately, you cannot tell which is the case just by looking at the problem instance. Instead you have to attempt to solve the problem to find out whether it is an easy one or a hard one with respect to whatever algorithms you have available. As the running time needed to solve the problem exactly with certainty needs to scale, in the worst case, as an exponential function in the size of the problem **NP-Complete** problems are regarded as generically hard to solve.

Having said that, in practice it is observed that many instances of **NP-Complete** problems are much easier to solve than we might expect. This dichotomy between the “official” opinion that **NP-Complete** problems are intractable and the “practical experience” that suggests this is not always the case, led some scientists to dissect the nature of **NP-Completeness** more carefully. Rather than focussing solely on the worst case scaling of the run time needed to ensure success, they looked instead at how the difficulty of solving instances of an **NP-Complete** problem of fixed size varied as you changed the degree of constrainedness of the problem instance. This approach was motivated by analogies the scientists saw between phenomena in computational problems and phase transition phenomena in statistical physics.

7.2 Physics-Inspired View of Computational Complexity

7.2.1 Phase Transition Phenomena in Physics

Many physical systems undergo a phase transition as some property of the system is varied. For example, from ancient times it was known that when certain hot metals,

such as iron, cooled they acquired a strong and persistent magnetic field. That is they transitioned from a non-magnetic phase to a magnetic phase. When the spin of the electron was discovered it was speculated that magnetism was the collective alignment of several spins. However, it was unclear how this alignment came about. Hence, the two-dimensional Ising model was invented to simulate the behavior of simple magnets.

In the 2D Ising model a potentially magnetic material is pictured as a 2D rectangular array of quantum spins, which interact locally with their nearest neighbors. Initially the spin orientations are random. However, the nature of the spin-spin interactions between nearest neighbors is such that it is energetically more favorable for neighboring spins to be aligned (i.e., spin-up/spin-up or spin-down/spin-down) than it is for them to anti-aligned (spin-up/spin-down or vice versa). At high temperatures the thermal excitations jostle the spins around swamping their spin-spin interactions. But as the system cools the spin-spin interactions eventually dominate the thermal excitations. In this regime the system attempts to relax to its lowest energy configuration by groups of spins aligning over large spatial regions, called magnetic domains. In this regime the system attempts to relax to its lowest energy configuration by groups of spins aligning over large spatial regions, called magnetic domains. A typical distribution of magnetic domains is shown in Fig. 7.2. The relative preponderance and sizes of spin-up domains relative to spin-down domains determines the net magnetization of the material. In this case the temperature of the system controls the magnetic phase. The material will be magnetized below a critical temperature and unmagnetized above it. Statistical physics aims to predict coarse effects, such as net magnetization, or the critical temperature at which mag-

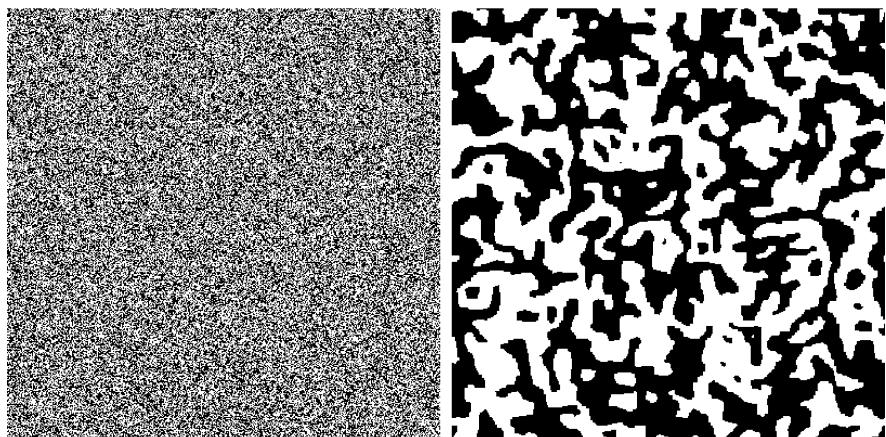


Fig. 7.2 Phase transition in magnetization based on the 2D Ising model. A magnetic material is modeled as a lattice of spins that can each be either “spin up” (white) or “spin down” black. At high temperature the spins are oriented randomly with respect to each other and the material has no net magnetization. However, as the temperature is reduced large regions of aligned spins (called magnetic domains) suddenly emerge and the material acquires a net magnetization

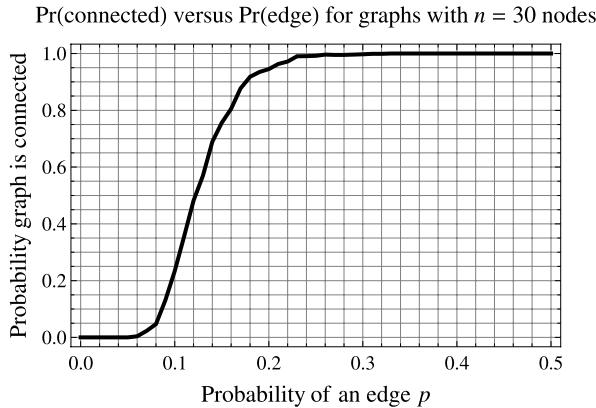


Fig. 7.3 Phase transition in connectivity of a random graph. For a graph with n nodes, when the probability of an edge $p < \frac{(1-\epsilon)\ln n}{n}$ the graph is likely to be unconnected, and when $p > \frac{(1+\epsilon)\ln n}{n}$ the graph is likely to be connected, i.e., has no isolated nodes. The data shows the probability a $n = 30$ node graph is connected as a function of the probability of an edge, p . A sharp phase transition in connectivity occurs around $p \approx \frac{\ln n}{n} = \frac{\ln 30}{30} = 0.113$. The step becomes more steep as $n \rightarrow \infty$. In the figure we computed the mean probability of being connected averaged over 1000 graphs per value of p used. Similar threshold phenomena occur in chromatic number, clique number, and the size of the giant component

netization appears, without having to know, or care about, the specific orientations of every particle.

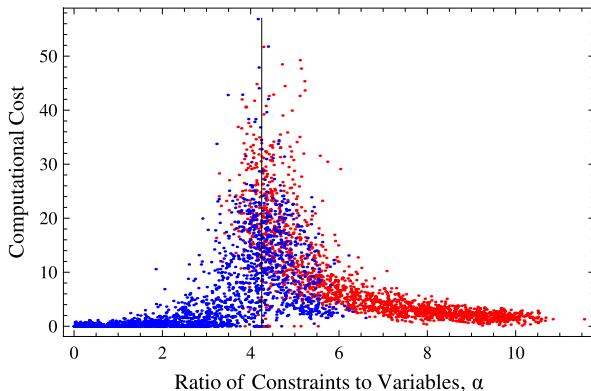
7.2.2 Phase Transition Phenomena in Mathematics

Similar phase transition phenomena occur in mathematics. For example, in the theory of random graphs there is a sharp phase transition in connectivity as the ratio of the number of edges to the number of nodes exceeds a critical value. Specifically, consider a graph having n nodes. Such a graph can have at most $\frac{1}{2}n(n - 1)$ edges. We can therefore create a random graph by fixing n and selecting each of the possible edges independently with probability p where $0 \leq p \leq 1$. You obtain graphs with different characteristics for different values of p . As shown in Fig. 7.3, when p is increased such graphs undergo a sharp phase transition from being unconnected (when $p < \frac{(1-\epsilon)\ln n}{n}$ where $0 \leq \epsilon \ll 1$) to being connected (when $p > \frac{(1+\epsilon)\ln n}{n}$ where $0 \leq \epsilon \ll 1$). Similar phase transitions occur in the chromatic number, clique number, and the size of giant component.

7.2.3 Computational Phase Transitions

Given the appearance of phase transition phenomena in Ising spin systems and in random graphs, it is not surprising that similar phase transition phenomena can arise

Fig. 7.4 Scatter plot of the cost of solving random instances of 3-SAT versus the ratio of the number of clauses to the number of variables. The hardest problem instances tend to cluster around the value $\alpha = 4.25$



in constraint networks too. Indeed there are strong similarities between Ising spin systems and constraint satisfaction problems. The spins in an Ising system, which can be “up” or “down”, play an analogous role to Boolean variables in a constraint satisfaction problem, which can be “True” or “False”. Likewise, the spin-spin interactions between a particular spin and its neighbors, play an analogous role to the constraints amongst the Boolean variables in a constraint satisfaction problem, which dictate what values they can assume.

Figure 7.4 plots the computational costs of solving random instances of 3-SAT or proving they are unsatisfiable as a function of the ratio of the number of clauses to number of variables, $\alpha = m/n$. The algorithm used is a complete SAT-solver known as the Davis-Putnam algorithm, so for every instance it is sure to determine a solution or sure to determine the instance is unsatisfiable. At a critical value of $\alpha \approx 4.25$ the problem instances suddenly become much harder to solve or much harder to prove unsatisfiable. This critical ratio also coincides with the point at which problem instances suddenly transition from being very likely to be satisfiable (for $\alpha < \alpha_{\text{crit}}$) to very likely to be unsatisfiable (for $\alpha > \alpha_{\text{crit}}$). Thus, we see a phase transition-like phenomenon going on in a computational problem.

Over recent years physicists and computer scientists have placed the analogy between Ising spin systems and constraint satisfaction problems on a firm footing by analyzing the computational phase transitions observed in constraint satisfaction problems using the tools of statistical physics [234]. This has led to insight on the internal structure of **NP-Complete** problems and how the degree of constrainedness of a problem instance is loosely correlated to the degree of difficulty in finding a solution or proving no solution is possible. Although the full mathematical methods used are complex, we can deduce the essential qualitative features of computational phase transitions using a very simple argument.

7.2.3.1 Approximate Analysis for k -SAT

In k -SAT, each clause is a disjunct of k variables (or negated variables). Such a clause can only be False when all of its k components are False. Hence, there

is only one out of the possible 2^k ways to assign Boolean values to the k components that can make all the components `False`. Hence, for a random assignment of Boolean values to the variables, each clause is `False` with probability $p = \frac{1}{2^k}$ and, therefore, `True` with probability $(1 - p) = (1 - \frac{1}{2^k})$.

Now let's assume the clauses in the CNF formula are independent of one another. This is not strictly true, of course, because for a solution to be self-consistent, the choice of Boolean assignment made to a variable in one clause is required to be the same as that made to the same variable in another clause. Nevertheless, we are only doing an approximate analysis so we will take such liberties freely. For the whole CNF formula of m clauses to be `True` we require each component of the conjunct is `True`. Hence, given a random assignment of Boolean values to variables, the probability that the whole CNF formula is `True` is (crudely) $(1 - p)^m = (1 - \frac{1}{2^k})^m$. Hence, since there are n variables in total, there are 2^n possible ways to assign Boolean values to the variables. Hence, the expected number of solutions to the CNF formula is given by $N_{\text{soln}}^{k\text{-SAT}} = 2^n(1 - \frac{1}{2^k})^m$. The parameter controlling the degree of constrainedness is the ratio of the number of clauses to the number of variables, i.e., $\alpha = \frac{m}{n}$. Hence, for a k -SAT problem having n variables, m clauses, and k variables (or negated variables) per clause, the expected number of solutions is given (roughly) by:

$$N_{\text{soln}}^{k\text{-SAT}} = 2^n \left(1 - \frac{1}{2^k}\right)^{\alpha n} \quad (7.1)$$

We can assume the hardest problem instances correspond to those cases when only one solution to the problem is viable. Hence, setting $N_{\text{soln}}^{k\text{-SAT}} = 1$ and solving for α we can estimate at what ratio of clauses to variables we expect the hardest graph coloring problems to occur. This gives us:

$$\alpha_{\text{crit}} = -\frac{\ln 2}{\ln(1 - \frac{1}{2^k})} \quad (7.2)$$

For 3-SAT this predicts the critical point at 5.2 whereas the empirically observed value is around 4.25.

7.2.3.2 Approximate Analysis for GRAPH-COLORING

In GRAPH-COLORING (k -COL), the constraints are the fact that the edges in a graph cannot be colored the same. There are k colors to choose from, so there are k^2 ways to pick a color pair for the nodes at either end of an edge. Of these k choice are forbidden because they would assign the same color to both nodes. Hence, crudely, the probability that a random coloring of a single edge is acceptable is then $p = \frac{k^2 - k}{k^2} = 1 - \frac{1}{k}$.

There are a total of m edges in the graph. Therefore, if we pretend the edges can be colored independently of one another, the probability that a random color assignment for all n nodes is acceptable is then $(1 - \frac{1}{k})^m$. So the expected number

of solutions is $k^n(1 - \frac{1}{k})^m$. This time we want to study what happens as we vary the ratio of the number of edges to the number of nodes. The average number of edges exiting a node is $\gamma = 2m/n$.

Hence, for a k -COL problem having n nodes, m edges, and k colors, the expected number of solutions is given (roughly) by:

$$N_{\text{soln}}^{k\text{-COL}} = k^n \left(1 - \frac{1}{k}\right)^{\frac{1}{2}\gamma n} \quad (7.3)$$

We can assume the hardest problem instances correspond to those cases when only one solution to the problem is viable. Hence, setting $N_{\text{soln}}^{k\text{-COL}} = 1$ and solving for γ we can estimate at what connectivity we expect the hardest graph coloring problems to occur. This gives us:

$$\gamma_{\text{crit}} = -2 \frac{\ln k}{\ln(1 - \frac{1}{k})} \quad (7.4)$$

For 3-COL this predicts the critical point at 5.4 whereas the empirically observed value is around 4.6.

7.2.4 Where Are the Really Hard Problems?

The approximate analyses neglected the correlations between the constraints. When these are taken into account they reduce the number of solutions at given values of α and γ , which pushes α_{crit} and γ_{crit} to lower values. With such corrections we can get closer to the observed phase transition points.

Experimental data on the actual computational cost encountered when solving progressively larger instances of **NP-Complete** problems, such as the data 3-SAT shown in Fig. 7.5, reveal a characteristic easy-hard-easy pattern with the cost peak coinciding with a phase transition in the solvability of the problem. Over the decade since these results first appeared more sophisticated techniques from statistical physics have been employed to bound the phase transition point more rigorously [4, 205, 206, 280, 281, 351, 361].

7.3 Quantum Algorithms for NP-Complete Problems

In the years following the publication of Shor's algorithm there was a great deal of effort put into searching for quantum algorithms that could solve **NP-Complete** problems in polynomial time. However, to date all such attempts have failed. Indeed, it has even proven to be difficult to adapt Shor's algorithm to tackle other problems, such as GRAPH-ISOMORPHISM, which are, like FACTORING and DISCRETE-LOG, in **NP** but are not **NP-Complete**. Nevertheless, these efforts did stimulate the

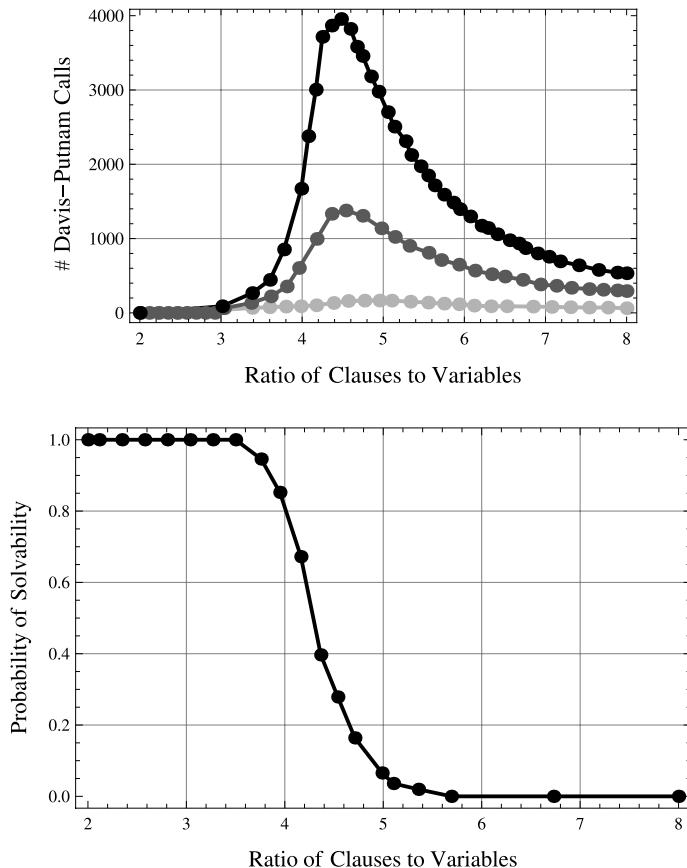


Fig. 7.5 3-SAT phase transition. The easy-hard-easy pattern is clearly visible as the ratio of the number of clauses to the number of variables increases from zero through 4.25. The hardest problem instances coincide with the region where the probability of the 3-SAT instances being soluble plummets from close to 1 to close to 0. This data is provided by Bart Selman of Cornell University

invention of new paradigms for quantum computing including, e.g., adiabatic quantum computing. Unfortunately, the early promise that adiabatic quantum computing might break the intractability of **NP-Complete** problems, in the sense of permitting such problems to be solved in guaranteed polynomial time [178], was subsequently shown to be illusory [355, 509].

7.3.1 Quantum Solution Using Grover's Algorithm

In contradistinction to Shor's quantum factoring algorithm, Grover's quantum search algorithm *can* be adapted quite readily to solve **NP-Complete** problems,

albeit again in exponential time, but with a reduced exponent compared to what is known classically. The idea is quite simple. Imagine a database containing all the possible solutions to an **NP-Complete** problem. Let's say there are N of them. Furthermore, let's imagine replacing the oracle used in Grover's algorithm with a polynomial cost "testing circuit" that can pronounce on whether or not a candidate solution is in fact a valid solution. Then one could apply the unstructured quantum search algorithm substituting the polynomial cost testing circuit in lieu of the oracle, to find a solution in square root the number of possible solutions, i.e., $O(\sqrt{N})$. For a typical **NP-Complete** problem in which one has to find an assignment of one of b values to each of μ variables, the number of candidate solutions, $N = b^\mu$, grows exponentially with μ . Hence, a classical exhaustive algorithm would therefore take a time $O(b^\mu)$ to find the solution whereas the aforementioned unstructured quantum search algorithm would take a time $O(b^{\mu/2})$. Unfortunately, although this is an impressive speedup there are already more sophisticated classical algorithms that can do better than $O(b^{\mu/2})$. Hence, a direct application of the quantum unstructured search algorithm to solving **NP-Complete** problems is not worthwhile.

7.3.2 Structured Search Spaces: Trees and Lattices

Fortunately, there is a way to improve upon a naïve use of quantum search in solving **NP-Complete** problems. This is possible because the search spaces of **NP-Complete** problems typically have structure in the sense that one can build up complete solutions (i.e., value assignments for all the variables) by extending *partial* solutions (i.e., value assignments for a subset of the variables). Thus, rather than performing an unstructured quantum search amongst *all* the candidate solutions (treating them as undifferentiated entities), in an **NP-Complete** problem, we can perform a quantum search amongst the *partial* solutions in order to narrow the subsequent quantum search amongst their extensions. This is reminiscent of a tree-search amongst the partial solutions, in which a parent node represents a partial solution and its children all the logical extensions of that partial solution. Such a tree of partial solutions is illustrated in Fig. 7.6.

Such a tree-structured quantum search allows us to find a solution to an **NP-Complete** problem in a time that grows, on average, as $O(b^{\alpha\mu/2})$ for the hardest problems (as localized by the phase transition results above), where $\alpha < 1$ is a constant depending on the problem instance considered.

To explain this approach more quantitatively we need to analyze the structure of these search trees. Our goal is to find the probability that node in a typical search tree is "good", i.e., is consistent with respect to all of the constraints against which it can be tested. These are the subset of the problem constraints that only involve those variables that have been assigned values so far in the search tree. Let us call this probability $p(i)$, the probability that a node at level i in the search tree is "good".

The determination of $p(i)$ is complicated by the fact that the *order* in which variables are assigned values can influence, greatly, the computational cost of solving

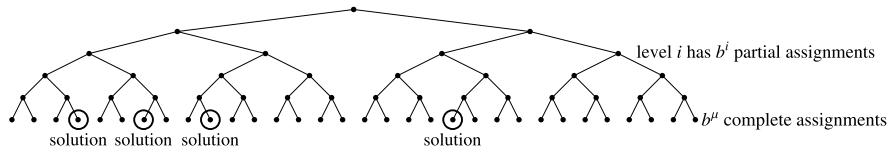


Fig. 7.6 A tree of partial solutions. Each node represents a distinct assignment of values to a particular subset of the variables in the problem, extending the assignment by one extra variable at each level. Thus, all the nodes at a given level correspond to the same subset of variables assigned values in all possible ways. If there are a total of μ variables each of which can take on one of b possible values, the tree will have a depth of μ , a branching ratio of b , and there will be at most b^μ possible nodes at level i . Moreover, there will be b^μ complete assignments (i.e., potential solutions) amongst the leaf nodes at the bottom level of the tree of which only a subset will correspond to complete solutions. If a node corresponding to a partial solution is found to be inconsistent, then descendants of that node need not be considered, and paths extending from these nodes can be omitted from the diagram

the problem. For example, suppose that due to the constraints a particular variable is forced to take only one value. If that variable is examined early in the search process, i.e., high up in the search tree, then the fact that it must take a certain value allows us to prune all partial solutions in which it took some other value. As this pruning occurs high in the search tree, an enormous part of the potential search space can be eliminated. Conversely, if this variable is examined late in the search process, much of the tree might already have been explored, resulting in relatively little gain. So we need a trick for computing the probability that a node is good, averaged over all variable orderings.

The simplest way to do this is to consider a *lattice* of partial solutions rather than a *tree* of partial solutions as shown in Fig. 7.7.

Each node in a lattice corresponds to a particular assignment of values to a particular subset of variables. The i -th level of such a lattice contains all possible subsets of variables of a certain size, assigned values in all possible ways. Thus a *lattice* of partial solutions effectively encodes *all possible variable orderings* simultaneously. Thus, a lattice over-represents a search space. But its advantage is that it makes it much easier to assess what is likely to be encountered in an average tree search, where we have factored out bias due to reliance on a particular variable ordering. However, the size of such lattices grows rapidly with increasing values of μ .

7.3.2.1 Computing the Lattice Parameters for a Constraint Satisfaction Problem

Quantitatively, the i th level of a lattice of partial solutions represents all possible subsets of i variables out of μ variables, assigned values in all possible combinations. Thus, in a lattice there are $\binom{\mu}{i} b^i$ nodes at level i rather than the b^i nodes in a tree. So each level of the lattice encodes the information contained in $\binom{\mu}{i}$ different trees.

As each constraint involves exactly k variables, and each variable can be assigned any one of its b allowed values, there are exactly b^k “ground instances” of each

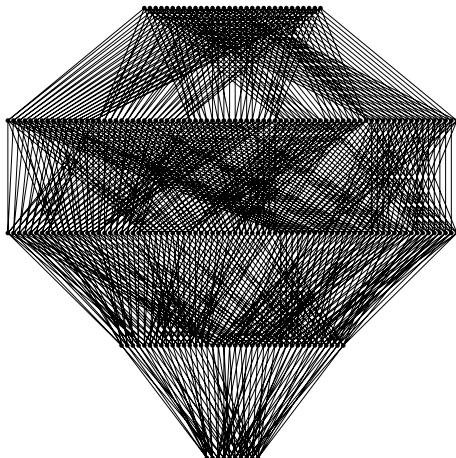


Fig. 7.7 A lattice of partial solutions. Here each row of the lattice represents all subsets of i out of μ variables, each assigned any of b values in all possible ways. Hence, there are $\binom{\mu}{i} b^i$ nodes at level i of the lattice. The figure corresponds to a lattice having $\mu = 5$ variables, each of which can be assigned any one of $b = 2$ values. The solutions (complete consistent assignments) reside at the top of the lattice. Lower down, the nodes correspond to partial assignments, i.e., tuples of variable/value pairs for a subset of i out of μ variables assigned values in all possible ways. Those partial assignments that satisfy all the problem constraints against which they may be tested are deemed “good”, and those that do not are deemed “nogood”. In a graph coloring problem, e.g., the pairwise assignments at level $i = 2$ would correspond to color (value) choices for any pair of vertices (variables). Each such choice is either “good” (meaning the two vertices are allowed to be colored in the way chosen) or “nogood” (meaning that the two vertices cannot be colored the same as they are directly connected by an edge). Complete solutions can only be extensions of good nodes. As soon as a node in the lattice is found to be nogood, no further nodes above it need be considered

constraint. Moreover, as each constraint involves a different combination of k out of a possible μ variables, there can be at most $\binom{\mu}{k}$ constraints. Each ground instance of a constraint may be “good” or “nogood”, so the number of ground instances that are “nogood”, x_i , must be such that $0 \leq x_i \leq b^k \binom{\mu}{k}$. If x_i is small the problem typically has many solutions. If x_i is large the problem typically has few, or perhaps no, solutions. The exact placement of the x_i nogoods is, of course, important in determining their ultimate pruning power.

7.3.2.2 Calculation of $p(i)$

To estimate $p(i)$ in an *average* tree, we calculate the corresponding probability that a node in the lattice (which implicitly incorporates *all* trees) is “nogood”, conditional on there being ξ “nogoods” at level k . For a node at level i of the lattice to be “good” it must not sit above any of the ξ “nogoods” at level k . A node at level i of the lattice sits above $\binom{i}{k}$ nodes at level k . Thus, out of a total possible pool of $b^k \binom{\mu}{k}$ nodes at level k , we must exclude $\binom{i}{k}$ of them. However, we can pick the ξ nogoods from

amongst the remaining nodes in any way whatsoever. Hence the probability that a node is “good” at level i , given that there are ξ “nogoods” at level k , is given by the ratio of the number of ways to pick the “nogoods” such that a particular node at level i is “good”, to the total number of ways of picking the ξ “nogoods”. As a consequence, the probability for a partial solution to be good at level i in a tree of height μ and branching ratio b can be approximated as [537–539]

$$p(i) = \frac{\binom{b^k(\mu)}{k} - \binom{i}{k}}{\binom{b^k(\mu)}{\xi}} \quad (7.5)$$

where k is the size of the constraint (i.e., number of variables involved in a constraint) and ξ is the number of “nogood” ground instances (or number of constraints). This approximation essentially relies on the assumption that the partial solutions at a given level are uncorrelated. Strictly speaking this is not exactly the case in real problems, but the approximation is good enough to predict the correct qualitative features of real search spaces.

Now, we are interested in obtaining an asymptotic expression for $p(i)$ for large problems, i.e., when the number of variables $\mu \rightarrow \infty$. Recall that to scale a constraint satisfaction problem up, however, it is not sufficient to increase only μ . In addition, we ought also to increase the number of constraints so as to preserve the “constrainedness-per-variable”, $\beta = \xi/\mu$. Thus, when we consider scaling our problems up, as we must do to assess the asymptotic behavior of the classical and quantum structured search algorithms, we have $\mu \rightarrow \infty$ and scale $\xi = \beta\mu$, keeping β , b and k constant.² We now make the assumption that $\xi \ll b^k \binom{\mu}{k}$ and $\xi \ll b^k \binom{\mu}{k} - \binom{i}{k}$, which is justified in the asymptotic regime. Using Stirling formula, we have

$$\frac{\binom{M}{K}}{\binom{N}{K}} \simeq \frac{(M-K)^K}{(N-K)^K} \simeq \left(\frac{M}{N}\right)^K \quad (7.6)$$

for large M and N , provided that $K \ll M, N$. This allows us to reexpress (7.5) as

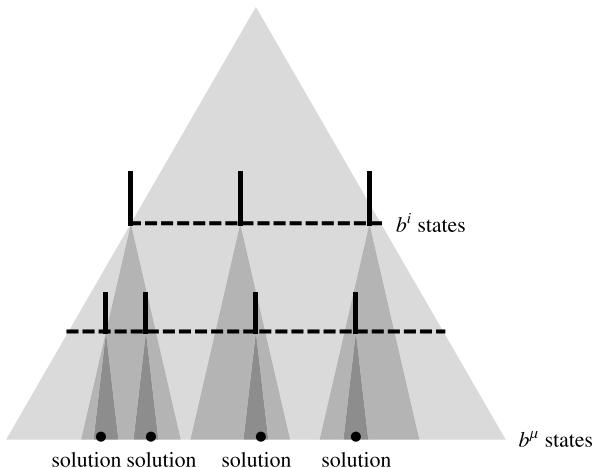
$$p(i) = \left(1 - b^{-k} \frac{\binom{i}{k}}{\binom{\mu}{k}}\right)^\xi \quad (7.7)$$

Now, assuming that $k \ll i$ and $k \ll \mu$, and reusing (7.6), we have

$$p(i) = \left(1 - b^{-k} \left(\frac{i}{\mu}\right)^k\right)^\xi \quad (7.8)$$

²For graph coloring, this scaling assumption corresponds to adding more edges to the graph as we allow the number of nodes to go to infinity, while simultaneously keeping the average connectivity (number of edges per node) and the number of colors fixed.

Fig. 7.8 Schematic representation of stages (1) and (2) of the quantum structured search algorithm. These operations partially amplify the solution states, and can be nested into a standard quantum search algorithm (3) in order to speedup the amplification of the solutions



for large i and μ . Finally, assuming for simplicity that $b^k \gg 1$ and $(i/\mu)^k \ll 1$, we obtain

$$p(i) = b^{-\mu(\frac{\beta}{\beta_c})(\frac{i}{\mu})^k} \quad (7.9)$$

where $\beta = \xi/\mu$ measures the difficulty of the problem and $\beta_c = b^k \log(b)$ is the critical value around which the problem is the most difficult.

7.4 Quantum Solution Using Nested Grover's Algorithm

Our improved quantum search algorithm works by *nesting* one quantum search within another, as illustrated in Fig. 7.8. Specifically, by performing a quantum search at a carefully selected level in the tree of partial solutions, we can narrow the effective quantum search amongst the candidate solutions so that the net computational cost is minimized. The resulting algorithm is the quantum counterpart of a *classical* nested search algorithm which scales as $O(b^{\alpha\mu})$, giving a square root speedup overall. The nested search procedure mentioned here corresponds to a *single* level of (classical or quantum) nesting, but it can be extended easily to several nesting levels.

The expected time to find a solution grows as $O(b^{\alpha\mu/2})$, that is, as the square root of the classical time for problem instances in the hard region. The constant α , depending on the problem considered, is shown to decrease with an increasing nesting depth (i.e., an increasing number of nesting levels).

7.4.1 The Core Quantum Algorithm

Assume that the Hilbert space of our search problem is the tensor product of two Hilbert spaces \mathcal{H}_A and \mathcal{H}_B . As before, A denotes the set of primary variables, that

is, the variables to which we assign a value in the first stage. The partial solutions correspond to definite values for these variables. Thus, \mathcal{H}_A represents the search space for partial solutions (of dimension d_A). The set of secondary variables, characterizing the extensions of partial solutions, is denoted by B , and the corresponding Hilbert space \mathcal{H}_B is of dimension d_B . The quantum algorithm with a single nesting level works as follows:

Quantum Structured Search Algorithm

1. Construct a superposition (with equal amplitudes) of all the could-be solutions at level i by use of the standard unstructured search algorithm based on H .
2. Perform a subsequent quantum search in the subspace of the descendants of *all* the could-be partial solutions, simultaneously. This second stage is achieved by using the standard quantum search algorithm with, as an input, the *superposition* of could-be solutions resulting from the first stage. The overall yield of stages (1) and (2) is a superposition of all states where the solutions have been partially amplified with respect to non-solutions.
3. Nest stages (1) and (2)—using them as a search operator U —inside a higher-level quantum search algorithm until the solutions get maximally amplified, at which point a measurement is performed. This is summarized in Fig. 7.8.

7.4.2 Analysis of Quantum Structured Search

Next we look at the steps in the quantum structured search algorithm in more detail and estimate the number of iterations required to ensure success. To start we assume there is a single “cut” level in the search tree. We can then think of performing a quantum search at some intermediate “cut” level i and using the superposition so created as the starting state for a quantum search in the leaves of the search tree. Ultimately, we’re going to nest these two operations together rather than doing them sequentially.

The starting state of the search is denoted as $|s, s'\rangle$, where $|s\rangle$ is state that lies in Hilbert space \mathcal{H}_A (i.e., the top of the tree up to the “cut” level) and $|s'\rangle$ in a state that lies in Hilbert \mathcal{H}_B (i.e., bottom of the tree from cut level to the leaves). The number of qubits in the combined register needs to be enough to hold the b^μ leaf nodes of the search tree at level μ .

Register A stores the starting state at an intermediate level i in the tree, while register B stores the continuation of that state at level μ . In other words, A holds partial solutions and B their elaboration in the leaves of the tree.

Step 1. Standard Quantum Search at Intermediate Level i

The first stage of the algorithm consists of a standard quantum search for *could-be* partial solutions $|c\rangle$ at level i , that is, states in subspace \mathcal{H}_A that do not violate any (testable) constraint.

We start from state $|s\rangle$ in subspace \mathcal{H}_A , and apply a quantum search based on the Walsh-Hadamard transformation H since we do not have *a priori* knowledge about the location of could-be solutions. The use of H is the least biased assumption since all states are *a priori* assumed to be equally likely to be solutions.

Using

$$\langle c|H|s\rangle = \pm 1/\sqrt{d_A} \quad (7.10)$$

we can perform an amplification of the components $|c\rangle$ based on $Q = -H\mathbb{1}_s H\mathbb{1}_c$ where

$$\mathbb{1}_s = \exp(i\pi|s\rangle\langle s|) \quad (7.11)$$

$$\mathbb{1}_c = \exp\left(i\pi \sum_{c \in C} |c\rangle\langle c|\right) \quad (7.12)$$

The states $|c\rangle$ correspond to the could-be partial solutions in \mathcal{H}_A (assignment of the primary variables that could lead to a solution), and belong to the subset $C = \{c_1, \dots, c_{n_A}\}$.

We assume that there are n_A could-be partial solutions, with $1 \ll n_A \ll d_A$. The quadratic amplification of these could-be solutions, starting from $|s\rangle$, is reflected by

$$\langle c|Q^n H|s\rangle \simeq n \langle c|H|s\rangle \simeq n/\sqrt{d_A} \quad (7.13)$$

for small rotation angle. Thus, applying Q sequentially, we can construct a superposition of all the could-be solutions $|c\rangle$, each with an amplitude of order $\sim 1/\sqrt{n_A}$. The required number of iterations of Q scales as

$$n \simeq \sqrt{d_A/n_A} \quad (7.14)$$

This amplitude amplification process can equivalently be described in the joint Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$, starting from the product state $|s, s'\rangle$, where $|s'\rangle$ denotes an arbitrary starting state in \mathcal{H}_B , and applying $(Q \otimes \mathbb{1})$ sequentially:

$$\langle c, s'|(Q \otimes \mathbb{1})^n(H \otimes \mathbb{1})|s, s'\rangle = \langle c|Q^n H|s\rangle \sim n/\sqrt{d_A} \quad (7.15)$$

Here and below, we use the convention that the left (right) term in a tensor product refers to subspace A (B).

Step 2. Standard Quantum Search Amongst Descendants

The second stage of the algorithm is a standard quantum search for the secondary variables B in the subspace of the “descendants” of the could-be solutions that have been singled out in stage (1).

As before, we can use the search operator H that connects extended could-be solutions $|c, s'\rangle$ to the actual solutions or target states $|t, t'\rangle$ in the joint Hilbert space:

$$\langle t, t'|(\mathbb{1} \otimes H)|c, s'\rangle = \langle t|c\rangle \langle t'|H|s'\rangle = \pm \delta_{c,t}/\sqrt{d_B} \quad (7.16)$$

Note that, this matrix element is non-vanishing only for could-be states $|c\rangle$ that lead to an actual solution. Define the operator $R = -(\mathbb{1} \otimes H\mathbb{1}_{s'}H)\mathbb{1}_t$, with

$$\mathbb{1}_{s'} = \exp(i\pi|s'\rangle\langle s'|) \quad (7.17)$$

$$\mathbb{1}_t = \exp\left(i\pi \sum_{(t,t') \in T} |t, t'\rangle\langle t, t'|\right) \quad (7.18)$$

where T is the set of solutions $|t, t'\rangle$ at the bottom of the tree, and $\#(T) = n_{AB}$, i.e., the problem admits n_{AB} solutions. We can apply the operator R sequentially in order to amplify a target state $|t, t'\rangle$, namely

$$\langle t, t' | R^m (\mathbb{1} \otimes H) | c, s' \rangle \simeq \begin{cases} m \langle t, t' | (\mathbb{1} \otimes H) | c, s' \rangle & \text{if } c = t \\ \langle t, t' | (\mathbb{1} \otimes H) | c, s' \rangle & \text{if } c \neq t \end{cases} \quad (7.19)$$

for small rotation angle. Note that, for a could-be state $|c\rangle$ that does not lead to a solution ($c \neq t$), we have $\mathbb{1}_t |c, x\rangle = |c, x\rangle$ for all x , so that $R^m (\mathbb{1} \otimes H) |c, s' \rangle = (-\mathbb{1} \otimes H\mathbb{1}_{s'}H)^m (\mathbb{1} \otimes H) |c, s' \rangle = (\mathbb{1} \otimes H) |c, s' \rangle$, and the matrix element is not amplified by m compared to the case $c = t$. In other words, no amplification occurs in the space of descendants of could-be partial solutions that do not lead to an actual solution. Thus, (7.19) results in

$$\langle t, t' | R^m (\mathbb{1} \otimes H) | c, s' \rangle \simeq \frac{m}{\sqrt{d_B}} \delta_{c,t} \quad (7.20)$$

Assuming that, among the descendants of each could-be solution $|c, s'\rangle$, there is either zero or one solution, we need to iterate R of the order of

$$m \simeq \sqrt{d_B} \quad (7.21)$$

times in order to maximally amplify each solution. We then obtain a superposition of the solution states $|t, t'\rangle$, each with an amplitude $\sim 1/\sqrt{n_A}$. This can also be seen by combining (7.15) and (7.20), and using the resolution of identity $\mathbb{1} = \sum_{x,y} |x, y\rangle\langle x, y|$:

$$\begin{aligned} & \langle t, t' | \underbrace{R^m (\mathbb{1} \otimes H) (Q \otimes \mathbb{1})^n (H \otimes \mathbb{1})}_{U} | s, s' \rangle \\ &= \sum_{x,y} \langle t, t' | R^m (\mathbb{1} \otimes H) | x, y \rangle \langle x, y | (Q \otimes \mathbb{1})^n (H \otimes \mathbb{1}) | s, s' \rangle \\ &= \langle t, t' | R^m (\mathbb{1} \otimes H) | t, s' \rangle \langle t, s' | (Q \otimes \mathbb{1})^n (H \otimes \mathbb{1}) | s, s' \rangle \\ &\simeq (m/\sqrt{d_B})(n/\sqrt{d_A}) \simeq 1/\sqrt{n_A} \end{aligned} \quad (7.22)$$

Thus, applying the operator Q^n followed by the operator R^m connects the starting state $|s, s'\rangle$ to each of the solutions $|t, t'\rangle$ of the problem with a matrix element of order $\sim 1/\sqrt{n_A}$.

Step 3. Nest the Two Previous Searches

The third stage consists in using the operator $U \equiv R^m(\mathbb{1} \otimes H)(Q \otimes \mathbb{1})^n(H \otimes \mathbb{1})$ resulting from steps (i) and (ii) as a search operator for a higher-level quantum search algorithm, in order to further amplify the superposition of n_{AB} target (or solution) states $|t, t'\rangle$. The goal is thus to construct such a superposition where each solution has an amplitude of order $\sim 1/\sqrt{n_{AB}}$. As before, we can make use of the operator $S = -U(\mathbb{1}_s \otimes \mathbb{1}_{s'})U^\dagger\mathbb{1}_t$ where $\mathbb{1}_s$, $\mathbb{1}_{s'}$, and $\mathbb{1}_t$ are defined in (7.11), (7.17), and (7.18), in order to perform amplification according to the relation

$$\langle t, t'|S^r U|s, s'\rangle \simeq r \langle t, t'|U|s, s'\rangle \simeq r/\sqrt{n_A} \quad (7.23)$$

for small rotation angle. The number of iterations of S required to maximally amplify the solutions is thus of the order of

$$r \simeq \sqrt{\frac{n_A}{n_{AB}}} \quad (7.24)$$

This completes the algorithm. At this point, it is sufficient to perform a measurement of the amplified superposition of solutions. This yields one solution $|t, t'\rangle$ with a probability of order 1.

7.4.3 Quantum Circuit for Quantum Structured Search

The quantum network that implements this nested quantum search algorithm is illustrated in Fig. 7.9. Clearly, a sequence of two quantum search circuits (a search in the A space followed by a search in the B space) is *nested* into a global search circuit in the whole Hilbert space \mathcal{H}_{AB} . This can be interpreted as a “dynamical” choice of the search operator U that is used in the global quantum search. This quantum nesting is distinct from a procedure where one would try to choose an optimum U before running the quantum search by making use of the structure *classically* (making several classical queries to the oracle) in order to speedup the resulting quantum search. Here, no measurement is involved and structure is used at the quantum level.

7.4.4 Quantum Average-Case Complexity

Let us estimate the total number of iterations, or more precisely the number of times that a controlled-phase operator ($\mathbb{1}_t$, which flips the phase of a solution, or $\mathbb{1}_c$, which flips the phase of a could-be partial solution) is used. Since we need to repeat r times the operation S , which itself requires applying n times Q and m times R , we obtain for the quantum computation time

$$T_q \simeq r(n + m) \simeq \frac{\sqrt{d_A} + \sqrt{n_A d_B}}{\sqrt{n_{AB}}} \quad (7.25)$$

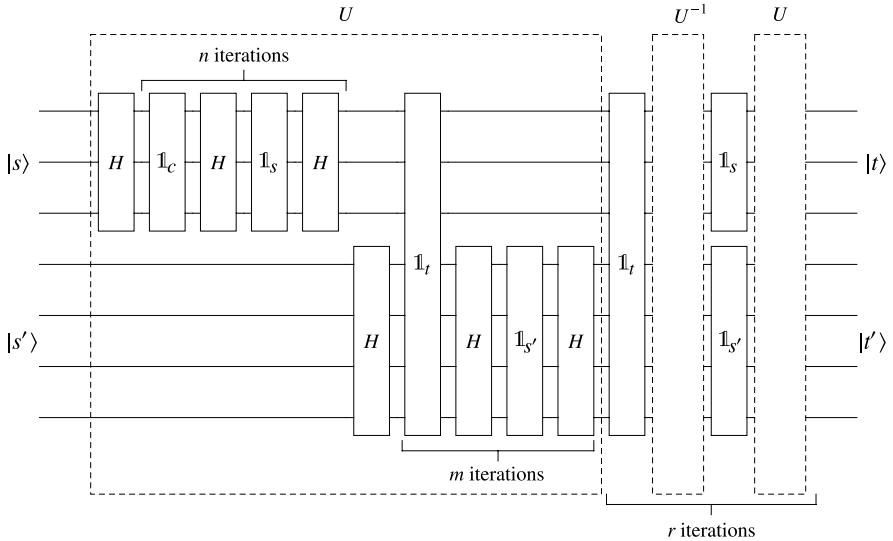


Fig. 7.9 Circuit implementing the nested quantum search algorithm (with a single level of nesting). The upper set of quantum variables, initially in state $|s\rangle$, corresponds to the primary variables A . The lower set of quantum variables, initially in states $|s'\rangle$, is associated with the secondary variables B . The quantum circuit makes use of controlled-phase gates $\mathbb{1}_s = \exp(i\pi|s\rangle\langle s|)$, $\mathbb{1}_{s'} = \exp(i\pi|s'\rangle\langle s'|)$, $\mathbb{1}_c = \exp(i\pi \sum_{c \in C} |c\rangle\langle c|)$, and $\mathbb{1}_t = \exp(i\pi \sum_{(t,t') \in T} |t,t'\rangle\langle t,t'|)$, and Walsh-Hadamard gates H . The entire operation of U (exhibited inside the dashed box) is repeated r times. Note that $U^{-1} = U^\dagger$ corresponds to same the circuit as U but read from right to left

This expression has the following interpretation. The first term in the numerator corresponds to a quantum search for the could-be partial solutions in space of size d_A . The second term is associated with a quantum search of actual solutions in the space of all the descendants of the n_A could-be solutions (each of them has a subspace of descendants of size d_B). The denominator accounts for the fact that the total number of iterations decreases with the square root of the number of solutions of the problem n_{AB} , as in the standard quantum search algorithm.

Let us now estimate the scaling of the computation time required by this quantum nested algorithm for a large search space ($\mu \rightarrow \infty$). Remember that μ is the number of variables (number of nodes for the graph coloring problem) and b is the number of values (colors) per variable. As before, if we “cut” the tree at level i (i.e., assigning a value to i variables out of μ defines a partial solution), we have $d_A = b^i$ and $d_B = b^{\mu-i}$. Also, we have $n_A = p(i)b^i$, and $n_{AB} = p(\mu)b^\mu$, where $p(i)$ is the probability of having a partial solution at level i that is “good” in a tree of height μ . (The quantity $p(\mu)$ is thus the probability of having a solution in the total search space.) We can reexpress the computation time as a function of i ,

$$T_q(i) = \frac{\sqrt{b^i} + \sqrt{p(i)b^\mu}}{\sqrt{p(\mu)b^\mu}} \quad (7.26)$$

In order to determine the scaling of T_q , we use the asymptotic estimate of $p(i)$, which is derived in [100], namely

$$p(i) = b^{-\mu(\frac{\beta}{\beta_c})(\frac{i}{\mu})^k} \quad (7.27)$$

Equation (7.27) is a good approximation of $p(i)$ in the asymptotic regime, i.e., when the dimension of the problem μ (or the number of variables) tends to infinity. Remember that, in order to keep the difficulty constant when increasing the size of the problem, we need to choose the number of constraints $\xi = \beta\mu$ when $\mu \rightarrow \infty$.³ The constant β corresponds to the average number of constraints *per variable*, and is a measure of the difficulty of the problem. The difficulty is maximum when β is close to a *critical* value $\beta_c = b^k \log(b)$, where k is the size of the constraint (i.e., number of variables involved in a constraint). Note that $p(\mu) = b^{-\mu(\beta/\beta_c)}$, implying that the number of solutions at the bottom of the tree is $n(\mu) = b^{\mu(1-\beta/\beta_c)}$. Thus, if $\beta \simeq \beta_c$, we have $p(\mu) \simeq b^{-\mu}$, so that the problem admits of the order of $n(\mu) \simeq 1$ solutions. This corresponds indeed to the hardest case, where one is searching for a single solution in the entire search space. When $\beta < \beta_c$, however, there are less constraints and the problem admits more than one solution, on average. If $\beta > \beta_c$, the problem is overconstrained, and it typically becomes easier to check the nonexistence of a solution.

Now, plugging (7.27) into (7.26), we obtain for the quantum computation time

$$T_q(i) \simeq \frac{\sqrt{b^i} + \sqrt{b^{\mu-\mu(\beta/\beta_c)(i/\mu)^k}}}{\sqrt{b^{\mu-\mu(\beta/\beta_c)}}} \quad (7.28)$$

Defining the *reduced* level on the tree as $x = i/\mu$, i.e., the fraction of the height of the tree at which we exploit the structure of the problem, we have

$$T_q(x) = \frac{a^x + a^{1-(\beta/\beta_c)x^k}}{a^{1-\beta/\beta_c}} \quad (7.29)$$

where $a \equiv \sqrt{b^\mu}$. Now, we want to find the value of x that minimizes the computation time $T_q(x)$, so we have to solve

$$(\beta/\beta_c) k x^{k-1} = a^{(\beta/\beta_c)x^k + x - 1} \quad (7.30)$$

For large μ (or large a), this equation asymptotically reduces to

$$(\beta/\beta_c) x^k + x - 1 = 0 \quad (7.31)$$

The solution x (with $0 \leq x \leq 1$) corresponds therefore to the reduced level for which $T_q(x)$ grows asymptotically ($\mu \rightarrow \infty$) with the smallest power in b . Note that this

³For the graph coloring problem, since $\xi = eb$ (where e being the number of edges and b the number of colors), it implies that the number of edges must grow linearly with the number of nodes for a fixed number of colors in order to preserve the difficulty. In other words, the average connectivity must remain constant.

optimum x is such that both terms in the numerator of (7.28) grow with the same power in b (for large μ). This reflects that there is a particular fraction x of the height of the tree where it is optimal to “cut”, i.e., to look at partial solutions. The optimum computation time can then be written as

$$T_q \simeq \frac{2a^\alpha}{a^{1-\beta/\beta_c}} \simeq \frac{\sqrt{b^{\alpha\mu}}}{\sqrt{b^{\mu(1-\beta/\beta_c)}}} \quad (7.32)$$

where the constant $\alpha < 1$ is defined as the solution x of (7.31).⁴ Note that, for a search with several levels of nesting, the constant $\alpha < x$ (see [100] for details).

Equation (7.32) implies that the scaling of the quantum search in a space of dimension $d = b^\mu$ is essentially $O(d^{\alpha/2})$ modulo the denominator (which simply accounts for the number of solutions). In contrast, the standard *unstructured* quantum search algorithm applied to this problem corresponds to $\alpha = x = 1$, with a computation time scaling as $T_q(\alpha = 1) = O(d^{1/2})$. This means that exploiting the structure in the quantum algorithm results in a decrease of the power in b by a coefficient α : the power $1/2$ of the standard quantum search is reduced to $\alpha/2$ for this nested quantum search algorithm. Consider this result at $\beta = \beta_c$, i.e., when the difficulty of the problem is maximum for a given size μ . This is the most interesting case since when $\beta < \beta_c$, the problem becomes easier to solve classically. For $\beta = \beta_c$, the nested algorithm essentially scales as

$$T_q \simeq d^{\alpha/2} = \sqrt{b^{\alpha\mu}} \quad (7.33)$$

where $\alpha = x < 1$ with x being the solution of $x^k + x - 1 = 0$, and $d = b^\mu$ is the dimension of the search space. This represents a significant improvement over the scaling of the unstructured quantum search algorithm, $O(d^{1/2})$. Nevertheless, it must be emphasized that the speedup with respect to the computation time $O(d^\alpha)$ of the classical nested algorithm is exactly a square root. This implies that this nested quantum search algorithm is the *optimum* quantum version of this particular classical non-deterministic algorithm.

For the graph coloring problem ($k = 2$), we must solve the linear equation of second order $x^2 + x - 1 = 0$, whose solution is simply $x = (-1 + \sqrt{5})/2 = 0.6180$. (When $k > 2$, the solution for x increases, and tends to 1 for large k .) This means that the level on the tree where it is optimal to use the structure is at about 62% of the total height of the tree, i.e., when assigning values to about 62% of the μ variables. In this case, the computation time of the nested algorithm scales as $O(d^{0.31})$, which is clearly an important computational gain compared to $O(d^{0.5})$.

Consider the regime where $\beta < \beta_c$, i.e., there are fewer constraints and therefore more than one solution on average, so that the problem becomes easier to solve. For a given k , the solution x of (7.31) increases when β decreases, and tends asymptotically to 1 for $\beta \rightarrow 0$. This means that we recover the *unstructured* quantum search

⁴We may ignore the prefactor 2 as it only yields an additive constant in the logarithm of the computation time.

algorithm in the limit where $\beta \rightarrow 0$. The denominator in (7.32) increases, and it is easy to check that the computation time

$$T_q \simeq \sqrt{b^{\mu(\alpha-1+\beta/\beta_c)}} \quad (7.34)$$

decreases when β decreases. As expected, the computation time of the nested algorithm approaches $O(\sqrt{d^{\beta/\beta_c}})$ as β tends to 0 (or $x \rightarrow 1$), that is, it reduces to the time of the standard unstructured quantum search algorithm at the limit $\beta \rightarrow 0$.

7.5 Summary

In the 1990's a handful of computer scientists with a background in physics began looking at computational complexity from a fresh perspective [101, 234, 280, 537–540]. They wanted to know how the computational cost to solve an **NP-Complete** problem varied with the degree of constrainedness of the problem instances. They found that there is a critical value in constrainedness at which the difficulty of finding a solution rises steeply. Moreover, empirically, this region also coincides with an abrupt collapse in the probability of there being a valid solution. This has led to more physics-insight into analogies between the structure of **NP-Complete** problems and physical phase transitions [4, 205, 206, 281, 351, 361].

The problems of factoring composite integers and computing discrete logarithms, which are addressed by Shor's algorithm, both fall within the **NP** complexity class, but neither is **NP-Complete**. Thus although Shor's algorithm is remarkable, it only achieves an exponential speedup on two of the easier problems within **NP**. This led many computer scientists to question whether similar quantum algorithms could be found that solve **NP-Complete** problems in polynomial time. At this point, no such quantum algorithms have been found. It has even proven difficult to extend Shor's algorithm to solve other problems, such as GRAPH-ISOMORPHISM, which are in **NP** that are not **NP-Complete**.

However, it is possible to apply Grover's algorithm to solve **NP-Complete** problems. Unfortunately, if this is done naively, by merely amplitude amplifying in the leaves of the search tree without exploiting its structure, the resulting speedup is insufficient to beat the best known classical algorithms for **NP-Complete** problems [379]. However, a variant of Grover's algorithm, which nests one Grover search within another (and another . . . and another . . . etc.), does make use of the implicit structure inherent in the **NP-Complete** problem, and thereby yields an average case complexity that beat the best known classical algorithms for solving that kind of **NP-Complete** problem. The average case complexity analysis of the expected running time of this algorithm makes use of the phase transition results mentioned earlier.

7.6 Exercises

7.1 Phase transition phenomena are very common, and even arise in graph theory.

- (a) Write a computer program that generates a random graph having n nodes and m edges
- (b) Write a computer program to test whether or not an n node, m edge, graph is connected
- (c) Choose ten values of m that span across the range from $0 < m < \frac{1}{2}n(n - 1)$. For each value of m generate 100 random graphs having $n = 50$ nodes and m edges and record the fraction of those graphs that are connected
- (d) Plot your data to visualize how the fraction of graphs that are connected varies as you increase the number of edges. Does the fraction change linearly from $m = 0$ to $m = n$ or nonlinearly?
- (e) What happens if you use larger graphs and much larger sample sets?

7.2 Suppose you are asked to find all ways to color a graph containing n nodes and m edges with k colors.

- (a) Roughly, how many k -colorings would you expect there to be?
- (b) Plot a graph illustrating how the number of colorings is expected to vary as you vary the ratio of $2m/n$ (for fixed n)
- (c) At what value of the ratio $2m/n$ would you expect the hardest graph coloring problems to be most often encountered?

7.3 Suppose you are asked to solve $k = \text{SAT}$ problems involving n variables and m clauses.

- (a) Roughly, how many solutions to the k -SAT problem would you expect there to be?
- (b) Plot a graph illustrating how the number of solutions is expected to vary as you vary the ratio clauses to variables (for fixed number of variables)
- (c) At what value of the ratio of number of clauses to number of variables, m/n , would you expect the hardest k -SAT problems to be most often encountered?

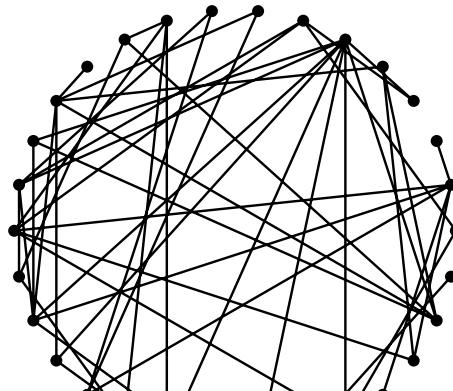


Fig. 7.10 Random graph having $n = 30$ nodes and each of its possible edges included with probability $\frac{\ln n}{n}$

7.4 We generate the random graph shown in Fig. 7.10 by fixing the number of nodes to be $n = 30$ and then including each of its possible $\frac{1}{2}n(n - 1)$ edges with probability $p = \frac{\ln n}{n}$. Use the data presented earlier in this chapter to estimate with what probability you must pick an edge in a $n = 30$ node graph to ensure:

- (a) A greater than 95% chance, on average, that the graph is connected?
- (b) A greater than 95% chance, on average, that the graph is unconnected?
- (c) Is the graph in Fig. 7.10 connected or unconnected? Explain your answer.
- (d) Is the problem of deciding graph connectivity in **P** or **NP**? Justify your answer.

Chapter 8

Quantum Simulation with a Quantum Computer

“Nature isn’t classical, dammit, and if you want to make a simulation of Nature, you’d better make it quantum mechanical, and by golly it’s a wonderful problem, because it doesn’t look so easy.”

– Richard P. Feynman

The main catalyst for funding quantum computers in the U.S.A. came with the publication of Peter Shor’s quantum algorithm for factoring composite integers and computing discrete logarithms, and hence the possibility of breaking public key cryptosystems in polynomial time. Whilst such an application might be of keen interest to intelligence agencies and criminals, it has little significance to the majority of scientists and engineers, nor does it have much commercial value. Industry is not going to adopt quantum computers if all they can do is factor integers and compute discrete logarithms. Moreover, a quantum computer needed to run Shor’s algorithm on problems of practical significance will require thousands of qubits and tens of thousands of quantum gate operations. We are still some way from achieving anything close to this experimentally. Luckily, however, there is another use of quantum computers that has much lower technical hurdles and might be of greater commercial interest. This is the idea of using quantum computers to simulate other quantum mechanical systems exponentially faster than is possible classically. A quantum computer having just 50–100 qubits would be able to outperform any foreseeable classical supercomputer in simulating quantum physical systems. The potential payoff here is considerably more valuable, not to mention scientifically and socially useful, than any business based on code breaking. As Geordie Rose puts it “*Success means a chance to fundamentally change three trillion dollar industries. Total market cap of pharma, chemical, and biotech industries [is] US\$964B plus US\$1153B plus US\$979B equals US\$3.1 trillion dollars.*”

Thus, in this chapter we address the question “How do you simulate the evolution of a quantum system, and extract useful predictions *efficiently*?”. This problem, which is the foundation of several potential new industries including nanotechnology and spintronics, is effectively intractable classically for all but the simplest of quantum systems, but is much easier for a quantum computer.

8.1 Classical Computer Simulations of Quantum Physics

It is common today to hear about supercomputers being used to simulate complex phenomena ranging from the trajectories of hurricanes, the movements of weather patterns, the interactions between biomolecules, global climate change, the stock market, flows across futuristic airplanes, and the folding of proteins. In fact, it is hard to find *any* field of science that has not benefitted greatly from supercomputer simulations.

Few people, however, give much thought to what goes on behind such simulations. Do these simulations capture a complete picture of what is going on? Are the results exact or approximate? Does it matter if we have to replace exact numbers with finite precision approximations to them? Can computers simulate absolutely anything or are there fundamental limits to what can be simulated on either a classical, or quantum, machine?

The essence of performing a quantum simulation is to be able to predict the final state of a quantum system given knowledge of its initial state, $|\psi(0)\rangle$, and the Hamiltonian, \mathcal{H} , by which it is governed. In other words, we need to solve the Schrödinger equation for the particular quantum system:

$$i\hbar \frac{\partial |\psi(t)\rangle}{\partial t} = \mathcal{H}|\psi(t)\rangle \quad (8.1)$$

which has the solution

$$|\psi(t)\rangle = e^{-i\mathcal{H}t/\hbar}|\psi(0)\rangle = U|\psi(0)\rangle \quad (8.2)$$

where $e^{-i\mathcal{H}t} \equiv \mathbb{1} - i(\mathcal{H}t/\hbar) - \frac{1}{2!}i(\mathcal{H}t/\hbar)^2 + \frac{1}{3!}i(\mathcal{H}t/\hbar)^3 - \frac{1}{4!}i(\mathcal{H}t/\hbar)^4 - \dots$ and powers of $\mathcal{H}t/\hbar$ are computed by taking their repeated dot product, e.g., $(\mathcal{H}t/\hbar)^3 = \mathcal{H}t/\hbar \cdot \mathcal{H}t/\hbar \cdot \mathcal{H}t/\hbar$. Thus, if we can compute the matrix exponential of the hermitian matrix describing the Hamiltonian, \mathcal{H} , we can predict the unitary evolution, $U = e^{-i\mathcal{H}t/\hbar}$, by which the system will change, and so predict its state at any time.

Although the Schrödinger equation has been known for the better part of a century [487] it turns out to be impractical to solve it *exactly* analytically in all but the simplest of contexts due to the difficulty of calculating the required matrix exponentials. In part this is because the matrices involved are very large indeed and many terms in the (infinite) series expansion are required to obtain a good enough approximation. Thus, while we can solve the Schrödinger equation of a simple molecule like hydrogen, H₂, exactly analytically, and from the result compute all of its observable properties, we cannot easily replicate this success for other molecules, such as caffeine, C₈H₁₀N₄O₂, that are only moderately more complex.

The infeasibility of analytic methods led scientists to use supercomputers to solve Schrödinger's equation numerically. This can be done using more sophisticated methods to estimate matrix exponentials, or by writing the state vector, $|\psi(t)\rangle$ as the superposition $|\psi(t)\rangle = \sum_{j=1}^N c_j(t)|\psi_j\rangle$ over some complete set of eigenstates, $|\psi_j\rangle$ ($j = 1, 2, \dots, N$), and then solving the implied set of N coupled differential

equations for the amplitudes, $c_j(t)$. Specifically, from the Schrödinger equation and the eigenbasis expansion for the state vector we obtain:

$$\frac{\partial c_j(t)}{\partial t} = -\frac{i}{\hbar} \sum_{k=1}^N \mathcal{H}_{jk} c_k(t) \quad \text{where } \mathcal{H}_{jk} = \langle \psi_j | \mathcal{H} | \psi_k \rangle \quad (8.3)$$

However, such an approach is still costly in terms of both computer time and computer memory.

8.1.1 Exact Simulation and the Problem of Memory

The computational challenge scientists face when simulating quantum systems is daunting. The crux of the problem lies in the amount of memory and computing time needed to solve the Schrödinger equation for any non-trivial atomistic system being simulated. By “size” we could mean the number of atoms or electrons within it, or the number of basis functions that must be used to represent its total wavefunction, or the number of lattice points used to discretize space. Fortunately, all ways of characterizing the “size” of the atomistic system are, for a fixed level of accuracy, proportional to one another and so it matters little which we use.

Thus, even considering a simple n -qubit quantum system, to merely *represent* its state vector requires, in the worst case, the classical computer to store 2^n complex numbers, each with perhaps several digits of precision. Moreover, if we want to predict how this state vector will evolve in space and time, we will have to solve the n -qubit Schrödinger equation, which will entail manipulating matrices containing 2^{2n} complex elements! As you can see n does not have to be that large before it becomes quite impractical to perform the required mathematical operations in a direct way. Worse still, simulating a quantum system containing just one extra qubit, by increasing $n \rightarrow n + 1$, means the length of the state vector *doubles*, and the number of elements in the matrices we manipulate *quadruples*! This limits our ability to determine the physical and chemical properties of nano-devices and materials by ab initio quantum mechanical calculations, even though, in principle, this ought to be possible.

8.1.2 Exact Simulation and the Problem of Entanglement

An even more perplexing problem of using classical computers to simulate quantum systems arises from the possibility that different parts of a composite quantum system may be entangled with one another and hence possess correlations that exceed values that are possible classically. This means that no classical process can truly replicate the statistics of such quantum entanglement unless it is augmented with artificial hidden variables that force the correlations to be as strong as the entanglement requires. In other words there are certain quantum phenomena that are

intrinsically un simulatable classically without the introduction of artificial “hidden variables”. This means that classical computers, even probabilistic classical computers, cannot *truly* simulate all quantum systems without “cheating” by conspiring to arrange, via secret hidden variables, correlations that the classical simulations would not naturally display. Indeed, the minimum amount of communication required to mimic the statistics of an entangled system is used nowadays as one way to quantify the degree of entanglement between sub-systems [77, 120, 495, 515].

8.1.3 Approximate Simulation and the Problem of Fidelity

In response to the intractability of simulating quantum systems exactly using classical computers, scientists have devised an assortment of computational techniques that trade the physical fidelity of a simulation for its computational efficiency (in memory and/or time). These models typically impose strictly incorrect, yet tolerable, simplifying assumptions that allow the mathematical models of composite quantum systems to be reduced to a more tractable form, which render the computation within reach of classical supercomputers. This gives scientists *approximate* solutions that, in many cases, can shed useful insight into complex quantum systems. Some of the standard approximate methods currently used are as follows:

8.1.3.1 Full Configuration Interaction

A “Configuration Interaction” (CI) model represents the wavefunction of an atom or molecule as a linear combination of its ground state and excited states wavefunctions. This allows correlation effects to be included in ab initio quantum mechanical calculations of quantum many-body systems. In turn, the component wavefunctions can each be expanded in terms of so-called “Slater determinants” (discussed in Chap. 9). This means that a CI model describes the total wavefunction as a superposition of molecular orbitals, and each of these molecular orbitals as a linear combination of basis functions. If there are N electrons and K orbitals the so-called “Full Configuration Interaction” will involve a total of $(2K!)/(N!(2K - N)!)$ terms. This number grows so rapidly that the FCI model cannot be used on anything other than the smallest of molecules. Nevertheless, when applicable, the Full Configuration Interaction gives excellent results with respect to the chosen basis set used, which is under the control of the programmer.

8.1.3.2 Tight-Binding

Tight-Binding (TB) methods are often used to determine the electronic properties of crystalline solids. In such materials the nuclei of atoms appear at the lattice sites of the appropriate crystal lattice. The TB model makes the assumption that the Fourier

transform of the Bloch-function, which gives the electron spatial distribution—or “Wannier function”, can be written as a “Linear Combination of Atomic Orbitals” (LCAO). Furthermore, it is assumed that the atomic orbitals decay rapidly on the scale less than the lattice spacing between the nuclei. Hence, to a good approximation, the Hamiltonian for the whole solid can be expressed in terms if the Hamiltonians of the individual atoms that comprise it. As such, the TB method is able to include certain electron-correlation effects if these are already implicit in the mathematical descriptions of the local atomic orbitals.

8.1.3.3 Hartree-Fock

The Hartree-Fock model is a much simpler wavefunction method, similar in spirit to a FCI model, except that it dramatically restricts the number of Slater determinants used. In fact, each energy eigenfunction is assumed to be described using a single Slater determinant. This means that the electrons are assumed to be distributed amongst individual single electron orbitals, and hence electron-electron correlation effects are neglected. It is widely used to compute approximations to the ground state energy and ground state itself. The basis for the method is a proof that any approximation to the true ground state energy has to be higher than the true energy. Consequently, if there are two approximations to the ground state energy and one is lower than the other, it is known which one is better. Moreover, at the true energy the variation in the energy goes to zero. The Hartree-Fock equations were devised by imposing this variational constraint on the ground state energy and requiring that the molecular orbitals be orthonormal.

8.1.3.4 Density Functional Theory

Density Functional Theory (DFT) estimates desired properties of multi-electron systems, such as their ground state energy, via determination of their overall spatial electron density distribution. In DFT electron-electron interactions are only characterized on average, and no strong electron-electron correlation effects are included. The foundation for the method rests on a paper by Hohenberg and Kohn that shows that the ground state energy and other properties of multi-electron systems can be approximated as functionals of the spatial electron density distribution. The computational cost of DFT is considerably less than that of wavefunction methods, such as Hartree-Fock and Full Configuration Interaction, and in many cases of practical interest the answers DFS provides are close enough to reality to provide useful guidance for chemists and material scientists. However, DFT fails badly [94] when the system under study no longer conforms adequately to the implicit assumptions upon which DFT is based, namely, that electron correlation effects can be neglected, e.g., in entangled systems and those possessing van der Waals forces.

Using such approximate techniques, scientists have built many useful models of large molecular systems and made quantitative predictions that are close to observed

values [193]. And others have conceived of possibly designing new materials from ab initio quantum simulations [154, 186].

Notwithstanding these successes, approximate methods don't always get a good enough answer. They seem to break down most noticeably for quantum systems that possess unusually strong electron correlations. This is not surprising because, typically, approximate methods tend to grossly simplify electron interactions. They may impose the assumption that each electron behaves largely independently of the other electrons, or at least sees them as some mean field charge distribution. Unfortunately, in certain materials neither of these assumptions are sufficiently close to being correct for the model to make reliable predictions. In particular, high temperature superconductors, Mott insulators, and quasi-low dimensional materials can all display unexpectedly complex properties that depend crucially on the presence of strong electron correlations.

The 1D Hubbard model is the most widely used model for probing correlated electron systems. There are many approximate solutions to the Hubbard model, the most robust being those based on the Bethe ansatz [60, 319], which work well for defining the ground state and the first few excited states. However, questions of high temperature dynamics are not well described by the ground state symmetries. Worse still, there are no adequate classical approximations to the 2D and 3D Hubbard models. It would seem that such models will require a fundamentally new approach to simulation.

Moreover, recent reviews of old experimental results have even revealed that aspects of certain *bulk* materials can *only* be explained by assuming the existence of *entanglement* on a large scale [85, 513]. Approximate methods used to simulate such materials that fail to account for entanglement will therefore never be able to model such materials adequately.

With the advent of nano-technology scientists are beginning to conceive of more exotic and intelligently designed nano-devices that are engineered to *harness* strong electron correlation effects and entanglement. Physically faithful simulations of such devices will be essential in order to design them correctly and make them robust to noise and imperfections. While supercomputers have proven to be great workhorses for the design and optimization of regular (crystalline) materials and devices, they appear to be quite limited in their ability to simulate such exotic nano-devices due to the amount of memory exact simulations will require, and an inadequate accounting of the physical effects in play. Hence, there appears to be a growing need for a genuinely new strategy for simulating such quantum-engineered devices.

8.1.3.5 Limited Speedup via Parallelization: Amdahl's Law

One might think that *parallelization* is the key to speeding up quantum mechanical simulations on classical computers. Well let's think about this. There is a formula, called Amdahl's law, which allows one to estimate the expected speedup of parallelized implementations of an algorithm relative to the non-parallelized algorithm.

Let us define the running time of the original algorithm as “1” in some appropriate unit. If this algorithm can be parallelized, we will likely find that different parts can be sped up to different degrees. So the question is what is the overall speedup we can expect?

Suppose we benchmark our algorithm and find that it consists of N parts, labeled 1 to N that can be sped up to various degrees. If P_i is the fraction of the net number of computational instructions devoted to performing part i and if S_i is the speedup to part i then Amdahl’s Law states that the net speedup will be:

$$\frac{T_{\text{old}}}{T_{\text{new}}} = \frac{1}{\sum_{i=1}^N P_i / S_i} \quad (8.4)$$

Here is an example. Suppose we have an algorithm that has three parts. The fraction of the total number of instructions in the algorithm consumed by the different parts is $P_1 = 0.3$, $P_2 = 0.5$, and $P_3 = 0.2$, and therefore, as required, $P_1 + P_2 + P_3 = 1$. Now let’s assume the speedup (or slow down) of each part is given by $S_1 = 5 \times$ (a 500% increase), $S_2 = 3.2 \times$ (a 320% increase) and $S_3 = 0.9 \times$ (a 10% slow down). Then the net speedup from parallelization will be $1/(\frac{P_1}{S_1} + \frac{P_2}{S_2} + \frac{P_3}{S_3}) = 1/(\frac{0.3}{5} + \frac{0.5}{3.2} + \frac{0.2}{0.9}) \approx 2.28$.

So parallelization offers some help with obtaining a speedup but it does nothing to eliminate the exponential explosion in memory needed to simulate bigger quantum system. Adding one more qubit doubles the size of the memory needed to encode its state. So parallelism is not sufficient to render exact quantum simulation on classical supercomputers tractable.

8.2 Quantum Computer Simulations of Quantum Physics

Up until the 1980’s people working in theoretical computer science did not have much interaction with those working in theoretical physics. Theoretical computer scientists tended to use pure mathematics, rigorous proofs, and formal systems, whereas theoretical physicists tended to use applied mathematics, approximate models, and less formal arguments. It was fairly unusual, therefore, that in the Spring of 1981, a group of 60 scientists met at MIT’s conference center, Endicott House, in Dedham, Massachusetts to discuss the inter-relationships between physics and computation. The meeting was arranged by Ed Fredkin a famous computer scientist, and Fredkin invited Richard Feynman, a famous Caltech physicist, to come. Initially, Feynman declined but after some cajoling accepted. It is fortunate that Feynman did participate because the ideas he presented at that meeting would prove to be the foundation for nanotechnology and quantum computing.

8.2.1 Feynman Conceives of a Universal Quantum Simulator

“A lecture by Dr. Feynman is a rare treat indeed. For humor and drama, suspense and interest it often rivals Broadway stage plays. And above all, it crackles with clarity. If physics is the underlying ‘melody’ of science, then Dr. Feynman is its most lucid troubadour.”

– Los Angeles Times science editor, 1967.

In his talk Feynman speculated on the efficiency with which computers could simulate different kinds of physical systems. The ideas expressed in this lecture became immortalized subsequently in a scientific paper entitled “Simulating Physics with Computers” that appeared in the International Journal of Theoretical Physics the next year [181].

Feynman realized that in order to make precise statements about the efficiency with which certain computations can be done, that he needed an operational definition of what it meant to be “efficient”. So, in a characteristically pragmatic manner, he adopted the common sense notion that simulation should be deemed efficient if, and only if, the space and time resources the simulation consumed were proportional to the size of the system being simulated. Given this criterion, Feynman posed four fundamental questions concerning the efficiency with which computers can simulate different types of physical system. He asked: *“Can classical physics be simulated efficiently by a classical computer?”*, *“Can quantum physics be simulated efficiently by a classical computer?”*, *“Can quantum physics be simulated efficiently by a quantum computer?”*, and *“Does there exist a universal quantum simulator?”*. Feynman answered some of these questions in full but could only answer others in part. In particular, whilst he was confident a quantum computer could simulate systems of bosons efficiently, he was less certain they could simulate systems of fermions efficiently. Specifically he asked (p. 476 [181]):

“... could we imitate every quantum mechanical system which is discrete and has a finite number of degrees of freedom? I know, almost certainly, that we could do that for any quantum mechanical system which involves Bose particles. I’m not sure whether Fermi particles could be described by such a system. So I leave that open. Well that’s what I meant by a general quantum mechanical simulator. I’m not sure that it’s sufficient, because I am not sure that it takes care of Fermi particles.”

It would take another 15 years before this question was settled definitively by quantum physicists Dan Abrams and Seth Lloyd, who proved that, indeed, quantum computers *can* simulate systems of fermions efficiently [1, 321]. However, we will postpone discussion of this case for now as it is more complicated, but we will revisit it in Sect. 8.4.

8.2.2 Quantum Systems with Local Interactions

For one quantum system to simulate another efficiently *all* stages of the simulation process need to be efficient including:

- Preparing the simulator in an initial state that mirrors the initial state of the system to be simulated.
- Evolving the state.
- Extracting an answer.

Unfortunately, there is a problem with quantum simulation as described above. In Chap. 3 on “Quantum Circuits”, we learned that the number of gates needed to implement an *arbitrary* unitary operator on n -qubits grows exponentially in n . Any such unitary, U say, can always be pictured as the result of a suitably complicated Hamiltonian, i.e., $U = e^{i\mathcal{H}}$ for some Hamiltonian \mathcal{H} . Hence, there must be Hamiltonians (in fact they will be the *majority* of *mathematically* allowed Hamiltonians) that cannot be implemented in a quantum circuit with a number of gates that is only a polynomial in n . This means that there must be many hypothetical quantum systems governed by such complicated Hamiltonians that will not be possible to simulate in polynomial time on any quantum computer!

This would seem to undermine the usefulness of a universal quantum computer as a quantum simulator. Although the quantum computer still knocks down the space (memory) complexity needed to perform the simulation by an exponential factor, it appears it has not made any significant dent on the exponential time complexity!

Fortunately, Nature has been kind to quantum computer scientists. Typically, the kinds of Hamiltonians that arise in practice involve particles that only interact strongly with near neighbors, or, if there are long range interactions, the governing Hamiltonian often has a simple tensor product structure. Either way these kinds of Hamiltonians can be simulated using only polynomially-sized quantum circuits.

8.2.3 Lloyd-Zalka-Wiesner Quantum Simulation Algorithm

Thus, we can make the simplifying assumption that we are dealing with a Hamiltonian having only local interactions, i.e., that \mathcal{H} has the form $\mathcal{H} = \sum_{\ell=1}^L \mathcal{H}_\ell$, where each \mathcal{H}_ℓ involves only few-body interactions.

The Schrödinger equation for a time independent, local Hamiltonian \mathcal{H} is:

$$i\hbar \frac{\partial |\psi\rangle}{\partial t} = \mathcal{H}|\psi\rangle \quad (8.5)$$

with solution $|\psi(t)\rangle = e^{-i\mathcal{H}t/\hbar}|\psi(0)\rangle = U|\psi(0)\rangle$. Thus we now need to find an efficient way to factor the unitary evolution operator U .

Unfortunately, if there is at least one pair of the component Hamiltonians that do not commute, i.e., if there exists \mathcal{H}_ℓ and $\mathcal{H}_{\ell'}$ such that $[\mathcal{H}_\ell, \mathcal{H}_{\ell'}] = \mathcal{H}_\ell\mathcal{H}_{\ell'} - \mathcal{H}_{\ell'}\mathcal{H}_\ell \neq 0$ then mathematically we cannot write $\exp(-i\sum_{\ell=1}^L \mathcal{H}_\ell) = \exp(-i\mathcal{H}_1) \cdot \exp(-i\mathcal{H}_2) \cdots \exp(-i\mathcal{H}_L)$. How then, do you build a circuit for U ?

The trick is to break up the evolution into small increments [1, 532, 556].

$$|\psi(t)\rangle = \underbrace{e^{-i\mathcal{H}\Delta t/\hbar} \cdot e^{-i\mathcal{H}\Delta t/\hbar} \cdots e^{-i\mathcal{H}\Delta t/\hbar}}_{M \text{ factors}} |\psi(0)\rangle = \left(\bigodot_{m=1}^M U(\Delta t) \right) |\psi(0)\rangle \quad (8.6)$$

where $\Delta t = t/M$, and $U(\Delta t) = \exp(-i\mathcal{H}\Delta t/\hbar) \approx \bigodot_{\ell=1}^L e^{-i\mathcal{H}_\ell\Delta t/\hbar} + \mathcal{O}((\Delta t)^2)$.

If each \mathcal{H}_ℓ is local and Δt is small enough, there is an efficient quantum circuit for $\exp(-i\mathcal{H}_\ell \Delta t/\hbar)$. The mathematical basis for this approximation rests on the limit:

$$\lim_{n \rightarrow \infty} (e^{-i\mathcal{H}_1 t/n} \cdot e^{-i\mathcal{H}_2 t/n})^n = e^{-i(\mathcal{H}_1 + \mathcal{H}_2)t} \quad (8.7)$$

Thus the simplest so-called “Trotter” formula is:

$$e^{-i\mathcal{H}\Delta t} = e^{-i\mathcal{H}_1 \Delta t} \cdot e^{-i\mathcal{H}_2 \Delta t} + \mathcal{O}((\Delta t)^2) \quad (8.8)$$

However, higher-order Trotter approximations having a smaller error term can easily be obtained [230, 466, 467].

$$e^{-i\mathcal{H}\Delta t} = e^{-i\mathcal{H}_1 \frac{\Delta t}{2}} \cdot e^{-i\mathcal{H}_2 \Delta t} \cdot e^{-i\mathcal{H}_1 \frac{\Delta t}{2}} + \mathcal{O}((\Delta t)^3) \quad (8.9)$$

$$e^{-i\mathcal{H}\Delta t} = e^{-i\mathcal{H}_2 \frac{\Delta t}{2}} \cdot e^{-i\mathcal{H}_1 \Delta t} \cdot e^{-i\mathcal{H}_2 \frac{\Delta t}{2}} + \mathcal{O}((\Delta t)^3) \quad (8.10)$$

8.3 Extracting Results from Quantum Simulations Efficiently

It is not enough, of course, to simulate a quantum system on a quantum computer. At the end of the day we need to extract some useful information from the end state of the simulation. In general, we can assume this final state will be some superposition, $|\psi(t)\rangle$ say, containing valuable information about the simulated quantum system. Unfortunately, we cannot learn what this final superposition state is merely by reading it (as we would on a classical computer). If we tried to read it we would collapse the superposition into some eigenstate of the observable operator being used. It would therefore take many repetitions of the simulation and subsequent measurement to build up enough information to re-construct the final superposition state using principles of “quantum state tomography”, which is a notoriously costly and inefficient process. In general, we would have to attempt to reconstruct the full density matrix for $|\psi(t)\rangle$, i.e., $\rho(t) = |\psi(t)\rangle\langle\psi(t)|$. If $|\psi(t)\rangle$ were (say) an n -qubit system, it would have a $2^n \times 2^n$ dimensional density matrix, containing 2^{2n} elements. We would, therefore, have to re-synthesize and measure $|\psi(t)\rangle$ on the order of $\mathcal{O}(2^{2n})$ times to be able to estimate this density matrix empirically. Thus naive quantum state tomography is inefficient, and quite impractical for even moderately complex quantum systems. In fact, reconstructing the final state using quantum state tomography would incur as much work as we would have incurred had we performed the simulation classically!

8.3.1 Single Ancilla-Assisted Readout

How then, can there be an advantage to simulating a quantum system on a quantum computer compared to simulating the same system on a classical computer if we

cannot “see” the final result? The answer is that, fortunately, we often only care about learning the expectation value of some operator of interest with respect to the quantum system in question, i.e., $\langle \psi(t) | \mathcal{O} | \psi(t) \rangle$, rather than having to know the final state of the simulated system, $|\psi(t)\rangle$, explicitly. In many cases of practical significance this is information we *can* extract efficiently. The following sections describe some examples of operators whose expectation values can be obtained with surprisingly little effort in comparison to what would be needed for quantum state tomography.

Specifically, as Somma et al. show in [464], suppose we want to know was the expectation value:

$$\mathcal{O} = \langle U^\dagger V \rangle = \langle \psi(t) | U^\dagger V | \psi(t) \rangle \quad (8.11)$$

where U and V unitary, when the system is in state $|\psi(t)\rangle$, but neither know, nor desire to know, $|\psi(t)\rangle$ explicitly. Such an expectation value can be obtained by a procedure which augments the simulated system with a *single* ancilla qubit, and then entangle the simulated system with this ancilla so that a certain expectation value of the ancilla is made to match the expectation value of the operator (on the simulated system) that we seek [354, 464]. The whole process requires re-synthesizing the state $|\psi(t)\rangle$ a number of times that is only polynomial in the size of the simulated system, and each time measuring the expectation value of an operator on only the *single* (ancilla) qubit. It is exponentially easier to estimate the state of a single qubit than that of n -qubits. Hence, this scheme results in an efficient way to extract the expectation value of interest. The procedure, embodied in the quantum circuit shown in Fig. 8.1, works as follows:

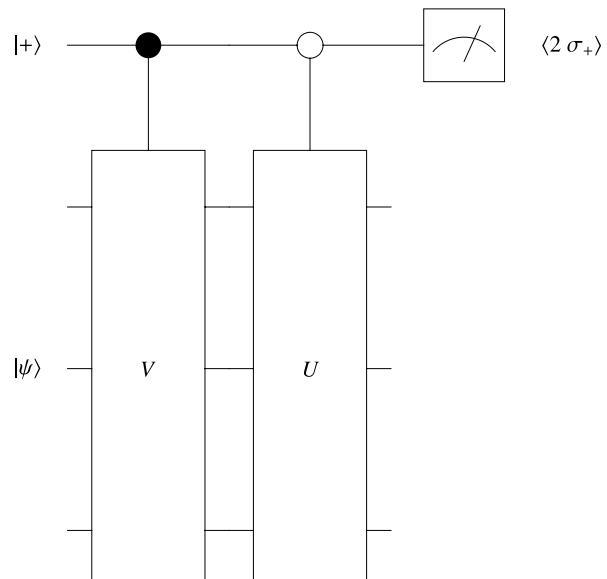


Fig. 8.1 Quantum circuit for ancilla-assisted readout. By determining the expectation value of the operator $\langle 2\sigma_+ \rangle$ for a single ancilla qubit using a number of trials polynomial in n , we can infer the expectation value $\langle \psi | U^\dagger V | \psi \rangle$ of the operator $U^\dagger V$ for an n -qubit quantum system in state $|\psi\rangle$. This means we can extract properties of the result of a quantum simulation without having to know the final state explicitly

Single Ancilla-Assisted Readout

1. Prepare the system in state $|\psi\rangle$
2. Introduce a single ancilla (i.e. extra) qubit and prepare it in equally weighted state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ by acting on state $|0\rangle$ with a single Walsh-Hadamard gate
3. Apply two controlled-gates, $\tilde{V} = |0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes V$ and $\tilde{U} = |0\rangle\langle 0| \otimes U + |1\rangle\langle 1| \otimes \mathbb{1}$. Thus V only acts on the system qubits if the ancilla is in state $|1\rangle$ and U only acts on the system qubits if the ancilla is in state $|0\rangle$.
4. Measure the expectation value of the operator $2\sigma_+ = \sigma_x + i\sigma_y$ of the ancilla qubit (only). This can be done by re-synthesizing $|\psi\rangle$ polynomially many times and measuring the observables σ_x and σ_y alternately on successive trials.
5. After a large but still polynomial number of trials $\langle 2\sigma_+ \rangle = \langle U^\dagger V \rangle$.

The significance of the ancilla-assisted readout is that by monitoring the state of a single ancilla qubit over only polynomially re-syntheses of the state $|\psi\rangle$, the expectation value of a property of $|\psi\rangle$ (an n -qubit system) can be extracted.

8.3.2 Multi-Ancilla-Assisted Readout

The foregoing trick can be extended to allow more complicated expectation values to be extracted efficiently from quantum simulations [380, 464]. Suppose we want to learn the expectation value of an operator that can be written as a *sum* of unitary products, i.e., an operator of the form:

$$\mathcal{O} = \sum_{i=1}^{2^m} a_i U_i^\dagger V_i \quad (8.12)$$

where U_i and V_i are unitary and the coefficients a_i are positive real numbers. One approach would be to use 2^m different single ancilla-assisted readout circuits computing the 2^m different expectation values $\langle U_i^\dagger V_i \rangle$ and then adding up the results weighted by the a_i . But this is not the most efficient strategy. A better approach is to use a quantum circuit such as that shown in Fig. 8.2, which involves the following procedure:

Multi-Ancilla-Assisted Readout Our goal is to estimate the expectation value $\langle \psi | \mathcal{O} | \psi \rangle$ where $\mathcal{O} = \sum_{i=1}^{2^m} a_i U_i^\dagger V_i$ where U_i and V_i are unitary operators.

1. Let $\sum_{i=1}^{2^m} a_i = N$, and then rewrite the operator \mathcal{O} as $\mathcal{O} = N \sum_{i=1}^{2^m} \alpha_i^2 U_i^\dagger V_i$, with $\alpha_i^2 = a_i/N$.
2. Prepare the system in state $|\psi\rangle$, which could be the output from a quantum simulation.
3. Introduce $m + 1$ ancillae qubits (labeled a_1, a_2, \dots, a_{m+1}) each prepared in the state $|0\rangle$.

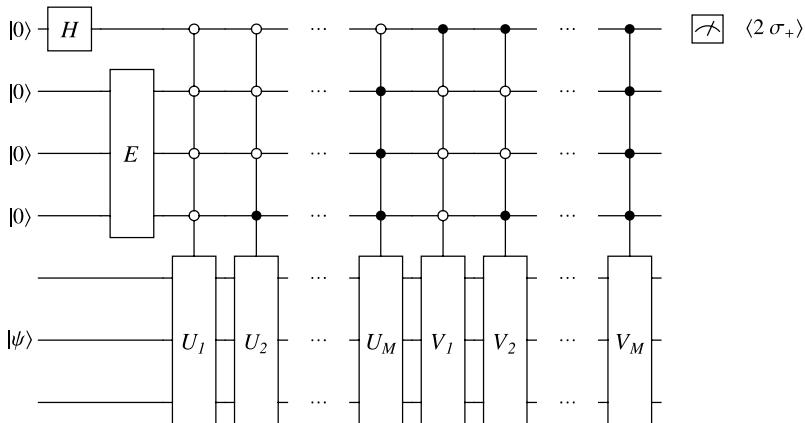


Fig. 8.2 Quantum circuit for ancilla-assisted readout. By determining the expectation value of the operator $\langle 2\sigma_+ \rangle$ for a single ancilla qubit using a number of trials polynomial in n , we can infer the expectation value of $\langle \psi | \sum_{i=1}^{2^m} a_i U_i^\dagger V_i | \psi \rangle$ for the operator $\sum_{i=1}^{2^m} a_i U_i^\dagger V_i$ for an n -qubit system in state $|\psi\rangle$. This means we can extract properties of the result of a quantum simulation without having to know the final state explicitly

4. Apply a Walsh-Hadamard gate to just the first ancilla qubit to put it in the state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.
5. Apply 2^m controlled- U_i gates to the (n) system qubits such that gate U_i acts on the system qubits iff the state of the ancillae qubits is $|0\rangle_1|i\rangle$, where the subscript “1” refers to the state of the first (ancilla) qubit.
6. Likewise, apply 2^m controlled- V_i gates to the (n) system qubits such that gate V_i acts on the system qubits iff the state of the ancillae qubits is $|1\rangle_1|i\rangle$.
7. Apply two controlled-gates, $\tilde{V} = |0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes V$ and $\tilde{U} = |0\rangle\langle 0| \otimes U + |1\rangle\langle 1| \otimes \mathbb{1}$. Thus V only acts in the system qubits if the ancilla is in state $|1\rangle$ and U only acts on the system qubits if the ancilla is in state $|0\rangle$.
8. Measure the expectation value of the operator $2\sigma_+ = \sigma_x + i\sigma_y$ of the single (top-most) ancilla qubit (only). This can be done by re-synthesizing $|\psi\rangle$ polynomially many times and measuring the observables σ_x and σ_y alternately on successive trials.
9. After a large but still polynomial number of trials $\langle 2\sigma_+ \rangle = \langle \sum_{i=1}^{2^m} \alpha_i^2 U_i^\dagger V_i \rangle$.
10. Thus we can obtain the desired expectation value by multiplying the measured value of $\langle 2\sigma_+ \rangle$ by N .

This provides and even more dramatic example of how it is possible to obtain expectation values of observables of quantum systems, without the need to learn the state of those quantum systems explicitly.

8.3.3 Tomography Versus Spectroscopy

The ancilla-assisted readout scheme shown in Fig. 8.1 enables us to determine experimentally the expected value of the operator $\mathcal{O} = U^\dagger V$, where U and V are unitary, when the quantum system is in state $|\psi\rangle$. Conceptually, if $|\psi\rangle$ were an (unknown) state produced as the result of some quantum simulation, we would therefore be able to predict properties of the simulated system. However, as pointed out by Miquel et al. [354], there is another way of thinking about the ancilla-assisted readout circuit that suggests how to extract other information of significance.

The output state of the ancilla depends on *both* the input state $|\psi\rangle$ and the operator \mathcal{O} . Therefore, in principle, if one knew the operator \mathcal{O} one could gain information about $|\psi\rangle$ and, conversely, if one knew the state $|\psi\rangle$, one could gain information about the operator \mathcal{O} , such as characteristics of its spectrum. In other words, the ancilla-assisted readout circuit could be used to perform quantum state tomography (when \mathcal{O} is known and $|\psi\rangle$ is not) and it can be used to perform spectroscopy (when $|\psi\rangle$ is known and \mathcal{O} is not).

For such applications it is useful to consider a slight variation on the ancilla-assisted readout circuit that allows for more general inputs and operators. The resulting circuit is shown in Fig. 8.3. Given a mixed state n -qubit input ρ and an $n + 1$ -qubit controlled- U operator that can be implemented in a polynomially sized quantum circuit, then it can be shown that by monitoring the expectation values of the Pauli spin operators, σ_z and σ_y of the ancilla, i.e., $\langle \sigma_z \rangle$ and $\langle \sigma_y \rangle$, we can obtain information about the real and imaginary parts of the trace of $U\rho$. Specifically,

$$\begin{aligned}\langle \sigma_z \rangle &= \text{Re}[\text{tr}(U\rho)] \\ \langle \sigma_y \rangle &= \text{Im}[\text{tr}(U\rho)]\end{aligned}\quad (8.13)$$

For example, if we pick the input n -qubit state to be maximally mixed, i.e., $\rho = \frac{1}{2^n}$, then the expectation value $\langle \sigma_z \rangle = \text{Re}[\text{tr}(U)]/2^n$, which is proportional to the sum of

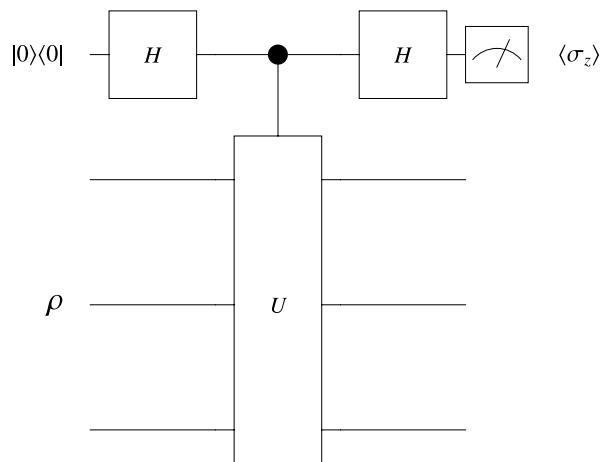


Fig. 8.3 Quantum circuit that can be used for both quantum state tomography and spectroscopy. Only the first qubit (the ancilla) is observed at each trial. By obtaining estimates for $\langle \sigma_z \rangle$ and $\langle \sigma_y \rangle$ we can obtain information about U if ρ is known, or we can obtain information about ρ if U is known. This follows from the fact that $\langle \sigma_z \rangle = \text{Re}[\text{tr}(U\rho)]$ and $\langle \sigma_y \rangle = \text{Im}[\text{tr}(U\rho)]$

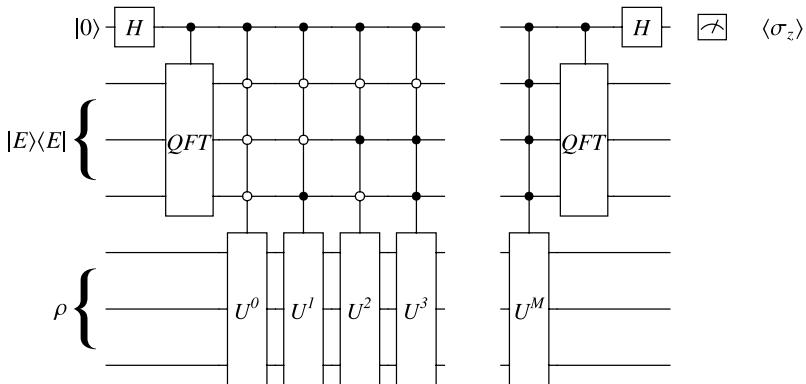


Fig. 8.4 Quantum circuit for determining spectral density in the vicinity of a specific energy

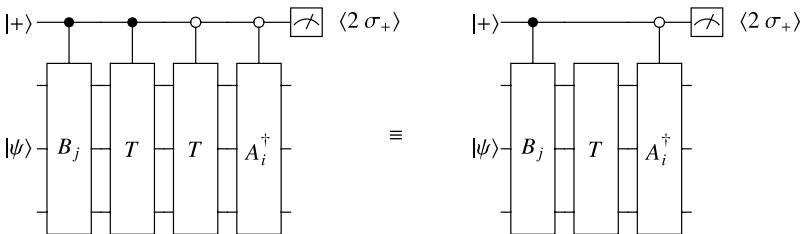


Fig. 8.5 Quantum circuit for measuring correlation functions

the eigenvalues of U , thereby giving some coarse information about the spectrum of U .

More precise characterization of the spectrum of U in the vicinity of specific energies can be obtained from the circuit shown in Fig. 8.4.

8.3.4 Evaluating Correlation Functions

Another type of information that is often sought from simulations of quantum systems is a temporal or spatial correlation function. Mathematically, temporal correlations amount to determining expectation values of operators such as $\langle \psi | T^\dagger A T B | \psi \rangle$, where $A = \sum_i \alpha_i A_i$ and $B = \sum_j \beta_j B_j$ are sums of unitary operators, and $T = \exp(-i\mathcal{H}t)$, with \mathcal{H} being the Hamiltonian, is the temporal evolution operator. Spatial correlation functions are computed similarly except that we take the operator T to be $T = \exp(-ip \cdot x)$, which is the space translation operator. Once posed in this form, it is apparent that correlation functions can be computed using the technique of ancilla-assisted readout introduced earlier by setting $U_i^\dagger = T^\dagger A_i$ and $V_j = T B_j$. Then a circuit for computing one of the components of the desired correlation function using a single ancilla would be that shown in Fig. 8.5. Notice

that the control-on-0 T gate and control-on-1 T gate can be merged since, regardless of the control value, T is to be applied to the system qubits. This reduces, quite significantly, the complexity of the quantum circuit needed to demonstrate the computation of correlation functions.

8.4 Fermionic Simulations on Quantum Computers

In Feynman's original paper on "Simulating Physics with Computers," [181] he alluded to uncertainty in whether or not a true universal quantum simulator might exist. By universal quantum simulator he meant a finite quantum system that could be made to simulate any other quantum system, which is discrete and has a finite number of degrees of freedom, such that the time and space (memory) resources of the simulator are linear in the size of the system being simulated. It is important to appreciate that *all* aspects of the simulation must be able to be done efficiently including its initialization, evolution and measurement. Feynman felt sure such a universal quantum simulator could exist for simulating bosonic systems, doubted whether one could exist for fermionic systems. However, nowhere in the paper did he spell out exactly why he had this doubt. So let us look at this question, and consider the special problems caused by simulating fermionic systems on a computer.

8.4.1 Indistinguishability and Implications for Particle Statistics

In classical physics we are accustomed to assuming that particles, even if they are identical, are nevertheless *distinguishable*. For example, using classical physics thinking, if we rack up a set of identical pool balls in a triangular frame we can tell one ball from another by the fact it has a distinct starting position. Subsequently, as we break, we can track the trajectory of each ball individually. Hence, even though the balls are identical, they remain, in principle, distinguishable. Scaling things up to very large numbers of identical classical particles does not remove this basic property of distinguishability.

Physicists discovered that in a "gas" of such particles, the distribution of their energies will always obey what is known as Maxwell-Boltzmann statistics. This predicts that the statistics of their distribution of energies will be predictable. In particular, the number of particles having energy ϵ_i , i.e., n_{ϵ_i} , will be given by:

$$n_{\epsilon_i} = \frac{g_i}{e^{\beta\epsilon_i + \alpha}} \quad (8.14)$$

where ϵ_i is the energy, g_i is the number of particle states having this energy, and α and β are two parameters related to the physical properties of the system. In particular, $\beta = \frac{1}{kT}$ where k is Boltzmann's constant ($k = 1.3805 \times 10^{-23}$ JK $^{-1}$)

and T is the absolute temperature (measured in degrees Kelvin). In addition, $e^{-\alpha} = N/Z$ where N is the total number of particles and Z is the partition function $Z = \sum_i g_i e^{\beta \epsilon_i}$. It is therefore possible to re-express the Maxwell-Boltzmann distribution in the form:

$$n_{\epsilon_i} = \frac{g_i}{e^{(\epsilon_i - \mu)/kT}} \quad (8.15)$$

where $\mu = -\alpha k T$.

When we go to the quantum scale, we need to re-examine our fundamental notion of distinguishable particles. For example, imagine that we have just two quantum particles of the same type (e.g., two electrons or two photons), which we'll call "particle 1" and "particle 2". Let particle 1 be in state $|\psi_m\rangle$ and let particle 2 be in state $|\psi_n\rangle$. Therefore, using superscripts to label the particles, we are tempted to write their joint state as $|\Psi^{12}\rangle = |\psi_m^1\rangle|\psi_n^2\rangle$. Is this an adequate description of the situation?

The product state representation would be a correct description if the particles were truly distinguishable. However, quantum particles do not have this basic property of guaranteed distinguishability—you cannot zoom in on a quantum particle using a gigantic microscope to see which way it is going without affecting its behavior. For example, a pair of electrons in the same orbital of an atom cannot be distinguished from one another since the orbital merely gives the spatial distribution of the collective electron probability amplitude. So if we know that we have two quantum particles, one of which is in state $|\psi_m\rangle$ and the other of which is in state $|\psi_n\rangle$ we cannot really say which particle is in which state. Therefore, we ought really to write the state as a superposition of the two possibilities, i.e., $|\Psi^{12}\rangle = c_{mn}|\psi_m^1\rangle|\psi_n^2\rangle + c_{nm}|\psi_n^1\rangle|\psi_m^2\rangle$ such that $|c_{mn}|^2 + |c_{nm}|^2 = 1$. Obviously, to be unbiased we should further require $|c_{mn}| = |c_{nm}| = \frac{1}{\sqrt{2}}$. However, this still does not specify things completely. Specifically, there are two simple ways to satisfy the constraint that $|c_{mn}| = |c_{nm}| = \frac{1}{\sqrt{2}}$. We could pick $c_{mn} = -c_{nm} = \frac{1}{\sqrt{2}}$ or we could pick $c_{mn} = +c_{nm} = \frac{1}{\sqrt{2}}$. Depending on our choice the quantum system will turn out to have profoundly different behavior!

8.4.2 Symmetric Versus Anti-Symmetric State Vectors

For example, suppose we had picked $c_{mn} = -c_{nm} = \frac{1}{\sqrt{2}}$ and consider what happens to the total state vector when we interchange particles. Our initial state vector is:

$$|\Psi^{12}\rangle = \frac{1}{\sqrt{2}}(|\psi_m^1\rangle|\psi_n^2\rangle - |\psi_n^1\rangle|\psi_m^2\rangle) \quad (8.16)$$

Now interchange the particles:

$$\begin{aligned}
|\Psi^{21}\rangle &= \frac{1}{\sqrt{2}}(|\psi_m^2\rangle|\psi_n^1\rangle - |\psi_n^2\rangle|\psi_m^1\rangle) \\
&= -\frac{1}{\sqrt{2}}(|\psi_m^1\rangle|\psi_n^2\rangle - |\psi_n^1\rangle|\psi_m^2\rangle) \\
&= -|\Psi^{12}\rangle
\end{aligned} \tag{8.17}$$

The state vector has acquired a minus sign upon interchange of the particles! Any state vector having this property is said to be *anti-symmetric*. Physical systems whose state vectors acquire a minus sign upon interchange of particles are called “fermions”.

Conversely, had we chosen $c_{mn} = +c_{nm} = \frac{1}{\sqrt{2}}$ then $|\Psi^{12}\rangle = \frac{1}{\sqrt{2}}(|\psi_m^1\rangle|\psi_n^2\rangle + |\psi_n^1\rangle|\psi_m^2\rangle)$ on interchange of particles we would have obtained:

$$\begin{aligned}
|\Psi^{21}\rangle &= \frac{1}{\sqrt{2}}(|\psi_m^2\rangle|\psi_n^1\rangle + |\psi_n^2\rangle|\psi_m^1\rangle) \\
&= \frac{1}{\sqrt{2}}(|\psi_m^1\rangle|\psi_n^2\rangle + |\psi_n^1\rangle|\psi_m^2\rangle) \\
&= \frac{1}{\sqrt{2}}(|\psi_n^2\rangle|\psi_m^1\rangle + |\psi_m^2\rangle|\psi_n^1\rangle) \\
&= \frac{1}{\sqrt{2}}(|\psi_m^2\rangle|\psi_n^1\rangle + |\psi_n^2\rangle|\psi_m^1\rangle) = +|\Psi^{12}\rangle
\end{aligned} \tag{8.18}$$

In this case the state vector remains unchanged upon interchange of a pair of particles. Any state vector having this property is said to be *symmetric*. Physical systems whose state vectors are symmetric are called “bosons”.

8.4.3 Bosons and Fermions

Thus, in quantum mechanics there are fundamentally two kinds of particles—fermions and bosons—that differ in whether they need to be described using anti-symmetric or symmetric state vectors. It turns out that particles having half integer spin (e.g., protons, neutrons, and electrons) are all fermions (and are therefore described by anti-symmetric state vectors), whereas particles having integer spin (such as photons) are all bosons (and are therefore described by symmetric state vectors). Moreover, a composite quantum object that contains an *odd* number of fermions is a fermion, e.g., a carbon-13 nucleus consisting of 6 protons and 7 neutrons (i.e., 13 fermions). Conversely, a composite object containing an even number of fermions is a boson. For example, a carbon-12 nucleus consisting of 6 protons and 6 neutrons (i.e., 12 fermions) is therefore a boson. Thus fermions and bosons are not restricted to merely elementary particles. Indeed, there are now known to be many other “quasi-particles” such as Cooper-pairs, electron-hole pairs, plasmons, magnons, and polarons.

As an aside we mention that if particles are confined to move in only *two* dimensions then a third and highly unusual class of quasi-particles, called “anyons”, is possible [533]. Whereas the state vectors of bosons and fermions pick up either a +1 or -1 phase factor upon interchange of particles, the state vectors describing anyons can pick up *any* phase factor $\exp(i\phi)$ upon interchange of particles. Hence the name “*any-on*”. Indeed, there is now a model of so-called “topological quantum computation” based on the braiding of anyons on a two dimensional surface, which we discuss in Chap. 15.

However, in *three* dimensions all particles and quasi-particles are either fermions or bosons, and are classified as one or the other depending on how their state vector transforms under particle interchange. This fundamental difference in the symmetry properties of the state vector causes profound differences in the properties bosons and fermions possess at very low temperatures and high particle densities.

8.4.4 Bose-Einstein Statistics

Bosons always have an integer value of spin, and the distribution of their energies obeys Bose-Einstein particle statistics:

$$n_{\epsilon_i} = \frac{g_i}{e^{(\epsilon_i - \mu)/kT} - 1} \quad (8.19)$$

At high temperatures this distribution converges to the Maxwell-Boltzmann distribution. But at low temperatures, and high particle densities, the bosonic features dominate behavior. In particular, bosons have the property that an arbitrarily large number may occupy the same quantum state at the same time. Thus, when a collection of bosons are cooled, they fall into the *same* quantum state, creating a so-called Bose-Einstein condensate. Ultimately, this phenomenon explains certain anomalous behaviors in low temperature systems including superfluidity and superconductivity (due to Cooper pairs).

8.4.5 Pauli Exclusion Principle and Fermi-Dirac Statistics

If bosons can all occupy the same state, can fermions do likewise? Let us consider a pair of fermions, such as electrons, described by the anti-symmetric state vector:

$$|\Psi^{12}\rangle = \frac{1}{\sqrt{2}}(|\psi_m^1\rangle|\psi_n^2\rangle - |\psi_n^1\rangle|\psi_m^2\rangle) \quad (8.20)$$

What happens if $|\psi_m\rangle = |\psi_n\rangle$? In this case, we would obtain $|\Psi^{12}\rangle = \frac{1}{\sqrt{2}}(|\psi_m^1\rangle|\psi_m^2\rangle - |\psi_m^1\rangle|\psi_m^2\rangle) = 0$. This is not a valid state! So the anti-symmetric nature of the state vector for a fermion tells us that we are never allowed to have a situation in which $|\psi_m\rangle = |\psi_n\rangle$. That is, no two fermions may ever be in the same quantum state at the same time!

Pauli Exclusion Principle No two identical fermions may occupy the same quantum state simultaneously.

As an illustrative example, consider an atom. The state of each electron in an atom is completely. The azimuthal quantum number specifies the shape of the orbital with $\ell = 0$ for a spherical s -orbital, $\ell = 1$ for a dumbbell shaped p -orbital, $\ell = 2$ for a dumbbell with doughnut shaped d -orbital etc. The magnetic quantum number specifies the projection of the orbital angular momentum along an axis. And the spin quantum number specifies the angular momentum of the electron. The allowed values for these quantum numbers are interrelated: $0 \leq \ell \leq n - 1$, $-\ell \leq m_\ell \leq \ell$, and $m_s = -\frac{1}{2}$ or $+\frac{1}{2}$, so for a given principal quantum number (the “electron energy level” or “shell”) there are a finite number of distinct and allowed combinations for the other three quantum numbers. The Pauli Exclusion Principle says that no two electrons in the same atom may have the same values for all four quantum numbers. Hence, as we move up the Periodic Table by considering nuclei with progressively more protons, to keep the atom electrically neutral the number of electrons must grow to match the number of protons. But by the Pauli Exclusion Principle the extra electrons cannot pile up in the same state. As electrons are added they must fill up a shell by exhausting all distinct combinations for the four quantum numbers. Having done so, the next electron added must start to fill up the next shell etc.

The electronic structure of an atom depends on this property. For example, in an atom the electrons (which are spin- $\frac{1}{2}$ particles and hence fermions) lie in orbitals, which constitute different quantum mechanical states of a certain energy, such that there can be at most two electrons per orbital. These two electrons cannot be in the same quantum state (i.e., have identical values for all their quantum numbers) even though they are in the same orbital. So they must differ in the value of some quantum number. Indeed they do. The two electrons will have opposing spins, thereby enforcing their anti-social nature as fermions. Using this rule we can build up the electronic structure of an atom. The orbitals fill up two electrons at a time. Orbitals of the same energy constitute an energy “shell” such that the n -th shell has at most $2n^2$ electrons in it, and then electrons need to go to the next shell etc. This electron distribution is what gives atoms their chemical properties. Hence, this anti-social characteristic of fermions essential to account for electronic structure of atoms.

In general, fermions have half-integer values for their spin and need to be described using anti-symmetric state vectors. Hence they all exhibit this anti-social tendency of avoiding the same quantum state at the same time. This behavior changes the statistics of the energy distribution of fermions compared with the classical Maxwell-Boltzmann distribution. Fermions, it turns out, must obey Fermi-Dirac statistics:

$$n_{\epsilon_i} = \frac{g_i}{e^{(\epsilon_i - \mu)/kT} + 1} \quad (8.21)$$

At high temperatures this distribution converges to the Maxwell-Boltzmann distribution. But at low temperatures, and high particle densities, the fermionic features

dominate behavior. For example, certain types of white dwarf stars are able to resist collapse due to pressure exerted because of Fermi-Dirac statistical effects.

Thus, we have now seen that the distribution laws for the energies of bosons and fermions are quite different and as a consequence bosonic systems and fermionic systems display quite different particle statistics at low temperatures and high densities. If we are to have a universal quantum simulator this means that the machine will have to mimic the right kind of particle statistics for the quantum system being simulated. It is not obvious *a priori* that this can be done. For example, if our quantum simulator uses bosons (e.g., photons) can it simulate fermionic systems (e.g., electrons) correctly and vice versa? Even Feynman in his original “Simulating Physics with Computers” paper was unsure whether this could be done and whether it was possible to have a *universal* quantum simulator. The trick, it turns out, is to use the Jordan-Wigner transformation.

8.4.6 Fermionic Simulations via the Jordan-Wigner Transformation

We would like to be able to use a universal quantum simulator to simulate all the various kinds of quantum objects described above—elementary particles, composite particles, and quasi-particles—regardless of whether they are bosons, fermions or anyons. Unfortunately, the basic problem of conceiving of a universal quantum simulator is that the quantum particles we want to simulate may obey completely different operator algebras from the quantum particles with which the simulator is built. How then can we be sure that one quantum algebra can be used to simulate another, and even if it can, that it can be done efficiently?

This is the crux of the problem to which Feynman alluded with fermions. As we saw above, fermions must be described by anti-symmetric state vectors. Therefore, if we want to simulate a fermionic quantum system, we are obliged to initialize our quantum simulator in an anti-symmetric quantum state. However, for an n particle system, such a state will contain $n!$ states in superposition. It was not obvious to Feynman, at the time he wrote his classic “Simulating Physics with Computers” paper, that there was an efficient procedure to create this initial anti-symmetric state having factorially many components. Hence, his doubt regarding the efficiency of fermionic simulations on a quantum computer. The question remained open until Dan Abrams and Seth Lloyd showed that this could be done [1]. Subsequently these ideas were generalized and embellished by the work of G. Ortiz, J.E. Gubernatis, E. Knill, R. Laflamme, and R. Somma [375, 376, 464], and extended to quantum simulations of topological field theories by Michael Freedman, Alexei Kitaev, and Zhenghan Wang [189]. Collectively, these works have made great progress in pushing forward the frontiers of quantum simulation.

The key to quantum simulation is therefore to map the operator algebra of the target quantum system to be simulated into the operator algebra of the quantum simulator, e.g., qubits (spin- $\frac{1}{2}$ particles) and unitary gates, perform the simulation

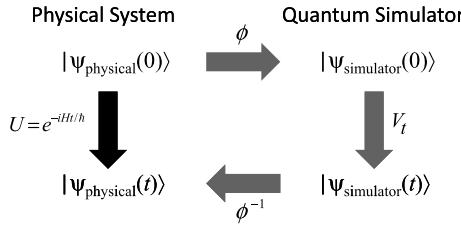


Fig. 8.6 Quantum simulation can be done provided we can map the operator algebra of the target quantum system into the operator algebra of the quantum simulator, e.g., qubits (spin- $\frac{1}{2}$ particles) and unitary gates, perform the simulation in the simulator’s algebra, and then invert the mapping to translate the answer back into the original operator algebra

in the simulator’s algebra, and then invert the mapping to translate the simulation result back into the original operator algebra. This basic strategy is illustrated in Fig. 8.6. By being able to interpret quantum mechanical calculations in as equivalent computations in different operator algebras, we can sometimes pick the algebra that makes the calculation easiest, translate the problem to that algebra, solve the problem, and then translate the result back to the original operator algebra.

A good example, is provided by trying to simulate the quantum mechanical behavior of a two-dimensional lattice of spinless fermions. The situation is described using an algebra of creation and annihilation operators, which create fermions on one site and annihilate them on another as a way of modeling how the fermions are allowed to hop around the lattice. The anti-social nature of the fermions, i.e., the fact that no two can share the same values for all their quantum numbers, restricts their allowed spatial distributions on the lattice. Moreover, the anti-symmetric nature of their state vector means that it must acquire a minus sign each time a pair of fermions are interchanged on the lattice.

The key to simulating such a *fermionic* system (which obeys the creation/annihilation operator algebra) using a simulator based on the standard quantum circuit model (which obeys the Pauli algebra of spin- $\frac{1}{2}$ particles) is to map the fermionic algebra into the Pauli algebra, simulate in the translated system in the Pauli algebra, and then map the result back to the fermionic algebra. In this case the relevant mapping is performed by way of the Jordan-Wigner transformation, which is defined as follows:

$$\begin{aligned} c_j &\rightarrow \left(\prod_{\ell=1}^{j-1} -\sigma_z^\ell \right) \sigma_-^j \\ c_j^\dagger &\rightarrow \left(\prod_{\ell=1}^{j-1} -\sigma_z^\ell \right) \sigma_+^j \end{aligned} \tag{8.22}$$

where

$$\begin{aligned} \sigma_+ &= \sigma_x + i\sigma_y \\ \sigma_- &= \sigma_x - i\sigma_y \end{aligned} \tag{8.23}$$

and σ_x , σ_y and σ_z are the usual Pauli matrices, which obey the SU(2) algebra, whereas a_j and a_j^\dagger are the annihilation and creation operators for a fermion at site j , which obey the fermionic anti-commutation algebra:

$$\begin{aligned}\{a_i, a_j\} &= 0 \\ \{a_i^\dagger, a_j^\dagger\} &= 0 \\ \{a_i^\dagger, a_j\} &= \delta_{ij}\end{aligned}\tag{8.24}$$

where $\{A, B\} = AB + BA$ is the anti-commutator of operators A and B . Once, in the Pauli-algebra of spin- $\frac{1}{2}$ particles, the simulation can be performed using the methods of Sect. 8.2.3. Moreover, the two-dimensional lattice of $N_x \times N_y$ sites can be mapped into a one-dimensional chain of sites by mapping the site with coordinate (ℓ, m) into the site $j = m + (\ell - 1)N_x$ on a one-dimensional chain, where $1 \leq \ell \leq N_y$ and $1 \leq m \leq N_x$. Hence, the fermionic algebra can be simulated using the Pauli-algebra.

8.4.7 Fermionic Simulations via Transformation to Non-interacting Hamiltonians

Recently, another approach to quantum simulation, especially suited to strongly correlated electron systems, was discovered. It pertains, amongst other things, to the famous XY -model [514]. The scheme uses a *fixed* quantum circuit to transform the Hamiltonian of a system having strong electron correlations into one corresponding to non-interacting particles, the subsequent simulation of which is trivial. It works because this fixed circuit has the effect of diagonalizing the strong electron correlation Hamiltonian. Moreover, this mapping is exact, and obviates the need to use any approximations, such as the Trotter formula, in simulating the Hamiltonian. This scheme is therefore conceptually simpler than the approaches discussed earlier, and ought to lead to higher fidelity simulations.

We start with the XY -Hamiltonian of a system having strong electron correlations:

$$\begin{aligned}\mathcal{H}_{XY}[\sigma] = \sum_{i=1}^n &\left(\frac{1+\gamma}{2} \sigma_i^x \sigma_{i+1}^x + \frac{1-\gamma}{2} \sigma_i^y \sigma_{i+1}^y \right) \\ &+ \lambda \sum_{i=1}^n \sigma_i^z + \frac{1+\gamma}{2} \sigma_1^y \sigma_2^z \cdots \sigma_{n-1}^z \sigma_n^y + \frac{1-\gamma}{2} \sigma_1^x \sigma_2^z \cdots \sigma_{n-1}^z \sigma_n^x\end{aligned}\tag{8.25}$$

γ is the anisotropy and λ an external transverse magnetic field. These last two terms arise from boundary conditions and are not important in the large n limit.

Suppose there is a unitary matrix, U_{dis} , which can be implemented in only polynomially many gates, that disentangles the dynamics. That is,

$$\mathcal{H}_{XY} = U_{\text{dis}} \cdot \tilde{\mathcal{H}} \cdot U_{\text{dis}}^\dagger \quad (8.26)$$

where $\tilde{\mathcal{H}}$ is the non-interacting Hamiltonian and U_{dis} is the unitary matrix corresponding to the fixed quantum circuit that diagonalizes the strong-electron correlation Hamiltonian \mathcal{H}_{XY} . If such a matrix exists it will then be possible to synthesize *arbitrary excited states* (not just ground states) of the strong electron correlation systems (a very difficult task ordinarily) merely by preparing an appropriate *product* state, and then applying U_{dis} . Likewise, the *time evolution* of a strong electron correlation system (an even harder task!) can be simulated from:

$$e^{-i\mathcal{H}_{XY}t} = U_{\text{dis}} \cdot e^{-i\tilde{\mathcal{H}}t} \cdot U_{\text{dis}}^\dagger \quad (8.27)$$

Frank Verstraete, Ignacio Cirac, and Jose Latorre discovered just such a U_{dis} having the remarkable property of being a fixed operation able to disentangle the dynamics of a strong electron correlation system [514]. Their recipe for disentangling the XY -Hamiltonian proceeds by transforming the strong electron correlation Hamiltonian into a disentangled form by applying a sequence of steps

$$\mathcal{H}_{XY} \xrightarrow{\text{Jordan-Wigner}} \mathcal{H}_2[c] \xrightarrow{\text{QFT}} \mathcal{H}_3[b] \xrightarrow{\text{Bogoliubov}} \mathcal{H}_4[a] \quad (8.28)$$

where $\mathcal{H}_4[a]$ is unentangled.

Direct Fermionic Simulation via Mapping to Non-interacting Hamiltonians

1. Start with the Hamiltonian $\mathcal{H}_{XY}[\sigma]$. This is expressed in terms of spin operator algebra.
2. Re-express the Hamiltonian $\mathcal{H}_{XY}[\sigma]$ in terms of fermionic modes by using the Jordan-Wigner transformation. This is not a physical operation it is merely using mathematical transformation to re-represent $\mathcal{H}_{XY}[\sigma]$ (in a spin-operator algebra) as $\mathcal{H}_2[c]$ (in a fermionic mode representation).
3. Likewise, also use the Jordan-Wigner transformation to re-express the input state of the system (represented initially by spin- $\frac{1}{2}$ particles) as a fermionic state:

$$|\psi\rangle = \sum_{i_1, i_2, \dots, i_n=0,1} \psi_{i_1, i_2, \dots, i_n} |i_1, i_2, \dots, i_n\rangle$$

$$\xrightarrow{\text{Jordan-Wigner}} \sum_{i_1, i_2, \dots, i_n=0,1} \psi_{i_1, i_2, \dots, i_n} (c_1^\dagger)^{i_1} (c_2^\dagger)^{i_2} \cdots (c_n^\dagger)^{i_n} |\text{vacuum}\rangle \quad (8.29)$$

4. Apply the QFT in the “fermionic mode” representation. That is, apply the transformation $b_k = \frac{1}{\sqrt{n}} \sum_{j=1}^n \exp(i2\pi jk/n) c_j$ for $k = -\frac{n}{2} + 1, \dots, \frac{n}{2}$. This operation maps $\mathcal{H}_2[c]$ into a momentum space representation $\mathcal{H}_3[b]$.

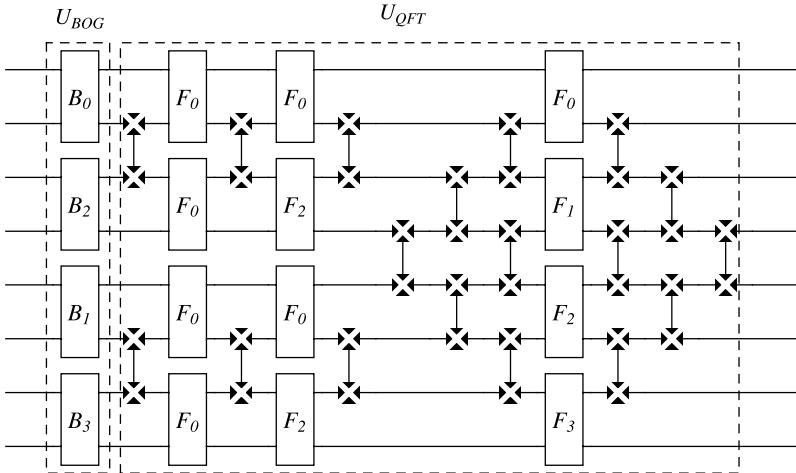


Fig. 8.7 Fermionic quantum simulation by transforming to an interaction-free Hamiltonian. This circuit diagonalizes the Hamiltonian of the XY-model for eight sites. The circuit performs a Bogoliubov transformation (involving the B gates), followed by a QFT (involving the F_k and fermionic SWAP gates). The circuit icon for the latter is taken here to be that of a SWAP gate with blacked in ends. In the disentangling direction, $\mathcal{H}_{XY} \rightarrow \mathcal{H}_4[a]$, this circuit would be used right-to-left. It is written left-to-right because it was derived by analyzing the mapping $\mathcal{H}_4[a] \rightarrow \mathcal{H}_{XY}$. Moreover, the full circuit needs some additional qubit permutation gates to re-order them in the same order they began

5. Finally, to completely disentangle $\mathcal{H}_{XY}[\sigma]$ we map $\mathcal{H}_3[b]$ into a momentum-dependent mixture of modes. Specifically, we obtain $\mathcal{H}_4[a] = \sum_{k=-(n/2)+1}^{n/2} \omega_k a_k^\dagger a_k$ where $a_k = \cos(\theta_k/2) b_k - i \sin(\theta_k/2) b_{-k}^\dagger$. The angles θ_k are given by

$$\theta_k = \arccos\left(\frac{-\lambda + \cos(\frac{2\pi k}{n})}{\sqrt{[\lambda - \cos(\frac{2\pi k}{n})]^2 + \gamma^2 \sin^2(\frac{2\pi k}{n})}}\right) \quad (8.30)$$

6. Thus we arrive at a disentangled Hamiltonian $\mathcal{H}_4[a] = \sum_{k=-(n/2)+1}^{n/2} \omega_k a_k^\dagger a_k$ where $\omega_k = \sqrt{[\lambda - \cos(\frac{2\pi k}{n})]^2 + \gamma^2 \sin^2(\frac{2\pi k}{n})}$. This is a sum of non-interacting terms whose spectrum is equivalent to a non-interacting Hamiltonian $\tilde{\mathcal{H}} = \sum_i \omega_i \sigma_i^z$.

The quantum circuit that performs the QFT and Bogoliubov transformations is shown in Fig. 8.7. In Fig. 8.7 the gate icons that resemble SWAP gates with blacked

in ends, represents a fermionic-SWAP gate, defined as:

$$\text{fermionicSWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad (8.31)$$

the F_k are the gates:

$$F_k = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{\alpha(k)}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{\alpha(k)}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & -\alpha(k) \end{pmatrix} \quad (8.32)$$

with $\alpha(k) = \exp(i 2\pi k/n)$. The combination of F_k gates and fermionic SWAP gates comprise the QFT operation. Similarly, the gates

$$B_k = \begin{pmatrix} \cos \theta_k & 0 & 0 & i \sin \theta_k \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ i \sin \theta_k & 0 & 0 & \cos \theta_k \end{pmatrix} \quad (8.33)$$

collectively implement the Bogoliubov transformation where

$$\theta_k = \arccos\left(\frac{-\lambda + \cos(\frac{2\pi k}{n})}{\sqrt{[\lambda - \cos(\frac{2\pi k}{n})]^2 + \gamma^2 \sin^2(\frac{2\pi k}{n})}}\right) \quad (8.34)$$

Quantum simulation via mapping to a non-interacting Hamiltonian represents a fresh perspective on fermionic simulation and appears to be implementable using even fewer qubits than conventional quantum simulation. In fact, it seems likely that such a scheme will be the basis for the first true quantum simulation to exceed the capabilities of any classical supercomputer simulation.

8.5 Summary

Exact numerical simulations of quantum systems are intractable using classical computers for all but trivial systems. Approximate simulations can omit phenomena that are of critical importance to the system being simulated. And, last but not least, certain quantum phenomena are not intrinsically simulatable by any classical device unless we introduce artificial hidden variables. A fundamentally new approach to the simulation of quantum systems is needed.

Quantum simulation provides such a breakthrough. The Lloyd-Zalka-Wiesner algorithm [1, 532, 556] shows that it is possible, in principle, to simulate any Hamiltonian by using the Trotter product formula. However, if we want to simulate quantum

systems that obey an operator algebra other than that native to the simulator, then we need to find the mapping from the target operator algebra to the Pauli-algebra of spin- $\frac{1}{2}$ particles, perform the simulation in the Pauli-algebra, and then map the result back to the target operator algebra. The most recent methods exploit these mappings between operator algebras to simulate a fermionic system by mapping it to a non-interacting Hamiltonian for which simulation is easy.

It is believed that a quantum computer having just 50–100 qubits would be able to outperform any foreseeable classical supercomputer in simulating quantum physical systems. Such special-purpose quantum simulators may, in fact, be the only way to design new nanotechnology devices that deliberately seek to exploit strong electron correlations and other exotic quantum effects, and which are beyond the reach of classical supercomputer-based simulation. Thus quantum simulation has the potential to have a far greater impact on science and engineering, and a far greater value to society, than that of quantum factoring engines. It is a pity so many resources have been directed towards the goal of achieving a quantum factoring engine when quantum simulation engines have so much greater potential.

8.6 Exercises

8.1 Matrix exponentials arise routinely in quantum computing, and certainly in quantum simulation. One way to compute a matrix exponential is to substitute the relevant *matrix* for all powers of the *variable* in the series expansion of the exponential function in the variable. If the order of the expansion is high enough you can often spot a pattern and thereby guess the correct closed form for the exact matrix exponential. Answer the following questions to obtain the exact matrix exponential $e^{-i\mathcal{H}}$ where \mathcal{H} is the XY -Hamiltonian. We will assume we are working in units of $\hbar = 1$.

- (a) Write down the series expansion of e^ω around $\omega = 0$ to order n .
- (b) Write down the series expansion of $\sin \omega$ around $\omega = 0$ to order n .
- (c) Write down the series expansion of $\cos \omega$ around $\omega = 0$ to order n .
- (d) Compute a series that *approximates* e^Ω to order n by substituting the matrix Ω for *all* power of the variable ω in the series you obtained in part (a) above. [Hint: be sure to account for *all* powers of ω and realize that $\omega^k \rightarrow \underbrace{\Omega \cdot \Omega \cdots \cdot \Omega}_k$]
- (e) Let \mathcal{H} be the XY -Hamiltonian $\mathcal{H} = \alpha X \otimes X + \beta Y \otimes Y$. Using the formulae you obtained in part (d) above, approximate the matrix exponential of the matrix $-i\mathcal{H}$ to order 7.
- (f) What is the significance of the matrix defined in part (e) from a quantum simulation perspective?
- (g) Look at the structure of the matrix elements you obtained in part (e) and use the formulae you obtained in parts (b) and (c) to recognize the closed form functions corresponding to the matrix elements of $e^{-i\mathcal{H}}$. The result is the *exact* matrix exponential $e^{-i\mathcal{H}}$.

8.2 If Exercise 8.1 was too easy, try the following variant on it. Answer the following questions to obtain the exact matrix exponential $e^{-i\mathcal{H}}$ where \mathcal{H} is the Hamiltonian defined below. We will assume we are working in units of $\hbar = 1$.

- (a) Write down the series expansion of e^ω around $\omega = 0$ to order n .
- (b) Write down the series expansion of $\sinh \omega$ around $\omega = 0$ to order n .
- (c) Write down the series expansion of $\cosh \omega$ around $\omega = 0$ to order n .
- (d) Compute a series that approximates the matrix exponential e^Ω to order n by substituting the matrix Ω for all power of the variable ω in the series you obtained in part (a) above.
- (e) Let \mathcal{H} be the Hamiltonian $\mathcal{H} = \alpha X \otimes X + \beta Z \otimes \mathbb{1} + \gamma \mathbb{1} \otimes Z$. Using the formulae you obtained in part (d) above, approximate $e^{i\mathcal{H}}$ to order 7.
- (g) Use your results in parts (b) and (c) to predict the exact closed form of the matrix exponential $e^{-i(\alpha X \otimes X + \beta Z \otimes \mathbb{1} + \gamma \mathbb{1} \otimes Z)}$.

8.3 In Sect. 2.9 we encountered the Krauss-Cirac decomposition of a maximally general 2-qubit unitary matrix. The core entangling operation in that decomposition, i.e., $N(a, b, c)$, was defined via the matrix exponential $N(a, b, c) = e^{i(aX \otimes X + bY \otimes Y + cZ \otimes Z)}$, where X , Y , and Z are the Pauli matrices. In Sect. 2.9 we stated the closed form of $N(a, b, c)$ without proof. Use the following steps to confirm that the stated the form given was correct.

- (a) Write down the series expansion of e^ω around $\omega = 0$ to order n .
- (b) Write down the series expansion of $\sin \omega$ around $\omega = 0$ to order n .
- (c) Write down the series expansion of $\cos \omega$ around $\omega = 0$ to order n .
- (d) Compute a series that approximates the matrix exponential e^Ω to order n by substituting the matrix Ω for all power of the variable ω in the series you obtained in part (a) above.
- (e) Let \mathcal{H} be the Hamiltonian $\mathcal{H} = aX \otimes X + bY \otimes Y + cZ \otimes Z$. Using the formulae you obtained in part (d) above, approximate $e^{i\mathcal{H}}$ to order 7.
- (f) Use your results in parts (b) and (c) to predict the exact closed form of the matrix exponential $e^{-i(aX \otimes X + bY \otimes Y + cZ \otimes Z)}$, and verify it matches the unitary matrix $N(a, b, c)$ of Sect. 2.9. This is the core entangling operation within any maximally general 2-qubit gate.

8.4 In quantum simulation we sometimes approximate a matrix exponential of the form $e^{-i(\mathcal{H}_1 + \mathcal{H}_2)\Delta t}$ such that $[\mathcal{H}_1, \mathcal{H}_2] \neq 0$ by the Trotter formula $e^{-i\mathcal{H}_1 \frac{\Delta t}{2}} \cdot e^{-i\mathcal{H}_2 \Delta t} \cdot e^{-i\mathcal{H}_1 \frac{\Delta t}{2}}$. Check the veracity of this approximation by answering the following questions.

- (a) Let $\mathcal{H}_1 = \frac{1}{3}X \otimes X + \frac{1}{5}Y \otimes Y$ and $\mathcal{H}_2 = \frac{1}{2}\mathbb{1} \otimes Z + \frac{1}{7}X \otimes \mathbb{1}$. Prove that:

$$[\mathcal{H}_1, \mathcal{H}_2] = \begin{pmatrix} 0 & -\frac{2}{35} & 0 & -\frac{2}{15} \\ \frac{2}{35} & 0 & \frac{8}{15} & 0 \\ 0 & -\frac{8}{15} & 0 & \frac{2}{35} \\ \frac{2}{15} & 0 & -\frac{2}{35} & 0 \end{pmatrix} \neq 0 \quad (8.35)$$

(b) For $\Delta t = 1$ prove

$$\begin{aligned} & e^{-i(\mathcal{H}_1 + \mathcal{H}_2)\Delta t} \\ &= \begin{pmatrix} 0.861 - 0.473i & -0.044 + 0.005i & -0.067 - 0.117i & -0.125i \\ -0.044 + 0.005i & 0.736 + 0.452i & -0.483i & 0.067 - 0.117i \\ -0.067 - 0.117i & -0.483i & 0.736 - 0.452i & -0.044 - 0.005i \\ -0.125i & 0.067 - 0.117i & -0.044 - 0.005i & 0.861 + 0.473i \end{pmatrix} \\ & e^{-i\mathcal{H}_1 \frac{\Delta t}{2}} \cdot e^{-i\mathcal{H}_2 \Delta t} \cdot e^{-i\mathcal{H}_1 \frac{\Delta t}{2}} \\ &= \begin{pmatrix} 0.861 - 0.475i & -0.041 + 0.014i & -0.067 - 0.118i & -0.115i \\ -0.041 + 0.014i & 0.748 + 0.475i & -0.442i & 0.067 - 0.118i \\ -0.067 - 0.118i & -0.442i & 0.748 - 0.475i & -0.041 - 0.014i \\ -0.115i & 0.067 - 0.118i & -0.041 - 0.014i & 0.861 + 0.475i \end{pmatrix} \end{aligned}$$

(c) For $\Delta t = 0.1$, i.e., a time times smaller time step, that the approximation becomes even better. In particular, prove:

$$\begin{aligned} & e^{-i(\mathcal{H}_1 + \mathcal{H}_2)\Delta t} \\ &= \begin{pmatrix} 0.999 - 0.05i & 0 & -0.001 - 0.014i & -0.013i \\ 0 & 0.997 + 0.05i & -0.053i & 0.001 - 0.014i \\ -0.001 - 0.014i & -0.053i & 0.997 - 0.05i & 0 \\ -0.013i & 0.001 - 0.014i & 0 & 0.999 + 0.05i \end{pmatrix} \\ & e^{-i\mathcal{H}_1 \frac{\Delta t}{2}} \cdot e^{-i\mathcal{H}_2 \Delta t} \cdot e^{-i\mathcal{H}_1 \frac{\Delta t}{2}} \\ &= \begin{pmatrix} 0.999 - 0.05i & 0 & -0.001 - 0.014i & -0.013i \\ 0 & 0.997 + 0.05i & -0.053i & 0.001 - 0.014i \\ -0.001 - 0.014i & -0.053i & 0.997 - 0.05i & 0 \\ -0.013i & 0.001 - 0.014i & 0 & 0.999 + 0.05i \end{pmatrix} \end{aligned}$$

8.5 Assume particles A and B are electrons (fermions) in quantum state $|\Psi_{AB}\rangle = \frac{1}{\sqrt{2}}(|\psi_A\rangle|\psi_B\rangle - |\psi_B\rangle|\psi_A\rangle)$. How is the state $\text{SWAP}|\Psi_{AB}\rangle$ related to the state $|\Psi_{AB}'\rangle$? What happens if $|\psi_A\rangle = |\psi_B\rangle$? Is the result a valid quantum state? What is to prevent this possibility?

8.6 Suppose a “programmable quantum simulator”, S , exists that can be programmed to simulate a unitary operator U applied to input quantum “data” $|d\rangle$ by effecting the transformation:

$$S|d\rangle|P_U\rangle = (U|d\rangle)|P'_U\rangle \quad (8.36)$$

Can such a device be built that is capable of deterministically simulating an arbitrary unitary operator of the appropriate dimensionality? To answer this, show that if N unitary operators, U_1, U_2, \dots, U_N , which are distinct up to a global phase, can be simulated by this device that:

(a) The program register is at least N -dimensional, i.e., requires $\log_2 N$ qubits and

- (b) The corresponding program states, $|P_{U_1}\rangle, |P_{U_2}\rangle, \dots, |P_{U_N}\rangle$ must be orthogonal.
- (c) What do these requirements on the dimensionality of the program register, and orthogonality of the program states, imply regarding the existence of a deterministic universal programmable quantum simulator?

(Hint: Start off by considering how S acts when asked to simulate two different programs acting on the same data, i.e.,

$$S|d\rangle|P_{U_1}\rangle = (U_1|d\rangle)|P'_{U_1}\rangle$$

$$S|d\rangle|P_{U_2}\rangle = (U_2|d\rangle)|P'_{U_2}\rangle$$

Then compute the overlap $\langle P_{U_2}|P_{U_1}\rangle$. What assumption would allow you to divide this overlap by $\langle P'_{U_2}|P'_{U_1}\rangle$? Do both sides of your resulting equation depend on the data d ? If not, what does the lack of d dependence imply regarding the form $U_2^\dagger U_1$ must take? Is this implication compatible with your starting assumptions? If not, which other assumption has to be wrong? What does this prove about the required orthogonality properties of the program states that you will need in order to simulate unitary operators that are distinct up to a global phase? Compare this to the number of potential unitary operators you will need to be able to simulate if your device is to be truly universal. What does that tell you about the existence of a deterministic programmable universal simulator?)

Chapter 9

Quantum Chemistry with a Quantum Computer

“The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble.”

– Paul Dirac¹

9.1 Classical Computing Approach to Quantum Chemistry

Quantum chemistry seeks to explain the chemical behavior of molecules by solving the molecular Schrödinger equation. In essence, it is an attempt to reduce chemistry to physics by using first principles calculations to predict chemical properties. Once the molecular Schrödinger’s equation is solved and the wavefunction is known, it is possible to predict *all* the chemical properties of the molecule by calculating the expectation values of the appropriate observables. In this manner bond angles, affinity, reactivity, electronic structure and the energy eigenspectrum of the molecule can all be obtained.

Typically, quantum chemists take their starting point to be the molecules’ time-independent many-electron Schrödinger equation. This describes the spatial part of the wavefunction, and can shed light on matters such energy eigenvalues of the molecule. In Cartesian coordinates, the time-independent Schrödinger equation has the form²:

$$\begin{aligned} & \left(-\frac{1}{2} \sum_{i=1}^N \nabla_i^2 + \sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i=1}^N V(\mathbf{r}_i) \right) \psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N) \\ & = E \psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N) \end{aligned} \tag{9.1}$$

¹Source: [145].

²This model uses the simplifying assumption that the positively charged nuclei are at fixed positions, and the electrons therefore move in a potential that is partly governed by the spatial distribution of these positively charged nuclei and partly by an external field.

where $\nabla_i^2 \equiv \frac{\partial^2}{\partial x_i^2} + \frac{\partial^2}{\partial y_i^2} + \frac{\partial^2}{\partial z_i^2}$ is the Laplacian operator, the \mathbf{r}_i signify the electron coordinates, $V(\mathbf{r}_i)$ the potential, and E the energy. This is an eigenvalue equation and can only be satisfied at discrete values the energy E . These are the only allowed energies the molecule can assume and define its energy eigenspectrum. Such eigen-spectra have many uses, e.g., they can allow us to recognize the chemical composition of a remote astronomical body by measuring the spectrum of the light it gives off and comparing the result to spectra predicted from first principles calculations.

As the electrons are fermions, the solution wavefunction of the many-electron Schrödinger equation is required to be anti-symmetric, i.e., it must acquire a minus sign whenever two electron coordinates are interchanged, i.e.,

$$\psi(\dots, \mathbf{r}_i, \dots, \mathbf{r}_j, \dots) = -\psi(\dots, \mathbf{r}_j, \dots, \mathbf{r}_i, \dots) \quad (9.2)$$

This anti-symmetric property can be guaranteed if we assume the solution wavefunction takes a certain form, namely, that it can be written as a sum of so-called “Slater determinants”:

$$\psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N) = \sum_I C_I \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_{i_1}(\mathbf{r}_1) & \dots & \phi_{i_N}(\mathbf{r}_1) \\ \dots & \dots & \dots \\ \phi_{i_1}(\mathbf{r}_N) & \dots & \phi_{i_N}(\mathbf{r}_N) \end{vmatrix} \quad (9.3)$$

where the index I runs over different configurations of orbitals. If there are N electrons and M one-electron orbitals there will be $\binom{M}{N} = M!/(N!(M-N)!)$ different Slater determinants in the expansion. If all these Slater determinants are included, the resulting model is called the “full configuration interaction” (FCI). However, the cost of performing calculations in the FCI is prohibitive as the number of determinants in the FCI grows factorially with the number of electrons and orbitals. Hence, it is only practical to use the FCI when dealing with molecules having only a few dozen electrons, and several million to a few billion determinants. When the FCI model is tractable, however, it yields solutions that are in good agreement with chemical properties that are measured experimentally.

To go beyond the few electron case, quantum chemists (limited to using classical computers) are obliged to impose various approximations and assumptions. These approximations and assumptions include asserting the geometric configuration of the molecule, ignoring relativistic effects, and choosing a particular basis set with which to expand the wavefunction of each atom. The choice of basis set is a critical factor in determining the fidelity of the model and, at this point in time, is something of an art.

Whereas the N electron/ M orbital full configuration interaction (FCI) model requires a runtime of $\mathcal{O}(M!)$ on a classical computer, which scales exponentially³ in M , an approximation called CCSD(T) (which stands for “Coupled-Cluster with Single and Double and Perturbative Triple excitations”) requires just $\mathcal{O}(M^7)$ steps,

³The exponential scaling can be seen from Stirling’s approximation of the factorial $M! \approx M^M e^{-M} \sqrt{2\pi M}$.

which is *polynomial* in M . However, even though CCSD(T) is considered the current “gold-standard” by quantum chemists, it is nevertheless still an *approximate* algorithm. Moreover, a seventh order polynomial scaling is not exactly cheap.

Using quantum computers, there is, however, an alternative approach. A quantum computer can run the full configuration interaction model (i.e., without approximation) in polynomial time, and we can extract certain information, such as ground state energy eigenvalues, in polynomial time too. The method uses a special quantum eigenvalue estimation algorithm that can only be run on a quantum computer. However, fairly sophisticated instances of this quantum algorithm have been simulated by Alán Aspuru-Guzik et al. [25], and an elementary quantum chemistry calculation has been performed on a rudimentary quantum computer [304]. These authors, which include leading edge quantum chemists, concluded that the quantum eigenvalue estimation algorithm can give an exponential speedup over the FCI model for the same level of accuracy, or alternatively, can give a polynomial speedup, but *greater* accuracy, over the highly respected CCSD(T) algorithm. Either way, the quantum algorithm shows great potential to make a positive impact on quantum chemistry. So let us take a look at this algorithm in some detail.

9.1.1 Classical Eigenvalue Estimation via the Lanczos Algorithm

One of the most important tasks for a quantum chemist is to calculate energy eigenvalues and energy eigenstates of a molecule. Usually, the ground state energy and the energies of the first few excited states are of greatest interest. In principle, given a Hamiltonian \mathcal{H} , which for realistic molecules is a *sparse* hermitian matrix, it ought to be possible to obtain its eigenvalues using elementary methods of linear algebra, such as finding the roots of the characteristic polynomial of the Hamiltonian matrix, i.e., finding the values of the eigenvalues λ for which $\det(\lambda\mathbb{1} - \mathcal{H}) = 0$. In practice, however, the dimensions of typical molecular Hamiltonian matrices are so large that naive approaches to eigenvalue determination are impractical. To combat this problem, more sophisticated methods must be used such as the Lanczos algorithm. For an explanation of this algorithm see pp. 470–507 of Ref. [204]. This works by mapping the original Hamiltonian matrix into a new basis in which it is tri-diagonal (without generating any full size intermediary matrices along the way) such that the eigenvalues of this tri-diagonal matrix are close to those of the original Hamiltonian. Moreover, there are specialized fast methods for obtaining the eigenvalues of tri-diagonal matrices. Even better, information about the Hamiltonian’s *extremal* eigenvalues (i.e., its largest and smallest ones) usually emerge long before the tri-diagonalization process completes. Hence, if it is only the smallest (or largest) eigenvalues that are sought, the Lanczos method is often a good tool to find them. Nevertheless, the complexity of the Lanczos algorithm is at least linear in the dimension of the Hamiltonian matrix. This, in turn, is set by the number of basis vectors used to describe the wavefunction, and this number can grow exponentially with the number of electrons in the molecule. This makes the determination of

molecular energy eigenvalues far more difficult than it might at first appear, as one is obliged to work with such large matrices.

9.2 Quantum Eigenvalue Estimation via Phase Estimation

Fortunately, quantum computing provides an alternative approach. Suppose we have a Hamiltonian \mathcal{H} with energy eigenstates $|\psi_\lambda\rangle$ and corresponding energy eigenvalues λ . That is, the eigenstates and eigenvalues satisfy $\mathcal{H}|\psi_\lambda\rangle = \lambda|\psi_\lambda\rangle$. The basic eigenvalue estimation problem is the following:

Eigenvalue Estimation Given a n -qubit Hamiltonian \mathcal{H} find the eigenvalue, λ , of \mathcal{H} corresponding to an eigenstate $|\psi_\lambda\rangle$, i.e., find λ such that $\mathcal{H}|\psi_\lambda\rangle = \lambda|\psi_\lambda\rangle$.

To accomplish this goal using a quantum computer we would like to switch attention to corresponding *unitary* operators for which we can build quantum circuits. If we had a physical system with Hamiltonian \mathcal{H} available to us, in time t this system would evolve according to the unitary operator $U = e^{-i\mathcal{H}t/\hbar}$. We can set the time $t = 1$ and work in units of $\hbar = 1$. Then, if λ is an eigenvalue of \mathcal{H} , $e^{-i\lambda}$ will be the corresponding eigenvalue of $U = e^{-i\mathcal{H}}$. Equally, we can associate the eigenvalues of the inverse of this unitary operator, $U = e^{+i\mathcal{H}}$ with the eigenvalues of \mathcal{H} too. Specifically, if λ is an eigenvalue of \mathcal{H} , then $e^{i\lambda}$ will be the corresponding eigenvalue of $U = e^{i\mathcal{H}}$, and $|\psi_\lambda\rangle$ will be an eigenstate of both \mathcal{H} and U . Thus, we can find an eigenvalue, λ , of the Hamiltonian \mathcal{H} by finding the corresponding eigenvalue, $e^{i\lambda}$, of $U = e^{i\mathcal{H}}$. This leads to the following slight reformulation of our problem as follows:

Quantum Eigenvalue Estimation Given a n -qubit Hamiltonian \mathcal{H} having eigenvalues and eigenstates such that $\mathcal{H}|\psi_\lambda\rangle = \lambda|\psi_\lambda\rangle$, define a related unitary operator $U = e^{i\mathcal{H}}$ having eigenvalues and eigenstates such that $U|\psi_\lambda\rangle = e^{i\lambda}|\psi_\lambda\rangle$. Find the eigenvalue, $e^{i\lambda}$, of U corresponding to an eigenstate $|\psi_\lambda\rangle$.

Having related the eigenvalues of \mathcal{H} to those of U the basic strategy of quantum eigenvalue estimation is to use U (or more precisely controlled- U^{2^k} gates) to create, efficiently, a quantum state, which we call the “phase state”, in which the desired eigenvalue appears as a phase factor, and then to extract this phase factor, efficiently, using a technique known as quantum phase estimation. Thus, quantum eigenvalue estimation reduces to a two-step process of creating the special phase state and then performing phase estimation on it.

9.2.1 The “Phase” State

We start by giving the structure of the required phase state and explain how to extract the phase from it using quantum phase estimation. Having done so it will be clear

why we would want to synthesize such a state. The desired “phase state” is:

$$\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{i\lambda y} |y\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \varphi y} |y\rangle \quad (9.4)$$

which contains the sought after eigenvalue, $\lambda = 2\pi\varphi$. Furthermore, it is sufficient to choose $0 \leq \varphi < 1$, which means we can express φ as a binary fraction $\varphi = 0.x_1x_2x_3\dots$. Thus, our desired phase state takes the form:

$$\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \varphi y} |y\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i (0.x_1x_2x_3\dots)y} |y\rangle \quad (9.5)$$

where each x_i is a binary digit, i.e., a 0 or a 1.

9.2.2 Binary Fraction Representation of the Phase Factor

The justification for the binary fraction expansion is as follows. We can write any real number $0 \leq \varphi < 1$ as the (potentially never-ending) binary fraction $\varphi = 0.x_1x_2x_3x_4\dots$, such that each x_i is a binary digit (i.e., a 0 or 1), and $\varphi = \sum_i x_i 2^{-i}$. Clearly, the maximum value we can obtain for φ is 1 (because $\sum_{i=1}^{\infty} 2^{-i} = 1$), and the minimum value is 0. Hence, in principle, any real number $0 \leq \varphi < 1$ can be so represented.

Binary fractions have interesting properties. By repeatedly multiplying by two’s we can split the number into a binary integer part (to the left of the dot) and a binary fraction part (to the right of the dot):

$$\begin{aligned} \varphi &= 0.x_1x_2x_3x_4\dots \\ 2^1\varphi &= x_1.x_2x_3x_4\dots \\ 2^2\varphi &= x_1x_2.x_3x_4\dots \\ &\vdots \\ 2^{(j-1)}\varphi &= x_1x_2x_3\dots x_{j-1}.x_jx_{j+1}\dots \end{aligned} \quad (9.6)$$

Consequently,

$$\begin{aligned} e^{2\pi i \varphi} &= e^{2\pi i 0.x_1x_2x_3x_4\dots} \\ e^{2\pi i 2^1\varphi} &= e^{2\pi i x_1.x_2x_3x_4\dots} = e^{2\pi i x_1} e^{2\pi i 0.x_2x_3x_4\dots} \\ e^{2\pi i 2^2\varphi} &= e^{2\pi i x_1x_2.x_3x_4\dots} = e^{2\pi i x_1x_2} e^{2\pi i 0.x_3x_4\dots} \\ &\vdots \end{aligned} \quad (9.7)$$

$$e^{2\pi i 2^{(j-1)}\varphi} = e^{2\pi i x_1 x_2 x_3 \dots x_{j-1} x_j x_{j+1} \dots} = e^{2\pi i x_1 x_2 x_3 \dots x_{j-1}} \cdot e^{2\pi i 0 \cdot x_j x_{j+1} \dots}$$

We can then eliminate the exponential factors, such as $e^{2\pi i x_1 x_2}$, because their arguments are always integer multiples of $2\pi i$ and so $e^{2\pi i x_1 x_2}$, and similar factors, are always 1. Thus, we have:

$$\begin{aligned} e^{2\pi i \varphi} &= e^{2\pi i 0 \cdot x_1 x_2 x_3 x_4 \dots} \\ e^{2\pi i 2^1 \varphi} &= e^{2\pi i 0 \cdot x_2 x_3 x_4 \dots} \\ e^{2\pi i 2^2 \varphi} &= e^{2\pi i 0 \cdot x_3 x_4 \dots} \\ &\vdots \\ e^{2\pi i 2^{(j-1)} \varphi} &= e^{2\pi i 0 \cdot x_j x_{j+1} \dots} \end{aligned} \tag{9.8}$$

Thus, as you can see, we can move the j -th bit in the expansion to immediately after the dot by multiplying by 2^{j-1} .

9.3 Quantum Phase Estimation

Given such a “phase” state, $\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \varphi y} |y\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i (0.x_1 x_2 x_3 \dots) y} |y\rangle$, we can reveal the digits x_1, x_2, x_3, \dots in the binary fraction representation of φ if we can transform the state into the corresponding computational basis state $|x_1 x_2 x_3 \dots\rangle$, and then read it out qubit-by-qubit. This operation is called “quantum phase estimation”.

Quantum Phase Estimation Given a state of the form $\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \varphi y} |y\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i (0.x_1 x_2 x_3 \dots) y} |y\rangle$ determine its phase factor φ by mapping the bits in its binary fraction expansion into the state $|x_1 x_2 x_3 \dots\rangle$.

Setting $\varphi \approx \frac{x}{2^n}$ for sufficiently large integer x , we see immediately that the phase state is nothing more than the quantum Fourier transform of the computational basis state $|x\rangle$. Specifically, we have:

$$\begin{aligned} |x\rangle &\xrightarrow{\text{QFT}} \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{xy}{2^n}} |y\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{y_1=0}^1 \sum_{y_2=0}^1 \dots \sum_{y_n=0}^1 e^{2\pi i x (\sum_{\ell=1}^n y_\ell 2^{-\ell})} |y_1 y_2 \dots y_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{y_1=0}^1 \sum_{y_2=0}^1 \dots \sum_{y_n=0}^1 \bigotimes_{\ell=1}^n e^{2\pi i x y_\ell 2^{-\ell}} |y_\ell\rangle \end{aligned}$$

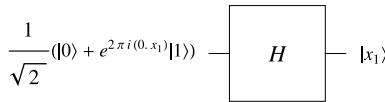


Fig. 9.1 Quantum circuit for transferring a phase factor, given as a 1-bit binary fraction in an exponent, into an output eigenstate. One can identify the bit value by reading this output eigenstate in the computational basis

$$\begin{aligned}
 &= \frac{1}{\sqrt{2^n}} \bigotimes_{\ell=1}^n \left(\sum_{y_\ell=0}^1 e^{2\pi i x y_\ell 2^{-\ell}} |y_\ell\rangle \right) \\
 &= \frac{1}{\sqrt{2^n}} \bigotimes_{\ell=1}^n (|0\rangle + e^{2\pi i x 2^{-\ell}} |1\rangle) \\
 &= \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2\pi i 0.x_n} |1\rangle) (|0\rangle + e^{2\pi i 0.x_{n-1}x_n} |1\rangle) \cdots (|0\rangle + e^{2\pi i 0.x_1x_2\dots x_n} |1\rangle)
 \end{aligned} \tag{9.9}$$

Since the phase state, $\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i (\frac{x}{2^n})y} |y\rangle$, is nothing more than the QFT of a computational basis state $|x\rangle$, this means we can extract the desired x (and so the desired phase factor $\varphi = \frac{x}{2^n}$) by applying the *inverse* quantum Fourier transform, QFT^\dagger , to the phase state (assuming we had such a state available)! We can see this most clearly by considering successive binary fraction approximations to φ .

Case 1: If $\varphi = 0.x_1$ Exactly

To begin imagine that $\varphi = 0.x_1$ exactly. In this case a single Walsh-Hadamard gate is sufficient to extract x_1 (see Fig. 9.1).

Case 2: If $\varphi = 0.x_1x_2$ Exactly

Next suppose instead that $\varphi = 0.x_1x_2$ exactly. In this case the circuit shown in Fig. 9.2 would be sufficient to extract x_1 and x_2 . This circuit uses two Hadamard gates and one controlled- R_2^{-1} gate where $R_2^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^2} \end{pmatrix}$.

Case 3: If $\varphi = 0.x_1x_2x_3$ Exactly

Now imagine $\varphi = 0.x_1x_2x_3$ exactly. In this case the circuit shown in Fig. 9.3 would be sufficient to extract x_1 , x_2 , and x_3 . This uses three Hadamard gates, two controlled- R_2^{-1} gates and one controlled- R_3^{-1} gate where $R_3^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^3} \end{pmatrix}$.

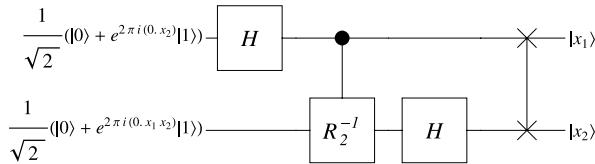


Fig. 9.2 Quantum circuit for transferring a phase factor, $\varphi = 0.x_1x_2$, i.e., a 2-bit binary fraction in an exponent, into the same 2-bit binary sequence in an eigenstate. One can identify the bit values by reading the eigenstate in the computational basis. Note that the last operation performed reverses the order of the qubits. In this circuit $R_2^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^2} \end{pmatrix}$. We can recover the value of φ from $\varphi \approx 2^{-1}x_1 + 2^{-2}x_2$

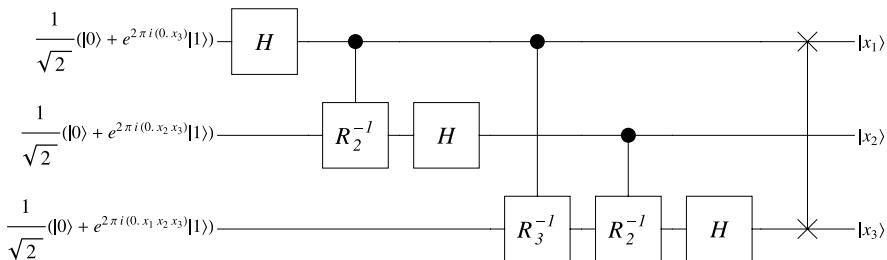


Fig. 9.3 Quantum circuit for transferring the 3-bit sequence of a binary fraction in an exponent into the same 3-bit sequence in an eigenstate. One can identify the bit values by reading the eigenstate in the computational basis. Note that the last operation performed reverses the order of the qubits. In this circuit $R_2^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^2} \end{pmatrix}$ and $R_3^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^3} \end{pmatrix}$. We can recover the value of φ from $\varphi \approx 2^{-1}x_1 + 2^{-2}x_2 + 2^{-3}x_3$

General Case: If $\varphi = 0.x_1x_2\dots x_n$ Exactly

In the general case, the overall transformation this circuit achieves is:

$$\sum_{y=0}^{2^n-1} e^{2\pi i \frac{xy}{2^n}} |y\rangle \rightarrow |x\rangle \quad (9.10)$$

where $\varphi = x/2^n$ and x is the integer which in binary is $(x)_{10} \equiv (x_1x_2\dots x_n)_2$. The reader will recognize this operation as the *inverse* Quantum Fourier Transform.

If $\varphi \neq 0.x_1x_2\dots x_n$ Exactly

In general, the sought after phase φ will not be exactly equal to a rational number $\varphi = \frac{x}{2^n}$ for any integers x and n . In this case, performing the aforementioned procedure will only yield an approximate answer.

To summarize, phase estimation is the process of extracting an unknown phase factor, $0 \leq \varphi < 1$, from a state of the form

$$|\Phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{i\lambda y} |y\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \varphi y} |y\rangle \quad (9.11)$$

To do so, we approximate φ to n bits of precision as the binary fraction $\varphi \approx (0.x_1x_2\dots x_n)_2 = (2^{-1}x_1 + 2^{-2}x_2 + \dots + 2^{-n}x_n)_{10}$ thereby re-casting the phase state as

$$|\Phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i (0.x_1x_2\dots x_n)y} |y\rangle \quad (9.12)$$

We then perform an operation that maneuvers the bit sequence defining the binary fraction representation of φ into one of the computational basis eigenstates, $|x_1x_2\dots x_n\rangle$. By reading this state in the computational basis we extract the bits x_1, x_2, \dots, x_n that determine the mystery phase factor. Moreover, we recognize that the required maneuver is a QFT run in the reverse direction, i.e., it is an *inverse* Quantum Fourier Transform.

9.4 Eigenvalue Kick-Back for Synthesizing the Phase State

Let us now return to eigenvalue estimation. To recap, we assume we know a Hamiltonian \mathcal{H} and wish to find the eigenvalue λ corresponding to eigenstate $|\psi_\lambda\rangle$. That is, given $|\psi_\lambda\rangle$ we seek λ such that

$$\mathcal{H}|\psi_\lambda\rangle = \lambda|\psi_\lambda\rangle \quad (9.13)$$

The eigenstate $|\psi_\lambda\rangle$ or an approximation to it may be known explicitly, or it may be unknown, but physically available to us, as the output from some preceding quantum simulation. Either way, we want to use knowledge of \mathcal{H} and the physical eigenstate $|\psi_\lambda\rangle$ to find λ in a manner that is more efficient than can be done classically.

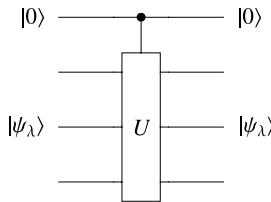
Rather than working with \mathcal{H} directly, our strategy is to work with the unitary operator $U = e^{i\mathcal{H}}$. This is because, if \mathcal{H} has eigenstate $|\psi_\lambda\rangle$ with eigenvalue λ , $U = e^{i\mathcal{H}}$ has eigenstate $|\psi_\lambda\rangle$ with eigenvalue $e^{i\lambda}$. That is, with $U = e^{i\mathcal{H}}$ we have:

$$U|\psi_\lambda\rangle = e^{i\lambda}|\psi_\lambda\rangle \quad (9.14)$$

Furthermore, we set $\lambda = 2\pi\varphi$ and express φ to n bits of precision as the binary fraction $\varphi = 0.x_1x_2\dots x_n$. So it is also true that

$$U|\psi_\lambda\rangle = e^{i\lambda}|\psi_\lambda\rangle = e^{2\pi i \varphi}|\psi_\lambda\rangle = e^{2\pi i 0.x_1x_2\dots x_n}|\psi_\lambda\rangle \quad (9.15)$$

Fig. 9.4 When the control qubit is set to $|0\rangle$ the output states match the input states and there is no eigenvalue kick-back



We showed in Sect. 9.3 that if we could create a state we call the “phase state” $|\Phi\rangle$ such that

$$\begin{aligned}
 |\Phi\rangle &= \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{i\lambda y} |y\rangle \\
 &= \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \varphi y} |y\rangle \\
 &= \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i (0.x_1x_2\dots x_n)y} |y\rangle \\
 &= \frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{2\pi i 0.x_n} |1\rangle \right) \left(|0\rangle + e^{2\pi i 0.x_{n-1}x_n} |1\rangle \right) \dots \left(|0\rangle + e^{2\pi i 0.x_1x_2\dots x_n} |1\rangle \right)
 \end{aligned} \tag{9.16}$$

we can extract the binary digits x_1, x_2, \dots, x_n by applying the inverse QFT to the phase state, i.e., $\text{QFT}^{-1}|\Phi\rangle = |x_1x_2\dots x_n\rangle$. Thus, to find the eigenvalue $\lambda = 2\pi\varphi$ we pursue a two-step strategy: first construct the phase state $|\Phi\rangle$ and then apply the inverse QFT to extract the digits x_1, x_2, \dots, x_n of the binary fraction for $\varphi = (0.x_1x_2\dots x_n)_2 = (2^{-1}x_1 + 2^{-2}x_2 + \dots + 2^{-n}x_n)_{10}$. Given the phase φ , we can then compute the eigenvalue $\lambda = 2\pi\varphi$. As the method for extracting the phase from the phase state is already known, all that remains is to understand how to *synthesize* the phase state $|\Phi\rangle$ given knowledge of \mathcal{H} (or equivalently $U = e^{i\mathcal{H}}$) and the state $|\psi_\lambda\rangle$. We will next show how to accomplish this using the technique of “eigenvalue kick-back”.

To see how eigenvalue kick-back works consider the circuit shown in Fig. 9.4. Conceptually, the qubits in this circuit are split into two registers; the first containing a single “control” qubit and the second containing n “target” qubits, and the circuit is comprised of a single controlled- U gate, which we assume can be implemented efficiently. Consider the effect of a “controlled- U ” gate on inputs $|0\rangle|\psi_\lambda\rangle$ and $|1\rangle|\psi_\lambda\rangle$ individually. When the control value is $|0\rangle$ the U gate does not act on its qubits, and the circuit passes the input states unchanged.

However, as shown in Fig. 9.5, if, instead, the control value is set to $|1\rangle$, making the input $|1\rangle|\psi_\lambda\rangle$, then U will act on its qubits and compute $U|\psi_\lambda\rangle = e^{i\lambda}|\psi_\lambda\rangle$ because $|\psi_\lambda\rangle$ is an eigenstate of both \mathcal{H} and $U = e^{i\mathcal{H}}$. However, from a mathematical point of view the states $|1\rangle \otimes (e^{i\lambda}|\psi_\lambda\rangle)$ and $(e^{i\lambda}|1\rangle) \otimes |\psi_\lambda\rangle$ are the same, so

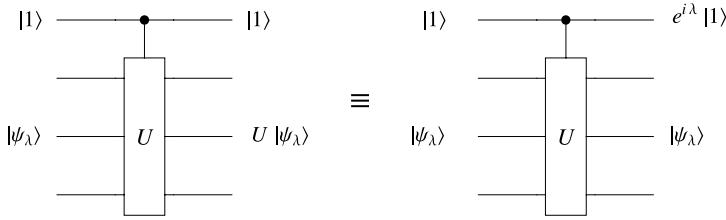


Fig. 9.5 When the control qubit is set to $|1\rangle$ the output state is $|1\rangle U|\psi_\lambda\rangle = |1\rangle e^{i\lambda}|\psi_\lambda\rangle$, since $|\psi_\lambda\rangle$ is an eigenstate of U with eigenvalue $e^{i\lambda}$. This state is equivalent, mathematically, to the state $e^{i\lambda}|1\rangle|\psi_\lambda\rangle$. In other words, this circuit, when the control qubit is in state $|1\rangle$, has kicked-back the eigenvalue from the second register (the qubits on which U acts) to the phase of the first register (the control qubit)

we are free to interpret the output either way. If we adopt the latter interpretation it appears that the action of the controlled- U gate has “kicked-back” the eigenvalue of U (namely $e^{i\lambda}$) as the phase factor of the control qubit. Hence, the name “eigenvalue kick-back”. Thus, the effect of the control- U gate with the control qubit in a computational basis state is given by:

$$|0\rangle|\psi_\lambda\rangle \xrightarrow{\text{controlled-}U} |0\rangle|\psi_\lambda\rangle \quad (9.17)$$

$$|1\rangle|\psi_\lambda\rangle \xrightarrow{\text{controlled-}U} |1\rangle(e^{i\lambda}|\psi_\lambda\rangle) \equiv (e^{i\lambda}|1\rangle)|\psi_\lambda\rangle \quad (9.18)$$

By the relationships defined previously we also note that

$$e^{i\lambda}|1\rangle|\psi_\lambda\rangle = e^{i2\pi\varphi}|1\rangle|\psi_\lambda\rangle = e^{i2\pi 0.x_1x_2\dots x_j}|1\rangle|\psi_\lambda\rangle \quad (9.19)$$

Unfortunately, as we saw in Chap. 1, a *global* phase shift of a quantum state $|1\rangle|\psi_\lambda\rangle$, to make it a state such as $e^{i\lambda}|1\rangle|\psi_\lambda\rangle$, has no measurable consequences. However, if the state of the control qubit is made to be a *superposition* of its two control values, as in $a|0\rangle + b|1\rangle$ say, then the phase factor we just saw becomes a *relative* phase between the $|0\rangle$ and $|1\rangle$ components in the superposition $a|0\rangle + b|1\rangle$. Specifically, we obtain the transformation

$$(a|0\rangle + b|1\rangle)|\psi_\lambda\rangle \xrightarrow{\text{controlled-}U} (a|0\rangle + be^{i\lambda}|1\rangle)|\psi_\lambda\rangle \quad (9.20)$$

which *does* have measurable consequences.

Next consider the effect of a controlled- U^k gate on the states $|0\rangle|\psi_\lambda\rangle$ and $|1\rangle|\psi_\lambda\rangle$. If $|\psi_\lambda\rangle$ is an eigenstate of U with eigenvalue $e^{i\lambda}$, it is also an eigenstate of U^k (an integer) with eigenvalue $e^{ik\lambda}$. Thus, if the control qubit is in the superposition $a|0\rangle + b|1\rangle$ we have:

$$(a|0\rangle + b|1\rangle)|\psi_\lambda\rangle \xrightarrow{\text{controlled-}U^k} (a|0\rangle + be^{ik\lambda}|1\rangle)|\psi_\lambda\rangle \quad (9.21)$$

Hence, using a controlled- U^k gate in lieu of a controlled- U gate multiplies the phase factor appearing in the control qubit by k . We can exploit this trick to systematically

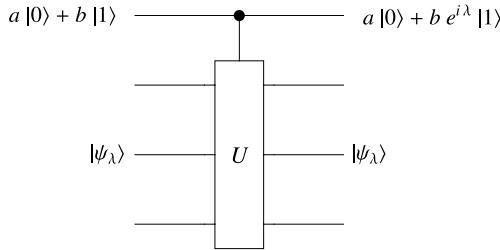


Fig. 9.6 When the control qubit is set to a superposition of its two control values, the circuit causes a relative phase to be introduced between the $|0\rangle$ and $|1\rangle$ eigenstates. This is significant because the state we are trying to synthesize can be written as the direct product of several single qubit states in which there is a prescribed relative phase difference between the $|0\rangle$ and $|1\rangle$ components

construct the single qubit states within $|\Phi\rangle$ as given in the last line of (9.16), which you will note is *unentangled*. That is, we want to synthesize the phase state

$$|\Phi\rangle = \frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{2\pi i 0.x_n} |1\rangle \right) \left(|0\rangle + e^{2\pi i 0.x_{n-1}x_n} |1\rangle \right) \cdots \left(|0\rangle + e^{2\pi i 0.x_1x_2\dots x_n} |1\rangle \right) \quad (9.22)$$

by synthesizing its terms individually. For this we use the transformation $(a|0\rangle + b|1\rangle)|\psi_\lambda\rangle \xrightarrow{\text{controlled-}U^k} (a|0\rangle + b e^{ik\lambda}|1\rangle)|\psi_\lambda\rangle$ with k equal to increasing powers of 2. To see why this works recall from Sect. 9.2.2 that multiplying a binary fraction $0.x_1x_2x_3\dots x_n$ by 2^{j-1} moves the j -th bit to the position immediately after the dot, i.e., $2^{j-1}0.x_1x_2x_3\dots x_n = x_1x_2\dots x_{j-1}.x_jx_{j+1}\dots x_n$. Splitting the resulting number into the parts to the left and the right of the dot and simplifying gives us $e^{2\pi i 2^{j-1}0.x_1x_2x_3\dots x_n} = e^{2\pi i 0.x_jx_{j+1}\dots x_n}$ etc. Therefore, setting $k = 2^{j-1}$ for $j = 1, 2, 3, \dots, n$ in the controlled- U^k gates gives us:

$$e^{2\pi i k\varphi} = e^{2\pi i 2^{j-1}\varphi} = e^{2\pi i 2^{j-1}0.x_1x_2\dots x_n} = e^{2\pi i 0.x_jx_{j+1}\dots x_n} \quad (9.23)$$

Hence for $a = b = \frac{1}{\sqrt{2}}$ and $k = 2^0, 2^1, 2^2, \dots, 2^{n-1}$ we can obtain the transformations

$$\begin{aligned} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\psi_\lambda\rangle &\xrightarrow{\text{controlled-}U^{2^0}} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.x_1x_2x_3\dots x_n} |1\rangle)|\psi_\lambda\rangle \\ \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\psi_\lambda\rangle &\xrightarrow{\text{controlled-}U^{2^1}} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.x_2x_3\dots x_n} |1\rangle)|\psi_\lambda\rangle \\ \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\psi_\lambda\rangle &\xrightarrow{\text{controlled-}U^{2^2}} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.x_3\dots x_n} |1\rangle)|\psi_\lambda\rangle \\ &\vdots \\ \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\psi_\lambda\rangle &\xrightarrow{\text{controlled-}U^{2^{n-1}}} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.x_n} |1\rangle)|\psi_\lambda\rangle \end{aligned} \quad (9.24)$$

The direct product of all these individual qubit states is the phase state $|\Phi\rangle$ that we seek to synthesize.

9.5 Quantum Eigenvalue Estimation Algorithms

“... quantum computers of tens to hundreds of qubits can match and exceed the capabilities of classical full configuration interaction (FCI) calculations”

– Alán Aspuru-Guzik [25]

The combination of the eigenvalue kick-back trick of Sect. 9.4 and the quantum phase estimation trick of Sect. 9.3 gives us everything we need to complete a full quantum eigenvalue estimation algorithm. That is, to perform quantum eigenvalue estimation we use eigenvalue kick-back to synthesize the phase state, $|\Phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i (0.x_1x_2\dots x_n)y} |y\rangle$, and then quantum phase estimation to extract the phase, i.e., $\text{QFT}^{-1}|\Phi\rangle = |x_1x_2\dots x_n\rangle$. Dan Abrams and Seth Lloyd were the first to combine the eigenvalue kick-back trick with the inverse quantum Fourier transform to perform eigenvalue determination [2]. Personally, I believe this is by far the most important quantum algorithm discovered to date because it is the core component of so many of the quantum algorithms that display exponential speedups.

9.5.1 Abrams-Lloyd Eigenvalue Estimation Algorithm

The Abrams-Lloyd algorithm requires two registers: the first containing n -qubits will be used obtain a n -bit approximation to the desired eigenvalue, $e^{i\lambda} = e^{2\pi i \varphi}$ by determining the bits in the binary fraction expansion of $\varphi = (0.x_1x_2x_3\dots x_n)_2$, and the second containing m -qubits sufficient to hold $|\psi_\lambda\rangle$ on which controlled- U^{2^k} operations are performed for $k = 0, 1, 2, \dots, (n - 1)$. The overall circuit is shown in Fig. 9.7.

9.5.2 Kitaev Eigenvalue Estimation Algorithm

An alternative to the Abrams-Lloyd eigenvalue estimation algorithm was proposed by Kitaev [282]. Kitaev’s algorithm obtains the bits in the binary fraction expansion of the phase factor φ one at a time rather than all-at-once as in the Abrams-Lloyd algorithm. However, it has a lower overhead in terms the number of qubits needed because it only requires a single extra qubit beyond those needed to perform controlled- U^{2^k} operations. A quantum circuit for Kitaev’s algorithm is shown in Fig. 9.8.

Given a unitary operator, U , one of whose eigenstates is $|\psi_\lambda\rangle$, find the corresponding eigenvalue $e^{i\lambda}$.

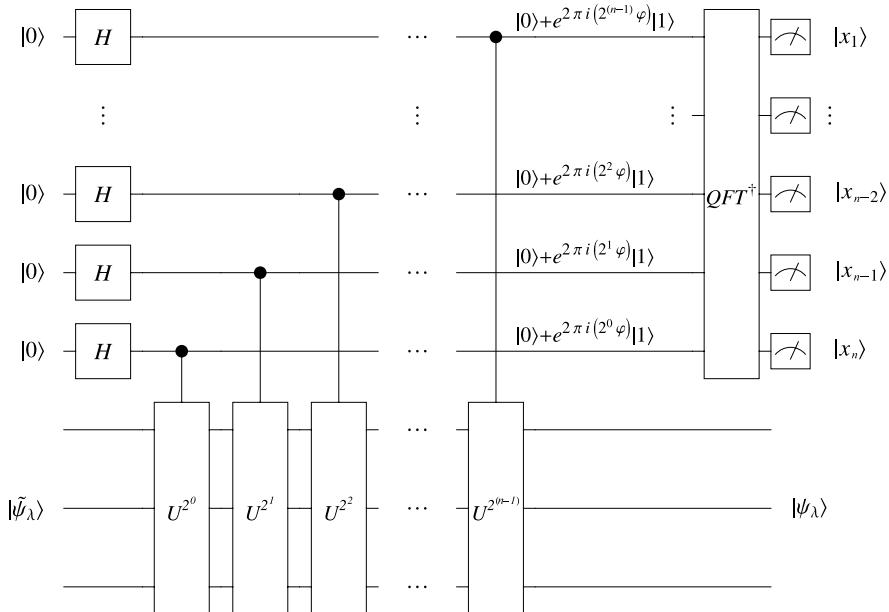
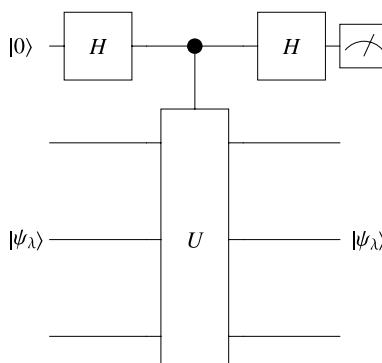


Fig. 9.7 Quantum circuit for the Abrams-Lloyd Eigenvalue Estimation Algorithm. The input $|\tilde{\psi}_\lambda\rangle$ is a close guess at the true eigenstate $|\psi_\lambda\rangle$ with eigenvalue $e^{i\lambda} = e^{2\pi i \varphi}$ where $0 \leq \varphi < 1$. Representing φ as a binary fraction, $\varphi = 0.x_1x_2\dots x_n$, measurement of the top n qubits in the computational basis reveals the bit values x_1, x_2, \dots, x_n and hence the eigenvalue $\lambda = 2\pi\varphi = 2\pi(0.x_1x_2\dots x_n)_2 = 2\pi(2^{-1}x_1 + 2^{-2}x_2 + \dots + 2^{-n}x_n)_{10}$

Fig. 9.8 Quantum circuit for the Kitaev Eigenvalue Estimation Algorithm



In Kitaev's algorithm, the key idea is to create, repeatedly, a superposition state of the form $|\psi(\varphi)\rangle = \sqrt{p_0(\varphi)}|0\rangle + \sqrt{p_1(\varphi)}|1\rangle$ in which the amplitudes are a known function of the phase factor φ . By preparing sufficiently many instances of the state $|\psi(\varphi)\rangle$ and measuring each one independently in the computational basis, one obtains an estimate of $p_0(\varphi)$, which you then invert to obtain φ . A key feature of the algorithm is that each round of preparation and measurement, the starting state

$|\psi(\varphi)\rangle$ is restored up to an unimportant global phase factor. This means that the repeated preparations of $|\psi(\varphi)\rangle$ are obtained automatically.

Kitaev's Eigenvalue Estimation Algorithm Given an eigenstate, $|\psi_\lambda\rangle$, of a unitary operator U , find the corresponding eigenvalue $e^{i\lambda} = e^{2\pi i\varphi}$

$$\begin{aligned}
 |0\rangle|\psi_\lambda\rangle &\xrightarrow{H \otimes \mathbb{1}} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\psi_\lambda\rangle \\
 &\xrightarrow{\text{controlled-}U} \frac{1}{\sqrt{2}}|0\rangle|\psi_\lambda\rangle + \frac{1}{\sqrt{2}}|1\rangle U|\psi_\lambda\rangle \\
 &= \frac{1}{\sqrt{2}}|0\rangle|\psi_\lambda\rangle + \frac{e^{i\lambda}}{\sqrt{2}}|1\rangle|\psi_\lambda\rangle \\
 &\xrightarrow{H \otimes \mathbb{1}} \frac{1}{\sqrt{2}}\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)|\psi_\lambda\rangle + \frac{e^{i\lambda}}{\sqrt{2}}\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)|\psi_\lambda\rangle \\
 &= \frac{1}{2}(1 + e^{i\lambda})|0\rangle|\psi_\lambda\rangle + \frac{1}{2}(1 - e^{i\lambda})|1\rangle|\psi_\lambda\rangle
 \end{aligned} \tag{9.25}$$

To follow the aforementioned steps you need to use the fact that $|\psi_\lambda\rangle$ is an eigenstate of U with eigenvalue $e^{i\lambda}$, i.e., $U|\psi_\lambda\rangle = e^{i\lambda}|\psi_\lambda\rangle$. If the first qubit of the final state is measured, it is found to be $|0\rangle$ with probability $p_0(\varphi) = \frac{1}{4}|1 + e^{i\lambda}|^2 = \frac{1}{4}|1 + e^{2i\pi\varphi}|^2 = \cos^2(\pi\varphi)$ and $|1\rangle$ with probability $p_1(\varphi) = \frac{1}{4}|1 - e^{i\lambda}|^2 = \frac{1}{4}|1 - e^{2i\pi\varphi}|^2 = \sin^2(\pi\varphi)$. Depending on the value obtained the output is projected into either $|0\rangle|\psi_\lambda\rangle$ or $|1\rangle|\psi_\lambda\rangle$. Notice that we can re-use the output (either directly or by negating the first qubit) as the input to the next round, to repeat the process an arbitrary number of times. By collecting statistics on how often we find the first qubit to be $|0\rangle$ and how often we find it to be $|1\rangle$ we can estimate $p_0(\varphi) = \cos^2(\pi\varphi)$ and $p_1(\varphi) = \sin^2(\pi\varphi)$. Hence we can estimate φ and therefore the eigenvalue $e^{2\pi i\varphi}$. Amazing!

However, we need to ask ourselves how good an estimate we obtain as a function of the number of times we repeat the process. Suppose we find the first qubit to be in state $|0\rangle$ N_0 times out of N trials. This allows us to estimate the probability p_0 to be $p_0^{\text{est}} = N_0/N$. Then by the weak Law of Large Numbers, for any δ , we have:

$$\text{Prob}(|p_0^{\text{est}} - p_0| > \delta) \leq \epsilon = \frac{2}{\sqrt{2\pi}} \exp\left(-\frac{\delta^2 N}{2p_0 p_1}\right) \tag{9.26}$$

There is good news and bad news in this equation. The good news is that for a fixed precision (i.e., fixed δ) the error probability decreases exponentially with increasing numbers of trials. However, conversely, for a fixed error probability (i.e., fixed ϵ) for each extra bit of precision we want to get in δ (i.e., each time we want to use a δ half the size it was last time), we can only do so and still maintain the desired fixed error probability by increasing the number of trials by a factor of *four*. That is, if we replace $\delta \rightarrow \delta/2$ and we want to keep ϵ fixed, we need to increase $N \rightarrow 4N$. Thus,

to get l bits of precision in the estimate of p_0 requires $O(4^l)$ trials. This is bad news. The Abrams-Lloyd algorithm is therefore my preferred version.

9.6 Quantum Chemistry Beyond Eigenvalue Estimation

There is much more to quantum chemistry than merely eigenvalue estimation of course. Quantum chemistry seeks to predict the fundamental properties of elements, molecules, and compounds from first principle quantum mechanical calculations. At the present time the vast majority of quantum chemistry calculations solve *approximate* quantum mechanical models on classical computers. Such approximations are necessary to make the computations tractable but they render the models less accurate and can lead to spurious predictions. A much better approach would be to use quantum mechanics to perform quantum mechanical calculations, such as we did for eigenvalue estimation.

In addition to eigenvalues, quantum chemists are also interested in predicting, on the basis of first principles calculations, the detailed physical and chemical properties of the chemical elements, the shape of complex molecules, complete descriptions of molecular spectra, heats of reaction, reactivity, etc. Remarkably, although not widely known, several quantum algorithms have been devised to perform such computations. For example, Wang et al. have developed a quantum algorithm for obtaining the spectrum of a molecule [522]; Jordan et al. use a quantum algorithm to simulate chemical dynamics [266]; Perdomo et al. use a quantum algorithm to predict lattice protein conformations [384]; and Rebentrost et al. have analyzed quantum transport phenomena, critical to light harvesting in photosynthesis, and have shown that such quantum transport can actually be enhanced by the presence of some noise in the quantum system [414].

9.7 Summary

In quantum chemistry, one is often interested in the *static* properties of a molecular quantum system, such its electronic structure, or its energy eigenvalues and eigenstates. In this chapter we describe the Abrams-Lloyd and Kitaev eigenvalue estimation algorithms. These provide efficient algorithms for determining the exact eigenvalue associated with a given eigenstate, a feat that is exponentially more difficult to do classically to the same precision.

In particular, using the quantum eigenvalue estimation algorithm, we see that quantum computers can determine the ground state energies of molecules exponentially faster than classical computers, and that chemically useful results can be obtained with just 20 qubits—far fewer qubits than are required for Shor’s algorithm. In fact, ignoring the qubits needed for error correction, quantum computers having 50–100 perfect qubits would exceed the capabilities of the World’s best supercomputers on such problems. This could potentially revolutionize quantum chemistry

and allow full configuration interaction calculations to be extended to much more complex molecules than can be analyzed today. Moreover, since the quantum eigenvalue estimation algorithm is not restricted to determining only ground state energy eigenvalues we can, by using different inputs, effectively probe multiple parts of the energy eigenspectrum of a molecule. However, we cannot compute the entire eigenspectrum any faster than is possible classically.

We ended by noting that quantum computing could potentially play a much greater role in quantum chemistry than just eigenvalue estimation, and we gave examples of quantum algorithms that have been devised for determining properties of molecular eigenspectra, molecular conformations, and even reaction rates. It is apparent that a great deal could be done to make special purpose quantum devices for assisting in quantum chemistry calculations.

9.8 Exercises

9.1 Rewrite the following real numbers as binary fractions to 16 bits of precision:

- (a) 0.84159_{10}
- (b) 0.29865_{10}
- (c) 0.11111_{10}

9.2 Rewrite the following binary fractions as real numbers to 5 decimal places:

- (a) 0.00110100_2
- (b) 0.11010101_2
- (c) 0.11111111_2

9.3 Any eigenvalue ξ of a matrix U can be expressed in “polar” form as $\xi = re^{i\theta}$.

- (a) Write down formulae for r and θ assuming ξ is known and possibly complex.
- (b) Prove that if U is unitary then it is always the case that $|r| = 1$.
- (c) Show, by way of providing an explicit formula connecting θ with φ , how to rewrite $\xi = re^{i\theta}$ as $\xi = e^{2\pi i \varphi}$ in which it is *guaranteed* that $0 \leq \varphi < 1$.
- (d) Use your formula to rewrite the eigenvalue $\xi = -0.749789 + 0.661677i$ in the form $\xi = e^{2\pi i \varphi}$ such that $0 \leq \varphi < 1$.
- (e) Express the value you found for φ as a binary fraction to 16 bits of precision.

9.4 Let \mathcal{H} be an hermitian matrix with eigenvector $|\psi_\lambda\rangle$ and eigenvalue λ , i.e., let $|\psi_\lambda\rangle$ and λ be defined such that $\mathcal{H}|\psi_\lambda\rangle = \lambda|\psi_\lambda\rangle$.

- (a) Prove that $U = e^{i\mathcal{H}}$ is a unitary matrix with eigenvector $|\psi_\lambda\rangle$ and eigenvalue $e^{i\lambda}$.
- (b) Prove further that, for integers $k = 0, 1, 2, \dots$, $|\psi_\lambda\rangle$ is also an eigenvector of U^k with eigenvalue $e^{ik\lambda}$.

9.5 Let \mathcal{H} be the Hamiltonian of a 2-qubit system such that $\mathcal{H} = \alpha X \otimes X + \beta Z \otimes Z$ where X and Z are Pauli matrices and α and β are real numbers.

(a) Prove that

$$\mathcal{H} = \begin{pmatrix} \beta & 0 & 0 & \alpha \\ 0 & -\beta & \alpha & 0 \\ 0 & \alpha & -\beta & 0 \\ \alpha & 0 & 0 & \beta \end{pmatrix}.$$

(b) Prove that

$$U = e^{i\mathcal{H}}$$

$$= \begin{pmatrix} \frac{1}{2}e^{i\beta-i\alpha} + \frac{1}{2}e^{i\alpha+i\beta} & 0 & 0 & -\frac{1}{2}e^{i\beta-i\alpha} + \frac{1}{2}e^{i\alpha+i\beta} \\ 0 & \frac{1}{2}e^{-i\alpha-i\beta} + \frac{1}{2}e^{i\alpha-i\beta} & -\frac{1}{2}e^{-i\alpha-i\beta} + \frac{1}{2}e^{i\alpha-i\beta} & 0 \\ 0 & -\frac{1}{2}e^{-i\alpha-i\beta} + \frac{1}{2}e^{i\alpha-i\beta} & \frac{1}{2}e^{-i\alpha-i\beta} + \frac{1}{2}e^{i\alpha-i\beta} & 0 \\ -\frac{1}{2}e^{i\beta-i\alpha} + \frac{1}{2}e^{i\alpha+i\beta} & 0 & 0 & \frac{1}{2}e^{i\beta-i\alpha} + \frac{1}{2}e^{i\alpha+i\beta} \end{pmatrix}. \quad (9.27)$$

(c) Prove that the normalized eigenvectors of U are

$$|\psi_{\lambda_0}\rangle = \frac{1}{\sqrt{2}}(|10\rangle - |01\rangle)$$

$$|\psi_{\lambda_1}\rangle = \frac{1}{\sqrt{2}}(|10\rangle + |01\rangle)$$

$$|\psi_{\lambda_2}\rangle = \frac{1}{\sqrt{2}}(|11\rangle - |00\rangle)$$

$$|\psi_{\lambda_3}\rangle = \frac{1}{\sqrt{2}}(|11\rangle + |00\rangle) \quad (9.28)$$

(d) Show that the corresponding eigenvalues of U are

$$\lambda_0 = e^{-i(\alpha+\beta)}$$

$$\lambda_1 = e^{i(\alpha-\beta)}$$

$$\lambda_2 = e^{-i(\alpha-\beta)}$$

$$\lambda_3 = e^{i(\alpha+\beta)} \quad (9.29)$$

(e) Show, with the aforementioned Hamiltonian $\mathcal{H} = \alpha X \otimes X + \beta Z \otimes Z$ that the operator for the Kitaev eigenvalue estimation circuit takes the form:

$$\begin{pmatrix} \frac{1}{2}(e^{i\beta} \cos(\alpha) + 1) & 0 & 0 & \frac{1}{2}ie^{i\beta} \sin(\alpha) & \frac{1}{2}(1 - e^{i\beta} \cos(\alpha)) & 0 & 0 & -\frac{1}{2}ie^{i\beta} \sin(\alpha) \\ 0 & \frac{1}{2}(e^{-i\beta} \cos(\alpha) + 1) & \frac{1}{2}ie^{-i\beta} \sin(\alpha) & 0 & 0 & \frac{1}{2}(1 - e^{-i\beta} \cos(\alpha)) & -\frac{1}{2}ie^{-i\beta} \sin(\alpha) & 0 \\ 0 & \frac{1}{2}ie^{-i\beta} \sin(\alpha) & \frac{1}{2}(e^{-i\beta} \cos(\alpha) + 1) & 0 & 0 & -\frac{1}{2}ie^{-i\beta} \sin(\alpha) & \frac{1}{2}(1 - e^{-i\beta} \cos(\alpha)) & 0 \\ \frac{1}{2}ie^{i\beta} \sin(\alpha) & 0 & 0 & \frac{1}{2}(e^{i\beta} \cos(\alpha) + 1) & -\frac{1}{2}ie^{i\beta} \sin(\alpha) & 0 & 0 & \frac{1}{2}(1 - e^{i\beta} \cos(\alpha)) \\ \frac{1}{2}(1 - e^{i\beta} \cos(\alpha)) & 0 & 0 & -\frac{1}{2}ie^{i\beta} \sin(\alpha) & \frac{1}{2}(e^{i\beta} \cos(\alpha) + 1) & 0 & 0 & -\frac{1}{2}ie^{i\beta} \sin(\alpha) \\ 0 & \frac{1}{2}(1 - e^{-i\beta} \cos(\alpha)) & -\frac{1}{2}ie^{-i\beta} \sin(\alpha) & 0 & 0 & \frac{1}{2}(e^{-i\beta} \cos(\alpha) + 1) & \frac{1}{2}ie^{-i\beta} \sin(\alpha) & 0 \\ 0 & -\frac{1}{2}ie^{-i\beta} \sin(\alpha) & \frac{1}{2}(1 - e^{-i\beta} \cos(\alpha)) & 0 & 0 & \frac{1}{2}ie^{-i\beta} \sin(\alpha) & \frac{1}{2}(e^{-i\beta} \cos(\alpha) + 1) & 0 \\ -\frac{1}{2}ie^{i\beta} \sin(\alpha) & 0 & 0 & \frac{1}{2}(1 - e^{i\beta} \cos(\alpha)) & \frac{1}{2}ie^{i\beta} \sin(\alpha) & 0 & 0 & \frac{1}{2}(e^{i\beta} \cos(\alpha) + 1) \end{pmatrix} \quad (9.30)$$

- (f) Show that with input $|0\rangle|\psi_{\lambda_0}\rangle$ the Kitaev circuit produces the state

$$\begin{aligned} & \frac{1}{2}(1 + \lambda_0)|0\rangle|\psi_{\lambda_0}\rangle + \frac{1}{2}(1 - \lambda_0)|1\rangle|\psi_{\lambda_0}\rangle \\ &= \left(-\frac{1}{2\sqrt{2}} - \frac{e^{-i\alpha-i\beta}}{2\sqrt{2}}\right)|001\rangle + \left(\frac{1}{2\sqrt{2}} + \frac{e^{-i\alpha-i\beta}}{2\sqrt{2}}\right)|010\rangle \\ &+ \left(-\frac{1}{2\sqrt{2}} + \frac{e^{-i\alpha-i\beta}}{2\sqrt{2}}\right)|101\rangle + \left(\frac{1}{2\sqrt{2}} - \frac{e^{-i\alpha-i\beta}}{2\sqrt{2}}\right)|110\rangle \end{aligned} \quad (9.31)$$

- (g) With what probability is the first qubit in the output measured to be $|0\rangle$ and with what probability is it measured to be $|1\rangle$? How are these probabilities related to the eigenvalue $\lambda_0 = e^{-i(\alpha+\beta)}$?

9.6 Let \mathcal{H} be the Hamiltonian of a 2-qubit system such that $\mathcal{H} = \alpha X \otimes X + \beta Y \otimes Y$ where X and Y are Pauli matrices and α and β are real numbers.

- (a) What is the unitary matrix $U = e^{i\mathcal{H}}$?
- (b) What are the eigenvalues of U ?
- (c) What are the normalized eigenvectors of U expressed as ket vectors over the computational basis?
- (d) The Abrams-Lloyd eigenvalue estimation algorithm makes use of the controlled- U^{2^k} gates. Write down the unitary matrices for controlled- U^{2^0} , controlled- U^{2^1} and controlled- U^{2^2} .
- (e) Given the gate embedding shown in Fig. 9.7 do these controlled- U^{2^k} gates commute? If so, is there any advantage in ordering these gates so that the value of k increases from left to right, or decreases from left to right, in the circuit?
- (f) How is k related to the precision with which we can estimate an eigenvalue of \mathcal{H} ?
- (g) How many times must we run the Abrams-Lloyd algorithm to obtain an estimate for the energy eigenvalue corresponding to a particular eigenstate?

9.7 Suppose that the ground state of a quantum system described by Hamiltonian \mathcal{H} is $|\psi_{\lambda_0}\rangle$, and its first, second, and third excited states are $|\psi_{\lambda_1}\rangle$, $|\psi_{\lambda_2}\rangle$, and $|\psi_{\lambda_3}\rangle$ respectively. Furthermore, assume each eigenstate is associated with a corresponding eigenvalue $\lambda_0 < \lambda_1 < \lambda_2 < \lambda_3$, i.e., $\mathcal{H}|\psi_{\lambda_j}\rangle = \lambda_j|\psi_{\lambda_j}\rangle$ for $j = 0, 1, 2, 3$.

- (a) If we wanted to use the Abrams-Lloyd eigenvalue estimation algorithm to find the eigenvalue corresponding to the first excited state, what input would we provide to the quantum circuit shown in Fig. 9.7?
- (b) If instead of using the exact eigenvalue $|\psi_{\lambda_j}\rangle$ in the input to the Abrams-Lloyd circuit we used an approximate eigenvalue $|\widetilde{\psi}_{\lambda_j}\rangle$ how would this affect the probability of obtaining the correct eigenvalue? Quantify your answer with reference to the overlap between the exact eigenstate and the approximation to it, i.e., $\langle \widetilde{\psi}_{\lambda_j} | \psi_{\lambda_j} \rangle$.

Chapter 10

Mathematics on a Quantum Computer

“No one has yet discovered any warlike purpose to be served by the theory of numbers or relativity, and it seems unlikely that anyone will do so for many years.”

– G.H. Hardy¹

Pure mathematics has an uncanny habit of becoming more applicable with age. When G.H. Hardy wrote the words opening this chapter he clearly picked number theory and relativity as exemplifying the most useless topics a mathematician might concern himself with, unaware of the decisive roles they were to play in World War II via the Enigma machine and Fish Codes (i.e., cryptography) and the Hiroshima and Nagasaki bombs (i.e., nuclear weapons). I mention this because, in the past decade, a great many new quantum algorithms have been discovered in areas of pure mathematics that are considered arcane, abstract, esoteric, and quite possibly “useless”, by most engineers and applied scientists. The new quantum algorithms exhibiting superpolynomial speedups are mostly related to solving problems in algebraic number theory [225, 510], group theory [135, 226, 259, 437], and topology [543]. Moreover, other less dramatic but potentially more applicable, quantum algorithms have been discovered exhibiting polynomial speedups for solving problems in linear algebra [87], calculus [3, 258, 485, 496], optimization [89], functional analysis [74] and graph theory [161, 231]. Indeed, there are now so many mathematics-related quantum algorithms it has become difficult to survey them all with any degree of detail. Nevertheless, in this chapter I will survey a representative cross section of these mathematics-related quantum algorithms to give a taste of how quantum computing might impact mathematics.

10.1 Quantum Functional Analysis

Many quantum algorithms work with a black box function $f(x)$ which is assumed to be available as a quantum oracle. If you pose a question to this oracle you will

¹Source: “A Mathematician’s Apology”.

receive a yes/no answer in unit time. If you pose a superposition of questions, as you are allowed to do when the “questions” are quantum states, you obtain a corresponding superposition of answers. With a function $f(x)$ encoded as such an oracle, you can determine many properties of $f(x)$ in fewer steps than would be required classically. Examples include counting the number of solutions to $f(x) = 1$ [78], finding the mean value of $f(x)$ [220], finding the median of $f(x)$ [218], finding maxima and minima of $f(x)$ [157], and finding function collisions [74]. The quantum mean estimation algorithm is especially useful as it paves the way to quantum numerical integration too. So let’s take a more detailed look at quantum mean estimation. The other quantum algorithms for functional analysis are quite similar.

10.1.1 Quantum Mean Estimation

A central problem in functional analysis is to determine the mean value of a function, $f(x)$, for $x \in [a, b]$. If $f(x)$ is available as a quantum oracle, we can approach this via a technique known as quantum mean estimation.

Whatever the function is to begin with, we can always add a constant offset and re-scale it so as to ensure the function values are bounded between 0 and 1. Hence, without loss of generality, we assume $0 \leq f(x) \leq 1$ for $x \in [a, b]$.

To numerically estimate the mean of such an $f(x)$ over the interval $[a, b]$ we evaluate $f(x)$ at discrete sample points x_0, x_1, \dots, x_{N-1} such that $x_i \in [a, b]$ and then average the results. The greater the number of sample points, N , the more accurately we can estimate $\langle f(x) \rangle$. Classically, to estimate $\langle f(x) \rangle$ to order ϵ requires $\mathcal{O}(\frac{1}{\epsilon^2})$ such sample points.

Let us take $N = 2^n$ sample points to be $\{x_j : x_j = j\Delta x + a\}$ where $\Delta x = \frac{b-a}{N-1}$. Thus as j ranges over the integers from 0 to $N - 1$, x ranges over the reals from a to b . Without loss of generality we can therefore define a daughter function $\tilde{f}(j) = f(j\Delta x + a)$, so that $\langle \tilde{f}(j) \rangle$ sampled at $j = 0, 1, 2, \dots, N - 1$ matches $\langle f(x) \rangle$ sampled at $x = x_1, x_2, \dots, x_{N-1}$. Thus we can work with $\tilde{f}(j)$ in lieu of $f(x)$ in the knowledge that $\langle \tilde{f}(j) \rangle = \langle f(x) \rangle$ over corresponding intervals.

Thus, we can now state the quantum mean estimation algorithm:

Quantum Mean Estimation Find the mean value of $0 \leq \tilde{f}(j) \leq 1$ for j taking the values $j = 0, 1, 2, \dots, N - 1$.

1. We begin by defining

$$R_{\tilde{f}(j)} = \begin{pmatrix} \tilde{f}(j) & \sqrt{1 - \tilde{f}(j)^2} \\ \sqrt{1 - \tilde{f}(j)^2} & -\tilde{f}(j) \end{pmatrix} \quad (10.1)$$

to be a single qubit “rotation” that depends on $\tilde{f}(j)$.

2. Using $R_{\tilde{f}(j)}$ we construct the “mean value” operator $U_{\langle \tilde{f} \rangle}$ as follows:

$$U_{\langle \tilde{f} \rangle} = (\underbrace{H \otimes H \otimes \cdots \otimes H}_{n} \otimes \mathbb{1})^{-1} \cdot \left(\bigoplus_{j=0}^{2^n-1} R_{\tilde{f}(j)} \right) \cdot (\underbrace{H \otimes H \otimes \cdots \otimes H}_{n} \otimes \mathbb{1}) \quad (10.2)$$

where

$$\bigoplus_{j=0}^{2^n-1} R_{\tilde{f}(j)} = \begin{pmatrix} R_{\tilde{f}(0)} & 0 & 0 & 0 \\ 0 & R_{\tilde{f}(1)} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & R_{\tilde{f}(2^n-1)} \end{pmatrix} \quad (10.3)$$

3. With $U_{\langle \tilde{f} \rangle}$ so defined, we have:

$$|\psi_{\langle \tilde{f} \rangle}\rangle = U_{\langle \tilde{f} \rangle}|00\ldots 0\rangle = \langle \tilde{f} | 00\ldots 0\rangle + \cdots \quad (10.4)$$

which shows that the mean of $\tilde{f}(j)$ (and equally $f(x)$) appears as the amplitude of the $|00\ldots 0\rangle$ component of the output superposition $|\psi_{\langle \tilde{f} \rangle}\rangle$.

4. If we were to measure $|\psi_{\langle \tilde{f} \rangle}\rangle$ in the computational basis immediately after it was prepared, the probability of obtaining $|00\ldots 0\rangle$ would be $|\langle \tilde{f} |\rangle^2$. Hence, by this method, to estimate $\langle \tilde{f} \rangle$ to accuracy ϵ would require $\mathcal{O}(\frac{1}{\epsilon^2})$ repetitions, which is no better than we could have done classically via direct sampling.
5. The “trick” to *quantum* mean estimation is to amplitude amplify only the $|00\ldots 0\rangle$ component of $|\psi_{\langle \tilde{f} \rangle}\rangle$ by a known amount prior to making any measurement. Fortunately, by the theory of amplitude amplification developed in Sect. 5.4.2, we have a quantitative understanding of how successive applications of an amplitude amplification operator changes the amplitude. In the current case we know the identity of the target state to be amplified, i.e., $|00\ldots 0\rangle$, and we know that this is the only “target” we have. Hence, in this case the amplitude amplification operator takes the form $Q = -U_{\langle \tilde{f} \rangle} \mathbb{1}_s U_{\langle \tilde{f} \rangle}^\dagger \mathbb{1}_t$ with $|t\rangle = |00\ldots 0\rangle$ (the target) known a priori, and $|s\rangle$ the starting state.

10.1.2 Quantum Counting

Another common task in mathematics is to determine the number of solutions to a problem without necessarily wanting to exhibit those solutions explicitly. If the problem happens to be of a certain kind we can sometimes just write down its number of solutions. For example, an n -th order polynomial has n roots, so knowing the order of the polynomial tells us the number of solutions (at least up to repeated roots). But if we do not have any special properties to rely on the best we can do using classical methods is to enumerate all possible solutions and then count them.

This means we come to know the solutions first and then count them. If the problem is unstructured and there are N possible candidate solutions, we will incur a cost $\mathcal{O}(N)$ in counting the number of solutions.² The need to know the number of solutions to a problem is even more pressing in Grover's algorithm for unstructured quantum search. In Grover's original unstructured quantum search algorithm, one needs to know the number of solutions to the search problem in advance in order to know how many rounds of amplitude amplification to perform before measuring the amplitude amplified state at such time as to maximize your chances of obtaining a solution. If one under amplitude amplifies, or over amplitude amplifies, the probability of success will be less than $\mathcal{O}(1)$ when the final measurement is made.

Is there a better way using quantum computing that allows us to count the number of solutions without learning what they are explicitly?

The answer appears to be “yes” if the function whose solutions are sought is available as a quantum oracle. Fortunately, in 1998 Gilles Brassard, Peter Høyer and Alain Tapp combined ideas from Grover's algorithm with those from the quantum phase estimation algorithm to conceive of a quantum *counting* algorithm that can return a good estimate of the number of solutions, t , to an unstructured search problem [76, 78]. Knowing t allows a subsequent quantum search to be configured optimally by choosing the number of amplitude amplification steps to be performed to be $\mathcal{O}(\frac{\pi}{4}\sqrt{N/t})$, at which point the success probability will be $\mathcal{O}(1)$. The trick is to exploit the fact that the eigenvalues of Grover's amplitude amplification operator, Q , are related to the number of targets t . This means we can perform quantum counting of the number of solutions as an eigenvalue estimation problem.

Specifically, the quantum counting algorithm returns an estimate for the number of index values, j , for which the function $f(j) : \{0, 1, 2, \dots, N - 1\} \rightarrow \{0, 1\}$ returns the value 1. The quantum counting algorithm exploits the fact that the eigenvalues of Grover's amplitude amplification operator, $Q = \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ -\sin 2\theta & \cos 2\theta \end{pmatrix}$, are related to the number of solutions, t , to the search problem. Thus by estimating the eigenvalues of Q using the quantum eigenvalue estimation algorithm, one infer t . The quantum search algorithm for finding one of t solutions, and hence the operator Q , was explained in Chap. 5. Likewise the quantum eigenvalue estimation was explained in Chap. 9. So once we understand how the eigenvalues of Q are related to the number of solutions t , the quantum counting algorithm follows quite easily.

The Grover amplitude amplification operator, Q , in the $\{|\psi_{\text{good}}\rangle, |\psi_{\text{bad}}\rangle\}$ basis, when there are t solutions out of $N = 2^n$ possibilities is (as shown in Sect. 5.5.1):

$$Q = \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ -\sin 2\theta & \cos 2\theta \end{pmatrix} \quad (10.5)$$

with $\sin \theta = \sqrt{t/N}$ where t is the unknown number of solutions that we seek, and $N = 2^n$ is the number of candidates.

To obtain t , we make use of the fact that Q has two eigenvalues and associated eigenvectors: the eigenvalues $e^{-2i\theta}$ is associated with the eigenvector $|\psi_+\rangle =$

²Big Θ notation is discussed in Sect. 4.4.2.

$\frac{i}{\sqrt{2}}|\psi_{\text{good}}\rangle + \frac{1}{\sqrt{2}}|\psi_{\text{bad}}\rangle$, and the eigenvalue $e^{2i\theta}$ is associated with the eigenvector $|\psi_-\rangle = -\frac{i}{\sqrt{2}}|\psi_{\text{good}}\rangle + \frac{1}{\sqrt{2}}|\psi_{\text{bad}}\rangle$. Thus the eigenvalues depend on θ , and θ depends on t . Hence, if we can find the eigenvalue $e^{-2i\theta}$ given knowledge of $|\psi_+\rangle$, or if we can find $e^{2i\theta}$ given knowledge of $|\psi_-\rangle$, we will be able to compute θ and hence $t = N\theta^2$.

Unfortunately, there is a problem: we do not know $|\psi_+\rangle$, and $|\psi_-\rangle$ because we do not know the two states from which they are built, i.e., $|\psi_{\text{good}}\rangle$ and $|\psi_{\text{bad}}\rangle$. Luckily, though, we can write a state that is easy to make—the equally weighted superposition state—as a sum of $|\psi_+\rangle$, and $|\psi_-\rangle$. Specifically, we have:

$$\begin{aligned} |\psi\rangle &= \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \\ &= \sin \theta |\psi_{\text{good}}\rangle + \cos \theta |\psi_{\text{bad}}\rangle \\ &= \left(\frac{ie^{-i\theta} - ie^{i\theta}}{2} \right) |\psi_{\text{good}}\rangle + \left(\frac{e^{-i\theta} + e^{i\theta}}{2} \right) |\psi_{\text{bad}}\rangle \\ &= \frac{e^{i\theta}}{\sqrt{2}} \left(\frac{i}{\sqrt{2}} |\psi_{\text{good}}\rangle + \frac{1}{\sqrt{2}} |\psi_{\text{bad}}\rangle \right) + \frac{e^{-i\theta}}{\sqrt{2}} \left(-\frac{i}{\sqrt{2}} |\psi_{\text{good}}\rangle + \frac{1}{\sqrt{2}} |\psi_{\text{bad}}\rangle \right) \\ &= \frac{e^{i\theta}}{\sqrt{2}} |\psi_+\rangle + \frac{e^{-i\theta}}{\sqrt{2}} |\psi_-\rangle \end{aligned} \quad (10.6)$$

Now we can use this state as the known input to the eigenvalue estimation algorithm of Sect. 9.5.

The quantum counting scheme is illustrated in Fig. 10.1.

Quantum Counting Suppose we are given $f(x) \rightarrow \{0, 1\}$ as a black box quantum oracle such that $f(x) = 1$ if $x \in \mathcal{G}$ (the “good” states) and $f(x) = 0$ if $x \in \mathcal{B}$. Find the number of values of x for which $f(x) = 1$.

1. Prepare $|\psi_0\rangle = |0\rangle^{\otimes(p+n)}$.
2. Apply a Walsh-Hadamard gate to each qubit to give $|\psi_1\rangle = H^{\otimes p} \otimes H^{\otimes n} |\psi_0\rangle$.
3. Apply a cascade of controlled- G^{2^j} gates for $j = 0, 1, \dots, p-1$.
4. Measure the second (n -qubit) register.
5. Apply $\text{QFT}_{2^p}^\dagger$ to the first (p -qubit) register.
6. Measure the bits values output from the first register. Let the result be the binary string $(b_1 b_2 \dots b_p)_2$.
7. Compute $m = 2^{p-1}b_1 + 2^{p-2}b_2 + \dots + 2^0b_p$.
8. Estimate the number of solutions, \tilde{t} , from $\tilde{t} = 2^n \sin^2(\frac{m\pi}{2^p})$.

In the quantum counting algorithm we must pick the number of bits of precision to which we wish to estimate the eigenvalue (and hence the precision with which we can estimate the number of solutions t). Without knowing the number of solutions in advance it is hard to know what precision to use. So the way the quantum counting

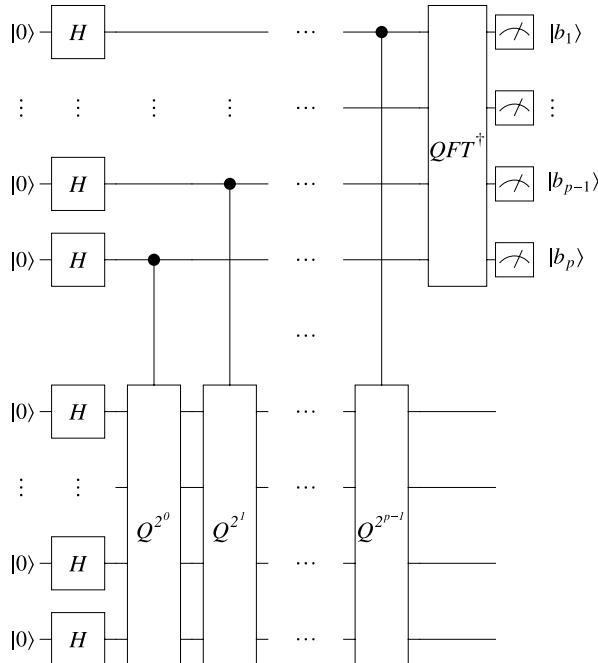


Fig. 10.1 Quantum circuit for quantum counting the number of solutions to a search problem with amplitude amplification operator Q . Note that the Walsh-Hadamard gates acting on the bottom set of qubits creates the uniform superposition $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$. However, this state may also be *interpreted* as a superposition of the two (unknown) eigenvectors of Q , i.e., $|\psi_+\rangle$ and $|\psi_-\rangle$. Specifically, we have $|\psi\rangle = \frac{e^{i\theta}}{\sqrt{2}} |\psi_+\rangle + \frac{e^{-i\theta}}{\sqrt{2}} |\psi_-\rangle$. So the output from the top set of qubits are the sequence of bits in the binary fraction expansion of one or other of the eigenvalues of Q

Table 10.1 Accuracy of the estimate in the number of solutions return by quantum counting as a function of the precision used in the inverse QFT

Precision p	Provable bound on t
$p = c\sqrt{N}$	$ \tilde{t} - t < \frac{2\pi}{c}\sqrt{t} + \frac{\pi^2}{c^2}$
$p = c\sqrt{\frac{N}{t}}$	$(1 + \frac{\pi}{c})^2 < \frac{\tilde{t}}{t} < 1/((1 + \frac{\pi}{c})^2)$
$p = c\sqrt{\tilde{t}N}$	$\tilde{t} = t$ with bounded probability

algorithm should be used is to pick a number of bits of precision $p = \sqrt{N}$ and obtain a first estimate of the number of solutions \tilde{t} such that $|t - \tilde{t}| < 2\pi\sqrt{t} + \pi^2$. Then reset the precision to $p = 20\sqrt{\tilde{t}N}$ and run the quantum counting algorithm again to obtain a better estimate of \tilde{t} . Following this procedure, one obtains the true t with probability of at least $8/\pi^2$ [76, 78]. Various other relations are now known, as summarized in Table 10.1, that bound the accuracy with which we know t to the precision p used in the quantum counting algorithm.

10.2 Quantum Algebraic Number Theory

“Hallgren has recently shown that the Fourier transform can also be used to find the periodicity of functions with irrational periods, and that this is useful in solving certain number theory problems such as finding solutions to Pell’s equation and finding class groups of number fields.”

– Peter Shor

The discovery of Shor’s algorithm for factoring composite integers in polynomial time stimulated the search for other quantum algorithms that solve similar mathematical problems in an analogous way. One of the best algorithms that resulted is Sean Hallgren’s algorithm for solving Pell’s equation [225].

If you are not a mathematician with a penchant for algebraic number theory most likely you are wondering “Who the heck is Pell?” and “What on Earth is Pell’s equation?”. Indeed, Pell’s equation is not often encountered outside of specialty mathematics courses. So on the face of it, it may seem a tad esoteric to worry about a quantum algorithm for solving an obscure equation. But Hallgren’s algorithm is important primarily because it extends the period-finding technique in Shor’s algorithm in a new direction. In particular, whereas Shor’s period finding algorithm can find, efficiently, the period of a periodic function having an *integer* period, Hallgren’s algorithm can find, efficiently, the period of a periodic function having an *irrational* period. The significance of this extension is lost on many people because they assume, wrongly, that you could simply approximate the irrational period by increasing the rate at which you sample the function so that the irrational period will be “close” to a rational number. Unfortunately, this strategy fails miserably because no matter how finely you partition the domain of the function into rational intervals the rounding errors you introduce when you try to sample a function with an irrational period on a grid whose spacing is in rational numbers messes up the period by too large a factor. So Hallgren’s algorithm deserves a special place in the history of quantum computing as being genuinely distinct from the period finding algorithm invented by Peter Shor.

Before looking at the details of Hallgren’s algorithm, let us review Pell’s equation and understand why it is so hard to solve.

10.2.1 The Cattle Problem of Archimedes and Pell’s Equation

John Pell was an English mathematician who lived between 1610–1685, and he turns out to have nothing whatsoever to do with the equation that bears his name! The mis-attribution can be traced back to Euler who confused Pell with another English mathematician William Brouncker (1620–1684) who had devised a method of solution [307]. But in fact the equation had been studied even earlier by Greek and Indian mathematicians. In particular, one of the earliest references to the equation was implicit in the “Cattle Problem” posed by Archimedes (287–212 B.C.) in rhyming verse concerning the numbers of cows and bulls of various colors (white,

black, dappled, and brown) belonging to the Sun god on the island of Sicily. The poem defined various relations between the numbers of cattle of different colors and genders and the final constraint Archimedes included is an instance of Pell's equation. See [307], p. 184 for an English translation of the “Cattle Problem of Archimedes”.

Pell's equation is any equation of the form:

$$x^2 - dy^2 = 1 \quad (10.7)$$

where d is an integer that is not a perfect square (i.e., the square root of d is required to be non-integer). The stipulation on d is easily motivated since if d were a perfect square, we would have $d = \tilde{d}^2$ for some integer \tilde{d} , and so Pell's equation would then read $x^2 - \tilde{d}^2 y^2 = x^2 - (\tilde{d}y)^2 = 1$, which would imply there are two integers that when squared are 1 apart, and this is clearly impossible. A solution to Pell's equation is deemed to be any *pair* of integers (x_j, y_j) for which the equality holds. However, the composite formula $x_j + \sqrt{d}y_j$ is often referred to as a “solution” where the \sqrt{d} factor plays a similar role in quadratic numbers to that played by $i = \sqrt{-1}$ in complex numbers. The equation has a smallest solution (x_1, y_1) expressed in the form:

$$x_1 + y_1 \sqrt{d} \quad (10.8)$$

This is called the *fundamental* solution because all the other solutions can be written as powers of the fundamental solution. Specifically, the j -th solution is the j -th power of the fundamental solution.

$$x_j + y_j \sqrt{d} = (x_1 + y_1 \sqrt{d})^j \quad (10.9)$$

Therefore, our goal is to find the fundamental solution, because once it is known we can easily find all the other solutions. Any given instance of Pell's equation has infinitely many solutions.

10.2.2 Why Solving Pell's Equation Is Hard

Table 10.2 shows some examples of the fundamental solution for increasing values of d . As you can see, the size of the solutions can vary widely with different values of d and no obvious pattern is apparent.

In the case of the cattle problem of Archimedes $d = 410286423278424$ (i.e., d is expressed in 15 digits) yet it takes 206,545 digits to express the smallest solution explicitly! In the worst case, if the input size of Pell's equation is taken to be d , the solution can be of order $\mathcal{O}(e^{\sqrt{d}})$. So merely *writing down* the fundamental solution will require $\mathcal{O}(\sqrt{d})$ digits which scales exponentially in d . The fact that one has to work with such unwieldy integers is what makes solving Pell's equation so hard.

Table 10.2 A sampling of the fundamental solutions of Pell equations for $2 \leq d \leq 62$ illustrating the wildly different solution values for different values of d

d	x	y	$x^2 - dy^2$
2	3	2	1
3	2	1	1
5	9	4	1
6	5	2	1
7	8	3	1
8	3	1	1
10	19	6	1
11	10	3	1
12	7	2	1
13	649	180	1
14	15	4	1
15	4	1	1
⋮	⋮	⋮	⋮
27	26	5	1
28	127	24	1
29	9801	1820	1
30	11	2	1
31	1520	273	1
32	17	3	1
33	23	4	1
⋮	⋮	⋮	⋮
58	19603	2574	1
59	530	69	1
60	31	4	1
61	1766319049	226153980	1
62	63	8	1

10.2.3 Solution by Finding the “Regulator”

Fortunately, there are a couple of ways to circumvent this problem. The first is to use a “power product” representation of the fundamental solution. For example, using power products the fundamental solution to the Cattle Problem of Archimedes can be expressed as $x_1 + y_1\sqrt{d} = \frac{2^{45}14(2175+\sqrt{d})^{18}(2184+\sqrt{d})^{10}(2187+\sqrt{d})^{20}(4341+2\sqrt{d})^6}{3^{27}7^529^931^{20}(2162+\sqrt{d})^{18}(4351+2\sqrt{d})^{10}}$ [307] (p. 190 therein). This is a much more compact representation that the required 206,545 digits to express the fundamental solution explicitly.

An alternative technique is to work with the *logarithm* of the fundamental solution, rather than the fundamental solution directly. The logarithm, $R = \log_e(x_1 + \sqrt{d}y_1)$, is called the *regulator*, which is an *irrational* number. Given the regulator

you can compute the power product representation of the solution. Moreover, the j -th solution to Pell's equation, $x_j + \sqrt{d}y_j$, is just the fundamental solution raised to the j -th power, $x_j + \sqrt{d}y_j = (x_1 + \sqrt{d}y_1)^j$ and so the j -th solution must also be given by jR .

10.2.4 The Regulator and Period Finding

The main thrust of Hallgren's algorithm for solving Pell's equation is to set up a periodic function whose period is R , the regulator, which is generally an *irrational* number. Then, by finding the period one finds the regulator and hence the fundamental solution from $R = x_1 + \sqrt{d}y_1$. Once the fundamental solution is known all others follow trivially, as integer multiples of the regulator, and hence Pell's equation is solved.

The translation from Pell's equation to period finding uses well-established mathematics. The novelty in Hallgren's algorithm is that he adapts the (integer) period finding algorithm underpinning Shor's algorithm to the case when the period is an irrational number. Thus for our purposes, we will focus only on the quantum aspect of Hallgren's algorithm. Readers interested in understanding the underlying algebraic number theory behind Pell's equation should read the excellent review paper on the topic written by Richard Jozsa [263].

10.2.5 Quantum Core of Hallgren's Algorithm

Thus the core of Hallgren's algorithm is a quantum technique for determining the period of a periodic function that possesses an irrational period.

The fact that period is irrational complicated periodic finding considerably. To see this, imagine a periodic function over the real numbers. We might sample this function at tick marks falling on rational numbers, $0, \frac{1}{3}, \frac{2}{3}, \dots$ etc. If the period of our function is an integer, then the function will repeat *exactly* after some number of tick marks. But if its period is an irrational number, this will no longer be true. Worse still the distance between the end of the true period and the closest integer to the true period will jump around erratically in each successive period. You can see this visually from Fig. 10.2.

10.2.6 Hallgren's Quantum Algorithm for Solving Pell's Equation

It turns out, if we can find the integer *closest* to the regulator, R , (either above or below it) there is a classical algorithm that will compute R to n digit accuracy in a time that scales as $\text{poly}(n, \log d)$. So to find R , it suffices to find the integer closest to R .

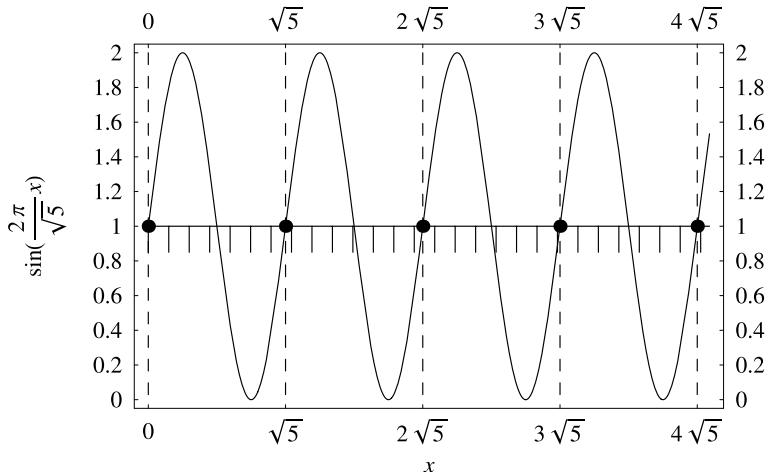


Fig. 10.2 When a period function having an irrational period is sampled on a grid whose points coincide with rational numbers, the true period is hard to recognize. The function shown is a sinusoid with period $\sqrt{5}$ sampled on a grid with spacings in increments of $1/3$. The beginnings and ends of periods are marked by dots. Notice that the dot at the end of the first period ($x = \sqrt{5}$) is closest to a tick mark on its right, whereas the dot at the end of the second period ($x = 2\sqrt{5}$) is closest to a tick mark on its left. Rounding to the nearest tick mark therefore causes us to miscalculate the true period

In Hallgren's algorithm we will make use of the following notation:

- $\lfloor \ell \rfloor$ = the closest integer less than ℓ
 - $\lceil \ell \rceil$ = the closest integer greater than ℓ
 - $[\ell]$ = an integer that is either $\lfloor \ell \rfloor$ or $\lceil \ell \rceil$
 - $\lfloor \ell \rfloor$ = the closest integer above or below ℓ implying $|\ell - \lfloor \ell \rfloor| \leq 1/2$
- (10.10)

The algorithm then works as follows.

Richard Jozsa [263], and more recently Andrew Childs and Win van Dam [104] have both written beautiful accounts of Hallgren's algorithm for solving Pell's equation and related mathematical problems. Below I give a less formal account of Hallgren's algorithm following the development of Childs and van Dam. The reader wanting details of the algebraic number theory behind Pell's equation should consult one of these texts.

Hallgren's Algorithm for Solving Pell's Equation Problem: Given a function $f(x)$ defined on the reals that has an *irrational* period R , i.e., $f(x) = f(x + R)$, find the *closest* integer to R .

- Create a uniform superposition of $x \in \mathbb{Z}/N\mathbb{Z}$:

$$|\psi_0\rangle = \sum_{x \in \mathbb{Z}/N\mathbb{Z}} |x\rangle |0\rangle \quad (10.11)$$

- Evaluate $f(x)$ on these values and place the result in the second register—the “ancilla”:

$$|\psi_1\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}/N\mathbb{Z}} |x, f(x)\rangle \quad (10.12)$$

- Measure the ancilla register, thereby preparing the first register in a superposition of values of x that all take the same value for $f(x)$. Thereafter we can ignore the ancilla register as it plays no further role. The first register is, however, left in a state of the form:

$$|\psi_2\rangle = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} |x_0 + [jr]\rangle \quad (10.13)$$

where “[jr]” can be either $\lfloor jr \rfloor$ or $\lceil jr \rceil$ and which it is jumps around erratically from term to term.

- Compute the QFT over the cyclic group $\mathbb{Z}/N\mathbb{Z}$. Setting ω_N to be the N -th root of unity, i.e., $\omega_N = \exp(2\pi i/N)$ we have:

$$|\psi_3\rangle = \frac{1}{\sqrt{nN}} \sum_{k \in \mathbb{Z}/N\mathbb{Z}} \omega_N^{kx_0} \sum_{j=0}^{n-1} \omega_N^{k[jr]} |k\rangle \quad (10.14)$$

This state is similar to the post-QFT state one obtains in Shor’s algorithm, for which the sought after period is guaranteed to be an *integer*. In the case of Hallgren’s algorithm, the period is, instead, an *irrational* number. But we can estimate by how much the amplitudes of eigenstate $|k\rangle$ differ in the case of Shor’s algorithm versus Hallgren’s algorithm. In Hallgren’s algorithm, the exponent contains the term $[jr] = jr + \delta_j$ where $-1 < \delta_j < 1$, and so $\sum_{j=0}^{n-1} \omega_N^{k[jr]} |k\rangle = \sum_{j=0}^{n-1} \omega_N^{kjr} \omega_N^{k\delta_j}$. If all the δ_j were zero, this would reduce to the form found in Shor’s algorithm. Hence, the difference in the amplitudes of eigenstate $|k\rangle$ in Hallgren’s algorithm versus Shor’s is the following:

$$\begin{aligned} \left| \underbrace{\sum_{j=0}^{n-1} \omega_N^{kjr} \omega_N^{k\delta_j}}_{\text{Hallgren}} - \underbrace{\sum_{j=0}^{n-1} \omega_N^{kjr}}_{\text{Shor}} \right| &\leq \sum_{j=0}^{n-1} |\omega_N^{k\delta_j} - 1| \\ &\leq \frac{1}{2} \sum_{j=0}^{n-1} \left| \frac{\pi k \delta_j}{N} \right| \leq \frac{\pi k n}{2N} \end{aligned} \quad (10.15)$$

which is *not necessarily small* for large enough values of k . Therefore, in Hallgren's algorithm, when we measure the post-QFT state (as we are about to do) we will only accept the result as "usable" whenever the k value we find is such that $k < N/\log r$. Luckily, this event occurs with probability $\Omega(1/\text{poly}(\log r))$ (cf. $\Omega(1)$ in the case of Shor's algorithm). But $\Omega(1/\text{poly}(\log r))$ is still a pretty high probability.

5. Measure $|\psi_3\rangle$ in the computational basis. This provides a sample from the QFT which is strongly peaked around values of $k = \lfloor \frac{jN}{r} \rfloor$ such that $j \in \mathbb{Z}$, which we only deem to be "usable" if $k < N/\log r$. The probability of obtaining such a k satisfying $k < N/\log r$ is $\Omega(1/\text{poly}(\log r))$.
6. Finally, we obtain an estimate of r from these samples. But because r is not an integer the continued fraction trick used in Shor's algorithm needs to be modified in Hallgren's algorithm. Specifically, by re-running steps 1–5 only polynomially many times we obtain *two* samples $k_1 = \lfloor \frac{j_1 N}{r} \rfloor$ and $k_2 = \lfloor \frac{j_2 N}{r} \rfloor$ such that j_1 and j_2 are relatively prime. Hallgren showed, provided $N \geq 3r^2$, that j_1/j_2 is guaranteed to be a convergent in the continued fraction expansion of k_1/k_2 . From this we can compute j_1 and hence the regulator, r , from

$$\left| r - \left\lfloor \frac{jN}{\lfloor jN/r \rfloor} \right\rfloor \right| \leq 1 \quad (10.16)$$

So what happens if we Fourier sample. Well for any integer period function we have a repeating superposition. Compute the FT and compute the continued fraction expansion on the result. When the period is integer it is sufficient to take one period and analyze to get the while thing. But when the period is irrational can't do that.

10.2.7 What Is the Significance of Pell's Equation?

One might wonder why solving such an esoteric problem as Pell's equation really matters? The answer lies in the fact that solving Pell's equation appears to be a slightly harder problem than factoring composite integers or computing discrete logarithms, and yet it too admit a polynomial time algorithm on a quantum computer. In particular, it is known that solving Pell's equation is at least as hard as factoring. The fastest classical factoring algorithm is the Number Field Sieve, which has complexity of order $O(\exp(n^{1/3}))$. In contrast, the fastest known classical algorithm for finding the regulator has complexity of order $O(\exp(n^{1/2}))$. Moreover, there is a polynomial time reduction from factoring to finding the regulator, but not vice versa. This implies that finding the regulator is at least as hard as factoring. Nevertheless, in the quantum domain, both problems can be solved in polynomial time on a quantum computer. Thus Hallgren's quantum algorithm for solving Pell's equation gives a distinct example of a superpolynomial speedup separation between the quantum and classical domains on a problem that has been known to mathematics since antiquity. Moreover, whereas Shor's algorithm can find the period of periodic

functions having an integral period, Hallgren's algorithms extends this to periodic functions having an irrational period. This is a significant advance in the field of quantum algorithms.

10.3 Quantum Signal, Image, and Data Processing

An important potential use of quantum computers, and a relatively under researched area, is to speed up certain signal, image, and data processing tasks. For example, in genomics, we might want a faster way to compare sequences [235]. Or we might want to match an image against a template [431]. To be able to perform such tasks on a quantum computer it is first necessary to create a quantum state that encodes the signal, image, or data. Once in this form we may then operate on it using quantum gates.

10.3.1 Classical-to-Quantum Encoding

A good example of the general data encoding problem is provided by the needs of image processing. Suppose we are given an image as a $N \times N = 2^n \times 2^n$ array of pixel values. The first step in quantum image processing is to create a pure quantum state that encodes this image. One way to do so is to append the columns of the image on top of one another, as we move from left to right across the pixel array, to obtain an $N^2 \times 1 = 2^{2n} \times 1$ dimensional column vector of pixel values. We then renormalize these pixel values so that sum of the squares of their absolute values equals one. The result is a sequence of real numbers that encodes the relative magnitudes of the pixel values in the image. Next, we simply choose to *interpret* this sequence of (now normalized) real numbers as the sequence of amplitudes of the successive eigenstates of a $2n$ -qubit pure quantum state. Let us call this state $|\psi_{\text{image}}\rangle$. Thus, the problem of encoding an arbitrary image in a quantum state reduces to the problem of synthesizing the state $|\psi_{\text{image}}\rangle$.

Fortunately, there are several ways this can be done [42, 223, 267]. A method based on Grover's algorithm was given in Sect. 5.7.2. Another method is to construct the matrix having the column vector of amplitudes implicit in $|\psi_{\text{image}}\rangle$ as the first column, and 1's down the main diagonal elsewhere [465].

$$\begin{pmatrix} \uparrow & & \\ |\psi_{\text{image}}\rangle & \ddots & \\ \downarrow & & 1 \end{pmatrix} \equiv \begin{pmatrix} p'_0 & & & \\ p'_1 & 1 & & \\ \vdots & & \ddots & \\ p'_{N^2-1} & & & 1 \end{pmatrix} \quad (10.17)$$

Then we apply the Gram-Schmidt orthogonalization procedure to this matrix. For an explanation of this see pp. 230–231 [204]. This result will be a unitary matrix.

A quantum circuit for this matrix can be found using the GSVD (Generalized Singular Value Decomposition) of Sect. 3.7.4.

The basic algorithm for synthesizing any pure state that has an arbitrary amplitude sequence is as follows [465]:

Algorithm SynthesizePureState

Input: Specification of the desired pure state $|\psi\rangle = \sum_{i=0}^{2^n-1} c_i |i\rangle$.

Goal: Find a quantum circuit that synthesizes $|\psi\rangle$.

1. Without loss of generality, assume amplitude $c_0 \neq 0$ (otherwise perform an operation that shuffles the amplitudes in the column vector such that the top one is non-zero).
2. Define the matrix, M , such that the first column of this matrix is the sequence of amplitudes in $|\psi\rangle$, the remaining entries down the main diagonal are 1's and the remaining elements are 0's, i.e., define M as:

$$M = \begin{pmatrix} c_0 & & & \\ c_1 & 1 & & \\ \vdots & & 1 & \\ \vdots & & & \ddots \\ c_{2^n-1} & & & 1 \end{pmatrix} \quad (10.18)$$

3. Use the Gram-Schmidt procedure to compute a set of orthonormal columns for the rest of the matrix, i.e. $U := \text{GramSchmidt}(M)$
4. The resulting matrix, U , is unitary. Find a quantum circuit for U using QCD.

Such state synthesis capability is very useful. For example, in addition to its role in encoding images, we might need the ability to create arbitrary superpositions in order to perform such tasks as the analysis of excited-state molecular spectra in quantum chemistry [2, 25, 304]. As we showed in Chap. 9 such tasks may require us to synthesize states that are close to a desired excited state, in order to determine the corresponding eigenvalue.

Using the `SynthesizePureState` algorithm, or other method, it is possible to synthesize a state such as $|\psi_{\text{image}}\rangle$ that encodes an arbitrary image in a quantum state in a number of steps that scales as $\mathcal{O}(2^{2n}) = \mathcal{O}(N^2)$. This is exponential in the number of qubits, $2n$, which looks bad. But in reality, whenever we handle an image on a classical computer we expect to incur costs that are at least linear in the size of the image. For example, the cost of handling an $N \times N$ image classically is $\mathcal{O}(N^2) = \mathcal{O}(2^{2n})$. So the cost of classical-to-quantum encoding of an arbitrary image is no that bad. If, subsequently, the quantum encoded image can be processed exponentially faster, than this encoding would be a price worth paying.

Furthermore, there are opportunities in quantum image encoding that are not present in classical image encoding. For example, if we have a pair of co-registered images taken in two different wavebands, such as visible and infrared, we could encode the visible image in a sequence of purely real numbers, and the infrared

image in a sequence of purely imaginary numbers, then add these two together, re-normalize the result, and use this as the basis for defining the amplitude sequence of our $2n$ -qubit pure state. The amplitude sequence will now consist of complex numbers, whose real part encodes the visible image and whose imaginary part encodes the infrared image. Subsequently, any manipulations of the quantum-encoded image essentially manipulate the visible and infrared images simultaneously, at no additional computational cost!

10.3.2 Quantum Image Processing: 2D Quantum Transforms

In Chap. 3 we described how to apply the QFT, QCT, and QWT to state vectors, and we saw that exponential speedups are quite common [288]. This is analogous to applying the DFT, DCT and DWT to one-dimensional signals or one-dimensional time series. However, in classical computing we also employ the DFT, DCT and DWT to transform *two*-dimensional data such as images, stored as arrays of pixel values, i.e., positive integers between 0 and 255 representing gray levels. It is natural to ask how are we to compute the QFT, QCT and QWT of two-dimensional data sets in the quantum domain?

As soon as we try to do this we hit a problem. Generically, if we have a unitary transform U , say, which could be the one-dimensional QFT, QCT or QWT, then the two-dimensional transform on an image, \mathcal{I} , encoded as an array of pixel values, is computed by first applying U to each column of \mathcal{I} independently, and then to each row of the result, independently. Overall, as the rows and columns of U are orthonormal for the U matrices we care about, this sequence of operations also happens to be writable in a more compact and elegant way. Specifically, if U is the one-dimensional transform, then the two-dimensional transform acting on an image \mathcal{I} can be written as:

$$U \cdot \mathcal{I} \cdot U^T \quad (10.19)$$

where T denotes taking the transpose and is equivalent to computing the matrix inverse (since the U matrices we care about are orthonormal). Classically, such a transform poses no problem.

Trying to compute such two-dimensional transforms on a quantum computer as a product of matrices is more problematic. For example, one might think to encode an image in a density matrix ρ because ρ is a naturally two-dimensional representation of a quantum state. However, an array representing an image can contain an arbitrary distribution of pixel values. But density matrices have to have conform to certain symmetries, e.g., their trace must be 1. So it is not possible to merely reinterpret an image array, \mathcal{I} , as a density matrix ρ . Instead, we have to stay with the idea of encoding an image in a pure state, ψ_{image} , made by extracting each column of \mathcal{I} and appending it to the base of the previous column. This is the so-called “vec” operation. Having computed $\text{vec}(\mathcal{I})$ and renormalizing the resulting column vector gives us a description of the amplitude sequence in a pure state that is sufficient

to encode an arbitrary image. Then the traditional image transformation $\mathcal{I} \rightarrow U \cdot \mathcal{I} \cdot U^T$ can be re-expressed as an equivalent operation on $\text{vec}(\mathcal{I})$. Specifically, the connection is:

$$\text{vec}(U \cdot \mathcal{I} \cdot U^T) \equiv (U \otimes U) \cdot \text{vec}(\mathcal{I}) \quad (10.20)$$

Thus, once we have one-dimensional version of any unitary transform, such as the 1D QFT, QCT or QWT, we can obtain the analogous two-dimensional transform on a 2D image or 2D data set using the aforementioned trick.

If these 2D transforms are augmented with the ability to synthesize an arbitrary pure state (which is needed to input an image into a quantum computer) we have the beginning of a method to encode an image in a quantum state, and then operate upon it using a two-dimensional version of our favorite quantum transform.

10.3.3 Quantum-to-Classical Readout

The final stage of image processing requires extracting information from the result. Traditionally, in classical computing, this is done by looking at the image that is the product of the various transformations we have applied. However, we cannot, of course, do this in the quantum domain as our transformed images will be stored in quantum states not classical states. So any attempt to observe them in the traditional face will destroy the very information we are trying to visualize.

The alternative is to make a measurement that either samples from the transformed image, or extracts some collective property of the transformed image. The easiest example is the provided by QFT. If one encodes a one-dimensional time-series signal in a pure state of n qubits, quantum Fourier transforms it, and then reads the quantum-transformed image, one is most likely to obtain a result that corresponds to a peak in the Fourier transformed signal. This indicates that the corresponding frequency component is strongly represented in the Fourier transform of the signal. This measurement therefore gives you information about the transformed signal, without you having to “see” the whole signal in the conventional sense.

It is early days yet for understanding the full gamut of methods able to extract useful information from quantum-transformed images. Nevertheless, we know we need to give up our assumption that we have to be able to “see” a processed image to be able to glean useful information from it. Instead, we need to shift our thinking to classes of measurements on the processed image that can reveal its properties even without seeing the processed image explicitly.

10.4 Quantum Walks

An altogether different area of mathematics where quantum computers might play a role is in the study of “random walks”. A classical random walk is a mathematical process that starts at a particular point, in some real or abstract space, and takes

Table 10.3 Comparison of quantum versus classical random walk adapted from Viv Kendon [274]. Algorithms for the one dimensional classical random walk (left) and its quantum counterpart (right). A quantum walker at position x with a coin in state c is denoted $|x, c\rangle$. The quantum analogs of the coin toss and stepping operation are embodied in the operators \mathbf{C} and \mathbf{S} . Note that the quantum walker never observes the outcomes of the coin tosses. If he did, this would project the quantum walk into a classical walk

Classical random walk	Quantum random walk
1. Start at the origin: $x = 0$	1. Start at the origin: $x = 0$
2. Initialize the coin	Initialize the coin
3. Toss a coin result is HEAD or TAIL	$\mathbf{C} x, 0\rangle \longrightarrow (a x, 0\rangle + b x, 1\rangle)$ $\mathbf{C} x, 1\rangle \longrightarrow (c x, 0\rangle + d x, 1\rangle)$
4. Move one unit left or right according to coin state: TAIL: $x \longrightarrow x - 1$ HEAD: $x \longrightarrow x + 1$	3. Move one unit left and right according to qubit state $\mathbf{S} x, 0\rangle \longrightarrow x - 1, 0\rangle$ $\mathbf{S} x, 1\rangle \longrightarrow x + 1, 1\rangle$
5. Repeat steps 2 and 3 t times	4. Repeat steps 2 and 3 t times
6. Measure position $-t \leq x \leq t$	5. Measure position $-t \leq x \leq t$
7. Repeat steps 1 to 5 many times → prob. dist. $P(x, t)$, binomial standard deviation $\langle x^2 \rangle^{1/2} = \sqrt{t}$	6. Repeat steps 1 to 5 many times → prob. dist. $P(x, t)$ has standard deviation $\langle x^2 \rangle^{1/2} \propto t$

successive steps in random directions. The next direction to step is determined by the toss of a coin (or multi-sided dice etc.) and the step size is fixed. Usually, one wants to know the typical distance from the starting point, or the probability of reaching an absorbing boundary, or the number of times a given point is re-visited, as functions of the number of steps taken by the walker. Such random walks are usually constrained, depending on their intended purpose, to take place along a line, on a surface, within a volume, or even on a graph.

Random walks are interesting in their own right as part of pure probability theory. However, they have also found application in almost every scientific field ranging from the physics of Brownian motion on the molecular scale, to the creation of stars and galaxies, on the cosmic scale. Random walks find applications across all intermediate scales too and have been used to shed light on phenomena ranging weather prediction, biology, sociology, and even financial markets.

The simplest type of random walk is the one-dimensional walk on a line, sometimes called the “drunkard’s walk”. The idea is that a drunkard is wandering along a line and chooses to step right or left depending on the outcome of the toss of a coin. A particular realization of the random walk is created by repeatedly tossing a coin, reading the outcome, and then stepping one space to the left or right depending on the result. This process is summarized in the left hand column of Table 10.3.

Some important questions to ask about such a classical random walk is how far, typically, the walker will be from his starting point after t steps, and what

is the uncertainty (or variance) in this estimate. We can calculate these quantities quite easily: Let the walk consist of a total of t steps, of which t_R are to the right and t_L are to the left. Clearly, $t = t_R + t_L$. Let the probability of a rightward step be p_R and that of a leftward step be $q_L = 1 - p_R$. Each of these rightward steps can occur in any order amongst the t trials. The number of possible orderings in which the rightward steps could be taken is $\binom{t}{t_R}$. Each such walk occurs with probability $p_R^{t_R} q_L^{t_L} = p_R^{t_R} (1 - p_R)^{t-t_R}$. Hence, the overall probability of a walk having t_R rightward steps is $\Pr(t_R) = \binom{t}{t_R} p_R^{t_R} (1 - p_R)^{t-t_R}$. Hence, the mean number of rightward steps taken is $\langle t_R \rangle = p_R t$ and the mean number of leftward steps is $\langle t_L \rangle = q_L t = (1 - p_R)t$. Moreover, the variance in the number of rightward steps (which equals the variance in the number of leftward steps) is $\sigma_R^2 = \langle t_R^2 \rangle - \langle t_R \rangle^2 = t p_R q_L = t p_R (1 - p_R)$, which grows linearly with the number of steps t . The typical distance of the walker from his starting point after t steps is roughly the standard deviation of the spatial probability distribution the walk generates, i.e., $\sigma_R = \sqrt{t p_R (1 - p_R)}$, and so scales as \sqrt{t} with increasing number of steps.

Given the tremendous versatility and success of random walks as a mathematical tool, a reasonable place to look for inspiration for new quantum algorithms would be to start with classical random walks and attempt to “quantumize” them. But how, exactly, are we to generalize the notion of a random walk to the quantum domain?

10.4.1 One-Dimensional Quantum Walks

In a classical random walk, once we have specified the initial position of the walker, and how the outcome of each coin toss is to decide in which direction to step next, we only need to keep track of two key pieces of information in order to determine an entire random walk, namely, the position of the walker after t steps, $x(t)$, and the state of the coin (“heads” or “tails”) after the t -th coin toss $c(t)$ say. Collectively, these determine the next position of the walker.

We therefore generalize the classical random walk to the quantum domain as follows: when we initialize the quantum walk, we start in a definite state for the position of the walker (which we can label as position $|x\rangle = |0\rangle$) and a definite state for the coin. This might be “tails”, which we can represent as $|c\rangle = |0\rangle$, or “heads”, which we can represent as $|c\rangle = |1\rangle$. Alternatively, a quantum coin can also start off in a superposition of both “heads” and “tails”, i.e., $|c\rangle = \alpha|0\rangle + \beta|1\rangle$. Hence, we can represent the initial state of the quantum walk as the state $|\psi(0)\rangle = |x\rangle|c\rangle$, where $|x\rangle$ represents the position state and $|c\rangle$ represents the coin state.

Now let us generalize the notions of the coin toss and the step operations. When we toss a coin classically, we cause it to spin in an attempt to randomize its state upon being measured. The analogous operation quantumly can be thought of as applying a unitary operator to the state of the coin, which causes it to enter a state whose identity (heads or tails) is ambiguous until measured. We can mimic this

operation quantumly by representing the coin toss as the application of a unitary transformation to the current state of the coin, as follows:

$$\begin{aligned}\mathbf{C}|x, 0\rangle &\longrightarrow a|x, 0\rangle + b|x, 1\rangle \\ \mathbf{C}|x, 1\rangle &\longrightarrow c|x, 0\rangle + d|x, 1\rangle\end{aligned}\tag{10.21}$$

where $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is a 2×2 unitary matrix. Thus, if the coin starts off in state $|0\rangle$ (i.e., “tails” say), the coin toss puts the coin in a state whose outcome is $|0\rangle$ with probability $|a|^2$ and whose outcome is $|1\rangle$ (i.e., “heads”) with probability $|b|^2$. Conversely, if the coin starts off in state $|1\rangle$ (“heads”) then the coin toss puts it in a state whose outcome is $|0\rangle$ with probability $|c|^2$ and or $|1\rangle$ with probability $|d|^2$. However, in a quantum walk *we do not read the coin after each “toss”*. Instead, we merely apply the coin toss operator, and compute the next joint state of the walker’s position and coin, $|\psi(t)\rangle$. In general, $|\psi(t)\rangle$ becomes a superposition of different position-state/coin-state pairs. Note that the quantum coin toss operator does not have to be fair. Any, 1-qubit unitary transformation could be used. However, unitarity of the matrix defining the coin toss operator is important in order to conserve probabilities.

Similarly, we can generalize the notion of the step operation as moving the walker one step to the left or right depending on the state of the coin (“heads” or “tails”). Formally, the step operator only acts on the position of the walker (not the state of his coin) and can therefore be defined to act as follows:

$$\begin{aligned}\mathbf{S}|x, 0\rangle &\longrightarrow |x - 1, 0\rangle \\ \mathbf{S}|x, 1\rangle &\longrightarrow |x + 1, 1\rangle\end{aligned}\tag{10.22}$$

Notice that, unlike the classical case, the quantum walker is not allowed to read the coin. After each coin toss the quantum walker therefore takes a step to the left *and* a step to the right simultaneously! An intuitive consequence of this is that, indeed, the quantum walker typically spreads out faster than the classical walker.

To predict the spatial probability distribution of the walker after t steps, we calculate how the joint state of the walker’s position and coin evolve after t steps, namely:

$$|\psi(t)\rangle = (\mathbf{S} \cdot \mathbf{C})^t |x\rangle |c\rangle\tag{10.23}$$

where \mathbf{S} only acts on the degrees of freedom describing the position of the walker (and leaves the coin state alone), and \mathbf{C} only acts on the degrees of freedom describing the coin (and leaves the position alone). By following this recipe, and provided that we do not read the coin during the walk, the state of the walk after t steps can be determined.

Note that this process is entirely deterministic up until the point any measurements are made. This is the reason we describe this quantum process as a “quantum walk”, rather than a “quantum random walk”. Randomness would only come in when at the point we make a final measurement of the walker’s position at the end of the time period of interest, after the walk is over.

Typically, quantum interference effects cause a quantum walk to behave quite differently from a classical walk. In particular, by choosing the initial state of coin and the coin flip operator appropriately, quantum walks can be configured so as to diffuse outwards dramatically faster than classical walks. Specifically, whereas in a classical random walk, the classical walker will be typically \sqrt{t} steps from the starting point after t coin tosses, in a quantum walk the quantum walker will be typically t steps from the starting point.

Moreover, the shape of the probability distributions over spatial locations can be dramatically different, with the classical random walk showing a uni-modal probability distribution centered at the starting point, and the quantum walk showing a predominantly bi-modal probability distribution, that may even preclude certain positions, and may or may not be symmetric about the starting point. That is, even if a quantum walker is required at each coin toss to take *one* step left or right, interference effects can, in some cases, preclude the quantum walker from ever occupying certain positions over which it can, in principle, pass. That is, if there are three positions in a line $x_1 < x_2 < x_3$, it is possible for a quantum walker to walk from x_1 to x_3 without it ever being possible to find it at the intermediate position x_2 ! In the following sections we will illustrate these phenomena by way of simple one-dimensional examples. Generalizations of quantum walks to higher dimensions are possible.

10.4.2 Example: Biased Initial Coin State & Hadamard Coin

The first quantum walk we shall look at is the one-dimensional quantum walk on a line, in which the starting position, the initial coin state, the coin flip operator, and the step operator are defined as follows:

$$\begin{aligned}
 |x\rangle &= |0\rangle \\
 |c\rangle &= |0\rangle \\
 |x, 0\rangle &\xrightarrow{\mathbf{C}} \frac{1}{\sqrt{2}}(|x, 0\rangle + |x, 1\rangle) \\
 |x, 1\rangle &\xrightarrow{\mathbf{C}} \frac{1}{\sqrt{2}}(|x, 0\rangle - |x, 1\rangle) \\
 |x, 0\rangle &\xrightarrow{\mathbf{S}} |x - 1, 0\rangle \\
 |x, 1\rangle &\xrightarrow{\mathbf{S}} |x + 1, 1\rangle
 \end{aligned} \tag{10.24}$$

where \mathbf{C} performs a Hadamard transform on the coin state, and \mathbf{S} performs a shift transformation on the position state. Notice the initial state of the coin, i.e., $|c\rangle = |0\rangle$, is biased as it is wholly $|0\rangle$.

To compute the state of the walk after t steps, we compute $|\psi(t)\rangle = (\mathbf{S} \cdot \mathbf{C})^t |x\rangle |c\rangle$ using the aforementioned values. The first five states of this quantum walk are found

to be:

$$\begin{aligned}
 |\psi(0)\rangle &= |0, 0\rangle \\
 |\psi(1)\rangle &= \frac{1}{\sqrt{2}}(|-1, 0\rangle + |1, 1\rangle) \\
 |\psi(2)\rangle &= \frac{1}{2}(|-2, 0\rangle + |0, 0\rangle + |0, 1\rangle - |2, 1\rangle) \\
 |\psi(3)\rangle &= \frac{1}{2\sqrt{2}}(|-3, 0\rangle + 2|-1, 0\rangle + |-1, 1\rangle - |1, 0\rangle + |3, 1\rangle) \\
 |\psi(4)\rangle &= \frac{1}{4}(|-4, 0\rangle + 3|-2, 0\rangle + |-2, 1\rangle - |0, 0\rangle + |0, 1\rangle \\
 &\quad + |2, 0\rangle - |2, 1\rangle - |4, 1\rangle)
 \end{aligned} \tag{10.25}$$

Notice that certain spatial positions for the quantum walker are disallowed. The true difference between a classical walk and this quantum walk can be grasped by a more visual comparison. Figure 10.3 shows the spatial probability distribution of the classical walker (dotted) and the quantum walker (solid) after $t = 100$ steps. Notice the appearance of a pronounced leftward bias in the quantum walk compared to the symmetry of the classical walk. Moreover, notice the emergence of “ripples” in the spatial probability function in the quantum case that are absent in the classical case.

Note that we have only plotted the probability distribution of the walker at even-numbered locations since there is no possibility the walker will ever be found at an odd numbered location. Therefore, strictly speaking, we should not have connected the dots in this plot. However, we did so to better visualize the probability distribution of the quantum walker, especially the oscillatory nature of the distribution.

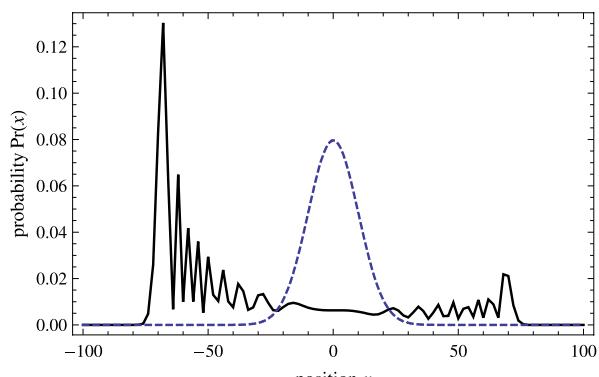


Fig. 10.3 Quantum walk using a starting state of $|0, 0\rangle$ and a Hadamard coin flip operator, $\mathbf{C} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. Notice that the walk is biased towards the left

10.4.3 Example: Symmetric Initial Coin State & Hadamard Coin

We might guess that the origin of the leftward bias of the quantum walk shown in Fig. 10.3 is due to having initialized the starting state of the coin to be purely “tails”, $|0\rangle$, rather than the equally weighted symmetric superposition of “heads” and “tails” ($\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$). So let us redo the quantum walk such that the starting position, the initial coin state, the coin flip operator, and the step operator are now defined as follows:

$$\begin{aligned}
 |x\rangle &= |0\rangle \\
 |c\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\
 |x, 0\rangle &\xrightarrow{\mathbf{C}} \frac{1}{\sqrt{2}}(|x, 0\rangle + |x, 1\rangle) \\
 |x, 1\rangle &\xrightarrow{\mathbf{C}} \frac{1}{\sqrt{2}}(|x, 0\rangle - |x, 1\rangle) \\
 |x, 0\rangle &\xrightarrow{\mathbf{S}} |x - 1, 0\rangle \\
 |x, 1\rangle &\xrightarrow{\mathbf{S}} |x + 1, 1\rangle
 \end{aligned} \tag{10.26}$$

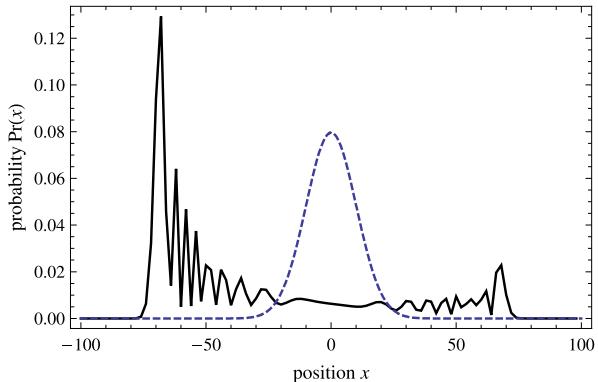
where as before \mathbf{C} performs a Hadamard transform on the coin state, and \mathbf{S} performs a shift transformation on the position state. Notice the initial state of the coin, $|c\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, is now unbiased as it is an equally weighted superposition of heads and tails. Does this removal of bias in the initial coin state make the subsequent quantum walk symmetric?

The first five states of this quantum walk are:

$$\begin{aligned}
 |\psi(0)\rangle &= \frac{1}{\sqrt{2}}(|0, 0\rangle + |0, 1\rangle) \\
 |\psi(1)\rangle &= |-1, 0\rangle \\
 |\psi(2)\rangle &= \frac{1}{\sqrt{2}}(|-2, 0\rangle + |0, 1\rangle) \\
 |\psi(3)\rangle &= \frac{1}{2}(|-3, 0\rangle + |-1, 0\rangle + |-1, 1\rangle - |1, 1\rangle) \\
 |\psi(4)\rangle &= \frac{1}{2\sqrt{2}}(|-4, 0\rangle + |-2, 0\rangle + |-2, 1\rangle - |0, 0\rangle + |2, 1\rangle)
 \end{aligned}$$

We can visualize what this walk looks like by carrying the pattern forward for $t = 100$ steps and then plotting the spatial probability distribution of the quantum walker at the end of that process. The result is shown in Fig. 10.4. We see that, perhaps surprisingly, the walk is again biased. How can we fix this?

Fig. 10.4 Quantum random walk using a starting state of $\frac{1}{\sqrt{2}}(|0, 0\rangle + |0, 1\rangle)$ and a Hadamard coin flip operator, $C = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. Notice that the walk is biased towards the left



10.4.4 Example: Chiral Initial Coin State & Hadamard Coin

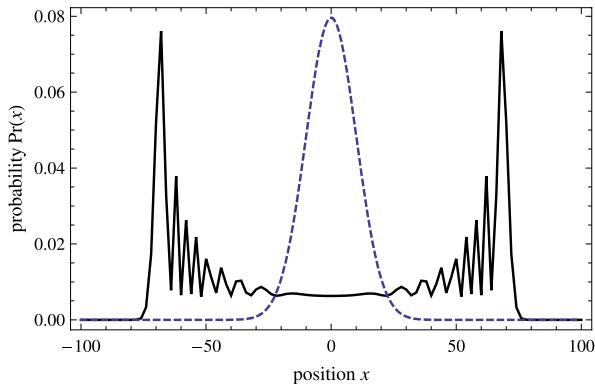
It turns out that there are two ways to fix the bias. One is to change the starting state of the coin to the “chiral” form $\frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$, while keeping the coin flip operator the same, and the other if to change the coin flip operator. For example, defining the initial coin state, the coin flip operator, and the step operator as follows:

$$\begin{aligned}
 |x\rangle &= |0\rangle \\
 |c\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \\
 |x, 0\rangle \xrightarrow{\mathbf{C}} &\frac{1}{\sqrt{2}}(|x, 0\rangle + |x, 1\rangle) \\
 |x, 1\rangle \xrightarrow{\mathbf{C}} &\frac{1}{\sqrt{2}}(|x, 0\rangle - |x, 1\rangle) \\
 |x, 0\rangle \xrightarrow{\mathbf{S}} &|x - 1, 0\rangle \\
 |x, 1\rangle \xrightarrow{\mathbf{S}} &|x + 1, 1\rangle
 \end{aligned} \tag{10.27}$$

we see that the initial coin state, i.e., $|c\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$, is now “chiral” but nevertheless still unbiased in terms of probability between heads and tails. The first five states of this quantum walk are:

$$\begin{aligned}
 |\psi(0)\rangle &= \frac{1}{\sqrt{2}}(|0, 0\rangle + i|0, 1\rangle) \\
 |\psi(1)\rangle &= \left(\frac{1}{2} + \frac{i}{2}\right)|-1, 0\rangle + \left(\frac{1}{2} - \frac{i}{2}\right)|1, 1\rangle \\
 |\psi(2)\rangle &= \frac{1}{\sqrt{2}}\left(\left(\frac{1}{2} + \frac{i}{2}\right)|-2, 0\rangle + \left(\frac{1}{2} - \frac{i}{2}\right)|0, 0\rangle\right. \\
 &\quad \left.+ \left(\frac{1}{2} + \frac{i}{2}\right)|0, 1\rangle - \left(\frac{1}{2} - \frac{i}{2}\right)|2, 1\rangle\right)
 \end{aligned}$$

Fig. 10.5 Quantum random walk using a starting state of $\frac{1}{\sqrt{2}}(|0, 0\rangle + i|0, 1\rangle)$ and a Hadamard coin flip operator, $C = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. In this case the quantum walk is symmetric



$$|\psi(3)\rangle = \left(\frac{1}{4} + \frac{i}{4}\right)|-3, 0\rangle + \frac{1}{2}|-1, 0\rangle + \left(\frac{1}{4} + \frac{i}{4}\right)|-1, 1\rangle \quad (10.28)$$

$$- \left(\frac{1}{4} - \frac{i}{4}\right)|1, 0\rangle - \frac{i}{2}|1, 1\rangle + \left(\frac{1}{4} - \frac{i}{4}\right)|3, 1\rangle$$

$$|\psi(4)\rangle = \frac{1}{\sqrt{2}}\left(\left(\frac{1}{4} + \frac{i}{4}\right)|-4, 0\rangle + \left(\frac{3}{4} + \frac{i}{4}\right)|-2, 0\rangle\right.$$

$$+ \left(\frac{1}{4} + \frac{i}{4}\right)|-2, 1\rangle - \left(\frac{1}{4} + \frac{i}{4}\right)|0, 0\rangle$$

$$+ \left(\frac{1}{4} - \frac{i}{4}\right)|0, 1\rangle + \left(\frac{1}{4} - \frac{i}{4}\right)|2, 0\rangle$$

$$- \left(\frac{1}{4} - \frac{3i}{4}\right)|2, 1\rangle - \left(\frac{1}{4} - \frac{i}{4}\right)|4, 1\rangle\left)$$

As before, we can visualize the spatial probability distribution induced by this quantum walk after $t = 100$ steps. The result is shown in Fig. 10.5. The sequence of states leading to this result can also be visualized by plotting the spread of the spatial probability distribution with increasing numbers of steps t as shown in Fig. 10.6.

Notice that the quantum walk is now symmetric about the starting location and the spatial probability distribution of the walker is predominantly bi-modal with several “ripples” at intermediate spatial locations. This is quite different from the central uni-modal hump seen in the analogous classical walk.

10.4.5 Example: Symmetric Initial Coin State & Non-Hadamard Coin

As we alluded to above, the other way to “fix” the asymmetry problem is to pick a symmetric initial state for the coin, where \mathbf{C} performs a non-Hadamard transform

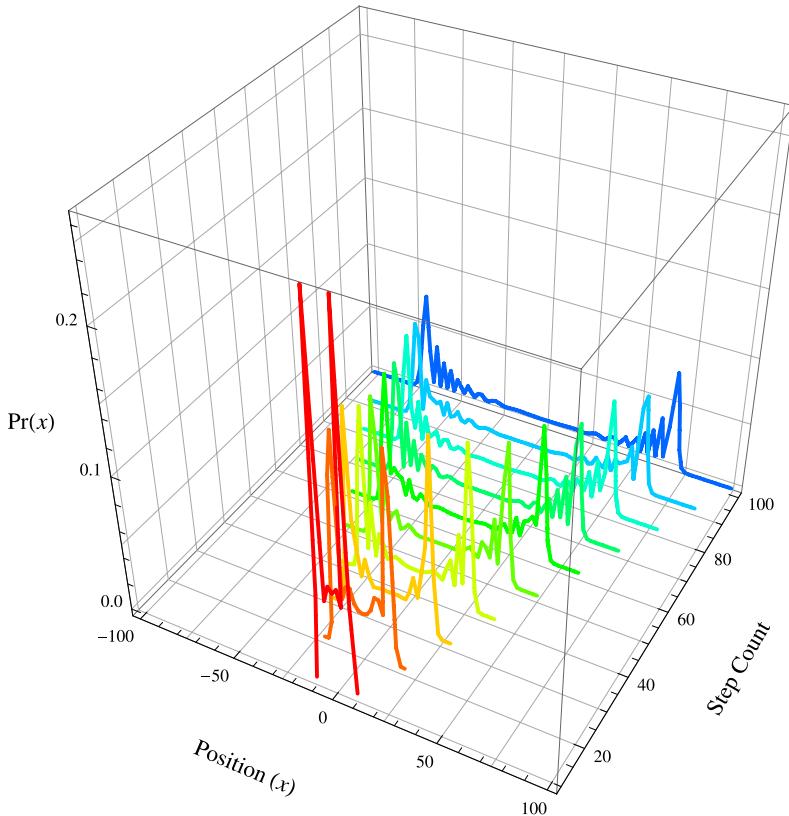


Fig. 10.6 The probability distribution of a quantum walk being at position x after (from front to back) $n = 10, 20, \dots, 100$ steps. The walk starts at $|x\rangle = |0\rangle$ and the coin is initialized to $\frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$. The coin flip operator is the Hadamard gate, i.e., $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

on the coin state, and \mathbf{S} performs a shift transformation on the position state. Notice the initial state of the coin, $|c\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and to change the coin flip operator to the more symmetric form $\mathbf{C} = \frac{1}{\sqrt{2}}\begin{pmatrix} i & 1 \\ 1 & i \end{pmatrix}$. Defining the initial coin state, the coin flip operator, and the step operator as follows:

$$\begin{aligned}
 |x\rangle &= |0\rangle \\
 |c\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\
 |x, 0\rangle &\xrightarrow{\mathbf{C}} \frac{1}{\sqrt{2}}(i|x, 0\rangle + |x, 1\rangle) \\
 |x, 1\rangle &\xrightarrow{\mathbf{C}} \frac{1}{\sqrt{2}}(|x, 0\rangle + i|x, 1\rangle)
 \end{aligned} \tag{10.29}$$

$$\begin{aligned}|x, 0\rangle &\xrightarrow{\mathbf{S}} |x - 1, 0\rangle \\|x, 1\rangle &\xrightarrow{\mathbf{S}} |x + 1, 1\rangle\end{aligned}$$

gives us the quantum walk whose first five states are as follows:

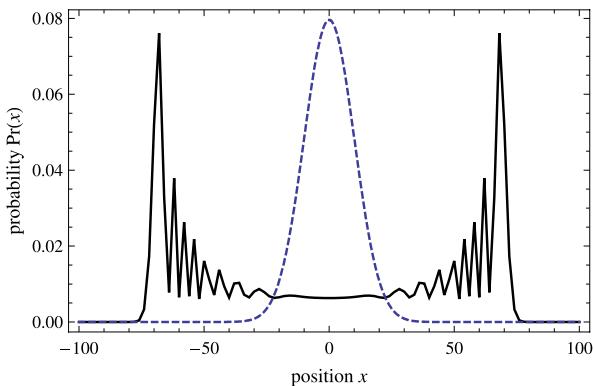
$$\begin{aligned}|\psi(0)\rangle &= \frac{1}{\sqrt{2}}(|0, 0\rangle + |0, 1\rangle) \\|\psi(1)\rangle &= \left(\frac{1}{2} + \frac{i}{2}\right)|-1, 0\rangle + \left(\frac{1}{2} + \frac{i}{2}\right)|1, 1\rangle \\|\psi(2)\rangle &= \frac{1}{\sqrt{2}}\left(\left(\frac{1}{2} + \frac{i}{2}\right)|-2, 0\rangle - \left(\frac{1}{2} - \frac{i}{2}\right)|0, 0\rangle\right. \\&\quad \left.- \left(\frac{1}{2} - \frac{i}{2}\right)|0, 1\rangle + \left(\frac{1}{2} + \frac{i}{2}\right)|2, 1\rangle\right) \\|\psi(3)\rangle &= \left(\frac{1}{4} + \frac{i}{4}\right)|-3, 0\rangle - \frac{1}{2}| -1, 0\rangle - \left(\frac{1}{4} - \frac{i}{4}\right)| -1, 1\rangle \\&\quad - \left(\frac{1}{4} - \frac{i}{4}\right)|1, 0\rangle - \frac{i}{2}|1, 1\rangle + \left(\frac{1}{4} + \frac{i}{4}\right)|3, 1\rangle \tag{10.30} \\|\psi(4)\rangle &= \frac{1}{\sqrt{2}}\left(\left(\frac{1}{4} + \frac{i}{4}\right)|-4, 0\rangle - \left(\frac{3}{4} + \frac{i}{4}\right)|-2, 0\rangle\right. \\&\quad - \left(\frac{1}{4} - \frac{i}{4}\right)|-2, 1\rangle - \left(\frac{1}{4} + \frac{i}{4}\right)|0, 0\rangle \\&\quad - \left(\frac{1}{4} + \frac{i}{4}\right)|0, 1\rangle - \left(\frac{1}{4} - \frac{i}{4}\right)|2, 0\rangle \\&\quad \left.- \left(\frac{3}{4} - \frac{i}{4}\right)|2, 1\rangle + \left(\frac{1}{4} + \frac{i}{4}\right)|4, 1\rangle\right)\end{aligned}$$

Continuing the pattern, we can visualize the spatial probability distribution induced by this quantum walk after t steps by computing $|\psi(t)\rangle = (\mathbf{S} \cdot \mathbf{C})^t |x\rangle |c\rangle$. The result is shown in Fig. 10.7. Again, we obtain a symmetric walk that appears to spread (or diffuse) faster than a similar classical walk. Let us next compare the rates of diffusion of these four quantum walks to each other and to a typical unbiased one-dimensional classical walk.

10.4.6 Quantum Walks Can Spread Faster than Classical Walks

As you can see, the choice of initial state for the coin, and the coin flip operator can affect the course of the quantum walk significantly. More importantly, though, the

Fig. 10.7 Quantum random walk using a starting state of $\frac{1}{\sqrt{2}}(|0,0\rangle + |0,1\rangle)$ and a coin flip operator, $C = \frac{1}{\sqrt{2}}\begin{pmatrix} i & 1 \\ 1 & i \end{pmatrix}$. Notice that the walk is symmetric.



long time behavior of a quantum walk is markedly different from the corresponding classical random walk, due to interference between the different possible paths. In particular, the shape of the probability distribution for finding the walker a certain distance from the starting point is dramatically different in the classical and quantum cases. The upshot is that quantum walkers find themselves further from their starting point on average than a classical walker, and this forms the basis of a quantum speed up that can be exploited to solve problems faster.

We can make the argument more precise by computing the variance of the quantum walk in space after t steps. We have:

$$\begin{aligned} \text{var}(x) &= \langle x^2 \rangle - \langle x \rangle^2 \\ &= \sum_{i=1}^t x^2 \Pr(x) - \left(\sum_{i=1}^t x \Pr(x) \right)^2 \end{aligned} \quad (10.31)$$

over the t steps taken. We can obtain $\Pr(x)$ from quantum walker's state after t steps, i.e., $|\psi(t)\rangle$, where the first index of $|\psi(t)\rangle$ defines the spatial coordinate of the walker.

In Fig. 10.8 we plot the variance in the spatial location of the quantum walker after t steps for up to $t = 50$ steps for the first three quantum walks outlined above. We then fitted polynomials to the data to obtain numerical estimates for the rate of growth of variance of the three quantum walks. In all cases, the rate of growth of the variance of the spatial probability distributions fits a quadratic form very well, with the fastest rate of increase in variance found for the symmetric quantum walk. As the rate of growth of variance of all the quantum walks fits a *quadratic* in t very well, this means that the typical distance of the quantum walker grows as t rather than \sqrt{t} for the closest analogous classical walk scenario. Hence, quantum walks can be configured to diffuse out (explore the space) much more rapidly than classical random walks. This phenomenon provides the basis on which many quantum-walk based algorithms can be devised that out-perform their classical counterparts.

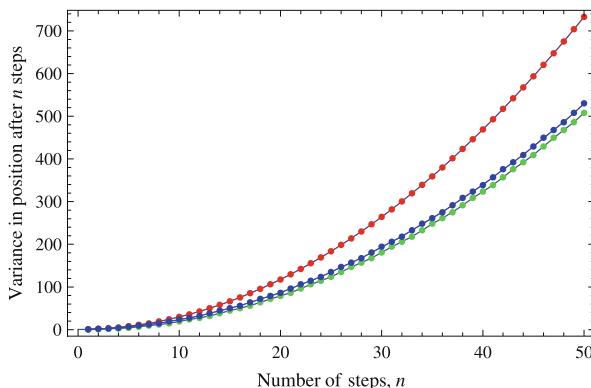


Fig. 10.8 Growth of variance of three quantum walks. All three walks start at the same position, i.e., $|x\rangle = |0\rangle$. The initial state of the coin is (a) $|0\rangle$ (for the blue curve), (b) $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ (for the green curve) and (c) $\frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$ (for the red curve). The probability distribution for the green and blue quantum walks are similar and show a high asymmetry in the walk, favoring a leftward drift. The probability distribution of the red quantum walk is highly symmetrical about the starting position. In all cases the growth in the variance is an excellent fit to a quadratic in the number of steps, n . Specifically, for quantum walk (a) the variance grows as $\text{var}(\Pr(t)) \approx 0.498 + 0.195t + 0.207t^2$. For quantum walk (b) the variance grows as $\text{var}(\Pr(t)) \approx 0.316 - 0.197t + 0.207t^2$. And for quantum walk (c) the variance grows as $\text{var}(\Pr(t)) \approx 0.582 - 0.004t + 0.293t^2$. Contrast these results to a classical random walk whose variance typically grows in proportion to t (the step count) rather than t^2 . This means that quantum walks can be configured so as to diffuse out faster than classical walks

10.5 Summary

Since the discovery of Shor's algorithm, people have been looking for quantum algorithms that can solve other mathematical problems apart from factoring composite integers and computing discrete logarithms. The result has been quantum algorithms for addressing many areas of mathematics, including, computing numerical integrals, determining eigensolutions of differential operators, solving Diophantine equations, computing properties of graphs. In addition, some useful mathematical techniques such as random walks and game theory have been generalized to the quantum domain and new versions of famous theorems related to the rate at which random walkers diffuse out and the equilibria of games have been found.

In this chapter we gave but a taste of these developments. The quantum counting algorithm is especially noteworthy since it combines ideas from both Grover's algorithm and phase estimation. Moreover, quantum counting is practically useful as it can be used as a preliminary step in a quantum search when the number of solutions to the search problem is not known a priori. Hallgren's algorithm for solving Pell's equation is noteworthy because it represents a significant extension of Shor's algorithm to the case of periodic functions having an irrational period. Quantum random walks are noteworthy because they have stimulated a great many new ideas for quantum algorithms and have also contributed to new physics understanding of

transport phenomena in materials. Quantum algorithms that work on times series data and 2D images are possible, but these require that the data be encoded in a quantum state prior to the application of the quantum algorithm. This generally requires a computational cost that is proportional to the size of the data. Nevertheless, this could still be advantageous if subsequent quantum processing is exponentially faster than classical alternatives.

10.6 Exercises

10.1 In the text we defined a quantum random walk in terms an operator that acts on $|x, c\rangle$, i.e., the instantaneous position of the walker, x , and the instantaneous state of his or her coin c . Consider the following as an alternative strategy: let the position of the quantum walker be x . At each time step, and for any position x , we imagine there is an amplitude for the walker to move left, an amplitude to stay in place, and an amplitude to move right. Thus, the transform we require, U , would perform the following mapping: $U|x\rangle \longrightarrow a_{\text{LEFT}}|x-1\rangle + a_{\text{STAY}}|x\rangle + a_{\text{RIGHT}}|x+1\rangle$. Then U^t would give the state describing the position of the quantum walker after t steps. Explain why this is *not* a suitable specification for a quantum random walk. [Hint: is the given random walk operator, U , physically allowed?]

10.2 Consider the quantum random walk starting in state $|0, 0\rangle$ and using the Hadamard coin flip operator, $C = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. Prove that the quantum walker can *never* be found at an odd-numbered location.

10.3 In Grover's algorithm with a single solution, or "good" state $|t\rangle$, we write the operator $\mathbb{1}_t = 1 - 2|t\rangle\langle t|$. When there are multiple solutions it is natural to write the superposition of good indices as $|\psi_{\text{good}}\rangle = \sum_{j \in X_{\text{good}}} |j\rangle$. Can we write the corresponding operator as $\mathbb{1}_{\text{good}} = 1 - 2|\psi_{\text{good}}\rangle\langle\psi_{\text{good}}|$?

10.4 In this chapter we described an algorithm for synthesizing an arbitrary *pure* state. We can build upon this technique to devise a method for synthesizing an arbitrary *mixed* state. The scheme can best be understood by inspecting the structure of the quantum circuit, show in Fig. 10.9, used to implement it. The mixed state synthesis algorithm works as follows:

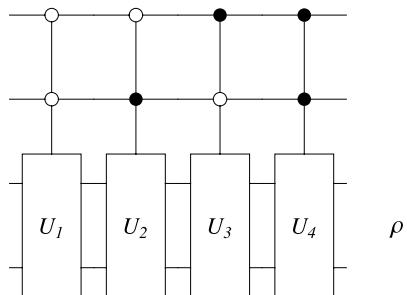
Algorithm `SynthesizeMixedState` A quantum circuit sufficient to synthesize an arbitrary $2^n \times 2^n$ dimensional density operator ρ can be determined as follows:

1. Compute the spectral decomposition of $\rho = \sum_{i=1}^k p_i |i\rangle\langle i|$. This reveals a set of quantum states $\{|i\rangle\}$.
2. Compute the unitary matrices U_i such that $|i\rangle = U_i|0\rangle$.
3. Compute a circuit for performing $U = U_1 \oplus U_2 \oplus \dots \oplus U_k$.

Fig. 10.9 Quantum circuit for synthesizing an arbitrary mixed state, ρ

$$|\varphi\rangle = \sum_{i=1}^k \sqrt{p_i} |i\rangle$$

$|00\dots 0\rangle$



4. Compute a circuit for preparing the “loaded dice” state $|\varphi\rangle = \sum_{i=1}^k \sqrt{p_i} |i\rangle$.
5. Compute the input state $|\psi\rangle = |\varphi\rangle \otimes |\underbrace{00\dots 0}_{n \text{ qubits}}\rangle$.
6. Push this state through U , and trace over the control qubits, C , i.e., perform the computation $\text{tr}_C(U|\psi\rangle\langle\psi|U^\dagger) = \rho$.

Apply the aforementioned `SynthesizeMixedState` algorithm to find a quantum circuit to synthesize the maximally-entangled mixed state, ρ , where:

$$\rho = \begin{pmatrix} \frac{1}{3} & 0 & 0 & \frac{1}{6} \\ 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{6} & 0 & 0 & \frac{1}{3} \end{pmatrix} \quad (10.32)$$

Part III

**What Can You Do with Quantum
Information?**

Chapter 11

Quantum Information

“One could caricature quantum information processing as the science of turning quantum conundrums into potentially useful applications.”

– Nicolas Gisin¹

Classical information theory, invented by Claude Shannon in 1948, addresses two main issues: the degree to which a classical message (i.e., a sequence of symbols) can be compressed, and the maximum rate at which reliable communications can be sustained over a noisy communications channel. The quantitative statement regarding the maximum compressibility of a symbol sequence is enshrined in Shannon’s “Noiseless Source Coding Theorem”, and the quantitative statement regarding the maximum rate of reliable communications, for a given noise level in the channel, is enshrined in Shannon’s “Noisy Channel Coding Theorem”. Together, these theorems laid the foundations for several multi-billion dollar industries such as telecommunications, cellular phone networks, internet, and disk drives. In fact, we make use of information theory everyday but barely give it any thought whatsoever.

Since information theory was invented, engineers have refined communications and data storage devices constantly so that they use fewer physical resources to encode more information. This has enabled dramatic increases in the storage capacity of computer memories, significant reductions in the power consumption of communications devices, and large increases in the rate at which information can be exchanged. Indeed, codes are now known that operate surprisingly close to the limits implied by Shannon’s theorems.

In this chapter we consider how information theory needs to be modified once we use the quantum states of simple systems (such as photons) to encode symbols. We might expect that some modification is necessary because, e.g., whereas symbols encoded in the states of classical physical systems are guaranteed to be distinguishable, the same cannot be said for symbols encoded in the states of quantum systems (e.g., if they are non-orthogonal). But, in fact, the reasons for modification runs much deeper than this: Some elementary information processing operations, such

¹Source: in “Quantum Cryptography” Reviews of Modern Physics, Volume 74, January (2002).

as copying data, which are permitted on classical information are impossible when attempted on quantum information. Conversely, other operations, such as teleportation, which are impossible when using classical information, can be achieved using quantum information.

As in the case of computer science, this shift in the foundations of the field turns out to have profound consequences. In particular, it leads to new (quantum) versions of both the noiseless coding theorem and the noisy channel coding theorem. As you shall see, quantum information theory forces us to revise our most cherished assumptions regarding how information should behave.

11.1 What is Classical Information?

“It might even be fair to observe that the concept that information is fundamental is very old knowledge of humanity, witness for example the beginning of the gospel according to John: “In the beginning was the Word” ”

– Anton Zeilinger²

Most people have an intuitive understanding of what they mean by “information”. It’s the stuff they read in newspapers, copy off blackboards, or absorb while watching CNN etc. However, when pressed to give a more precise definition, I find that most people equate “information” with the knowledge imparted during some communicative act, i.e., what they know now that they didn’t know before. This implicitly connects “information” with the meaning of a communication, i.e., its qualitative aspects.

A problem with this position, is that it makes the “information” contained within a message highly subjective, hard to quantify, and context dependent. For example, the “information” two people may attach to a CNN report would then depend on what they knew beforehand. It is tricky to make any mathematical headway with such a subjective basis for a notion of “information”. So the commonsense view of “information” as the knowledge imparted during some communicative act is not very useful in a practical sense.

In 1948 Claude Shannon hit upon an alternative view of what we should mean by “information”. He suggested the information within a message was simply the minimum number of 0s and 1s needed to transmit it. In Shannon’s own words [14]:

“The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one selected from a set of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.”

²Source: [560].

Shannon's insight was as ingenious as it was dehumanizing! By equating “information” with the minimal resources needed to *represent* a message, rather than its *knowledge content* per se, it became possible to derive laws describing how the amount of information would change under various operations, such as compressing messages or sending them through noisy communications channels. In turn, such understanding led to breakthroughs in data compression, encryption, and telecommunications.

Yet the cost is severe. Shannon's perspective strips all humanity from the notion of information. In Shannon's theory a love letter might have the same information content as a bus schedule, since his notion of information only addresses its quantitative aspects not its qualitative aspects. “Information” became something sterile, lifeless, and devoid of passion or creativity. Nevertheless, the operational utility of regarding information as the minimum number of 0s and 1s needed to encode some message is currently the best handle we have on quantifying the elusive and abstract notion of “information”.

11.1.1 Classical Sources: The Shannon Entropy

We can think of a source of classical information as a device that produces a stream of classical symbols, such as lowercase letters, uppercase letters, numbers, and punctuation marks. After large numbers of such symbols have been produced we can determine their probability distribution. In principle, all sorts of subtle correlations amongst the symbols are possible. For example, in English the symbol “*q*” is followed, invariably, by the symbol “*u*” as in the words such as “quantum”, “quest”, “quibble”, and “quoff”. Nevertheless, as each distinct symbol can be encoded as a corresponding binary string, we can equally think of a source of classical information as a device that produces sequences of *bits*, i.e., 0s and 1s. Consequently, correlations amongst the symbols would then appear as correlations amongst *subsequences* of bits. However, correlations at the level of *individual* bits would tend to be diluted out.

How one sets up the mapping between symbols and bit strings makes a difference. For example, the frequencies with which different letters arise in written English are different (see Fig. 11.1) with “*e*” being the most common letter. Similarly, one could treat whole words as “symbols” and plot *their* frequency of occurrence too. Such statistical insights into the structure of natural languages have permitted modern marvels such as smarter internet search engines (which exploit word correlations to infer context and relevance) and statistical machine translation tools (which can teach themselves to translate documents by being “trained” to infer the mathematical correlations between the words and phrases found in matching pairs of human-translations of large corpora of documents). When one makes the sensible choice of using shorter bit strings to encode more frequent symbols in a language, one finds that although we can model a source of the language as a stream of independent, identically distributed, bits in which 0 occurs with probability $p_0 = p$ and

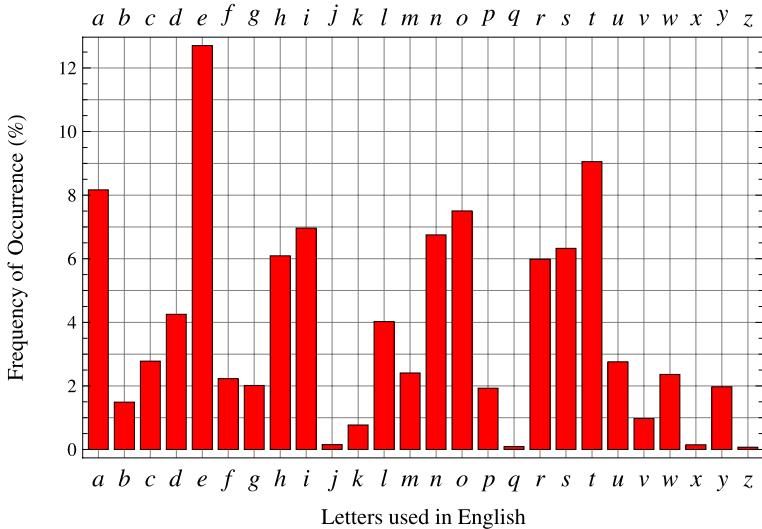


Fig. 11.1 Letter frequency distribution in English

1 occurs with probability $p_1 = 1 - p$, that for real languages there is an asymmetry between p_0 and p_1 . Ultimately, this asymmetry is what allows us to compress messages.

Specifically, if 0 occurs with probability $p_0 = p$ and 1 occurs with probability $p_1 = 1 - p$, a “typical” n -bit message will have roughly np 0s and $n(1 - p)$ 1s. Hence, the number of “typical” bit strings is therefore:

$$\binom{n}{np} = \frac{n!}{(np)!(n-np)!} \quad (11.1)$$

Using Stirling’s formula $N! \approx N^N e^{-N} \sqrt{2\pi N}$ for $N \gg 1$ we have $\log_e N! \approx N \log_e N - N$ and so:

$$\begin{aligned} \log_e \binom{n}{np} &\approx n \log_e n - n - (np \log_e np - np \\ &\quad + (n - np) \log_e (n - np) - (n - np)) \\ &= n(-p \log_e p - (1 - p) \log_e (1 - p)) \\ &= \frac{1}{\log_2 e} n(-p \log_2 p - (1 - p) \log_2(1 - p)) \\ &\approx nH(\{p, 1 - p\}) \end{aligned} \quad (11.2)$$

where $\{p_i\}$ is the set of positive real numbers defining the *probability* with which each possible symbol appears in the message. In the case of bit string messages there are only two symbols, $i = 0$ and $i = 1$, and so the probabilities are simply p_0 and

$p_1 = 1 - p_0$. The function $H(\{p_0, p_1\}) = -\sum_{i=0}^1 p_i \log_2 p_i$ is called the *Shannon entropy*. For symbols that are just single bits we have $H(\{p_i\}) \equiv H(\{p_0, p_1\}) \equiv H(\{p, 1-p\})$.

The choice of which base to use for the logarithm is somewhat arbitrary as different choices only serve to re-scale the measure of information (or entropy) by a constant factor. If we choose to use base 2, our scale has a certain natural feel to it. Using base 2 logarithms, if $p_0 = p_1 = \frac{1}{2}$, an n -bit classical message would be completely random (and hence incompressible) and would convey exactly $nH(\{\frac{1}{2}, \frac{1}{2}\}) = n$ bits of information. At the other extreme, a string of n identical bits, such as n 0s (and hence devoid of any useful information), would convey $nH(\{1, 0\}) = 0$ bits of information. So by choosing base 2, we arrive at a fairly intuitive scale for information.

11.1.2 Maximal Compression (Source Coding Theorem)

“Source coding” refers to the data compression problem. That is, given a source producing a sequence of symbols in accordance with some a priori probability distribution, by what factor can we compress a typical message from this source without corrupting it? If no information whatsoever is lost, the compression is said to be “lossless”. But in many cases we are content with a “lossy” compression provided the losses do not rise to a level we perceive as significant.

We can approach this question with the help of Shannon information theory. Suppose we model the source as emitting a sequence of independent, identically distributed, bits in which 0 occurs with probability p_0 and 1 occurs with probability $p_1 = 1 - p_0$. Then most n -bit messages generated by such a source will be close to the “typical” messages. That is, they will have close to np_0 0’s and $n(1 - p_0)$ 1’s. Therefore, we need only worry about how sending “typical” messages. So rather than there being $\mathcal{O}(2^n)$ messages to worry about, we only really need to figure out how to handle $\mathcal{O}(2^{nH(\{p_0, p_1\})})$ typical messages. All we need to do is to assign a unique positive integer to each typical message, and send that integer, which requires only $nH(\{p_0, p_1\})$ bits, rather than the message, which requires n bits. As $n \rightarrow \infty$ almost all message will be close to typical. For example, if $p_0 = 0.3$ and $p_1 = 0.7$, then a “typical” 20-bit message would have six 0’s and fourteen 1’s, and instead of there being $2^{20} \approx 1,000,000$ possible messages to send there would be only $2^{nH(\{0.3, 0.7\})} \approx 200,000$ typical messages to send.

The notion of the entropy of a source that emits one of two possible symbols, i.e., a binary source, can be generalized readily to one that emits one of d possible symbols, x_1, x_2, \dots, x_d . Assuming symbol x_i appears with probability p_i , a typical message of length $n \gg 1$ symbols from such a source will have roughly np_1 occurrences of x_1 , np_2 occurrences of x_2 , etc. Hence the number of such typical messages is given by the number of ways np_1 x_1 ’s, np_2 x_2 ’s, etc. can be placed within a string of length n symbols, which is just the multinomial formula:

$$\frac{n!}{\prod_{i=1}^d (np_i)!} \quad (11.3)$$

such that $0 \leq p_i \leq 1$ and $\sum_{i=1}^d p_i = 1$. We can write this approximately as an exponential function of a modified entropy function

$$\frac{n!}{\prod_i (np_i)!} \approx 2^{nH(\{p_1, p_2, \dots, p_d\})} \quad (11.4)$$

if we define

$$H(\{p_1, p_2, \dots, p_d\}) = - \sum_{i=1}^d p_i \log_2 p_i \quad (11.5)$$

Such a generalization to the case of alphabets having d -symbols gives the Source Coding Theorem:

Source Coding Theorem *If n independent, identically distributed, random variables taken from a finite d -symbol alphabet each with entropy $H(\{p_1, p_2, \dots, p_d\})$ are compressed into no fewer than $nH(\{p_1, p_2, \dots, p_d\})$ bits then there is negligible risk of information loss, but compression beyond this limit makes some loss almost certain.*

For natural languages this notion of source coding is appropriate. But in other fields, e.g., mathematics and computer science, strings of letters and symbols arise that although outwardly complex if viewed as a symbol sequence, are actually much simpler if one understands the underlying generator. In such cases algorithmic information theory is a better tool for understanding their compressibility. In particular, Kolmogorov complexity is the shortest program needed to reproduce some sequence. So the Kolmogorov complexity of a truly random sequence is the sequence itself as a random sequence is, by definition, incompressible. In contrast, the sequence of (say) Fibonacci numbers, in which each successive number is the sum of the last two numbers, i.e., $1, 1, 2, 3, 5, 8, 13, 21, 34, \dots$ can be described more compactly via the recursive formula $f(n) : f(n) = f(n-1) + f(n-2)$ for $n \geq 3 \wedge f(1) = f(2) = 1$. This is dramatically shorter than writing out the sequence itself.

11.1.3 Reliable Transmission (Channel Coding Theorem)

Besides compression, another aspect of information theory is to ask how *reliably* information may be *conveyed* over a noisy communications channel. A typical communications channel adds noise to any signal sent through it causing errors in the data received. Attempts to correct such errors are prone to errors themselves. It is not obvious a priori, therefore, that a noisy communications channel *can* be used to transmit messages without error. Remarkably, in 1948 Claude Shannon proved a theorem that showed, regardless of how noisy a given channel may be, that it is always possible to communicate information over such a channel almost error

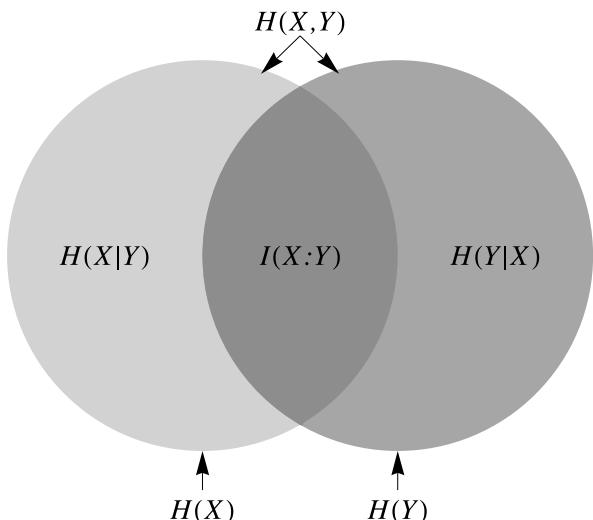
free up to a certain maximum rate set by the Channel Coding theorem. The method for doing so relies upon the use of error correcting codes, but the Channel Coding theorem does not tell us how to find these good codes, only that they exist. Nevertheless, since the advent of the Channel Coding theorem many excellent codes have been discovered, driven in large part by the needs of deep Space communications for supporting reliable communications during NASA Space missions. In particular, Turbo Codes, and Low Density Parity-Check Codes now come close to saturating the limit set by Shannon's Channel Coding theorem.

To state the theorem quantitatively we need a few key ideas. First the notion of a discrete channel is one consisting of an input alphabet \mathcal{X} and an output alphabet \mathcal{Y} and a probability transition matrix $p(Y|X)$, which specifies the probability of receiving symbol $Y \in \mathcal{Y}$ given that symbol $X \in \mathcal{X}$ was sent. When this probability distribution only depends on the last input to the channel, the channel is said to be "memoryless". We can also define the marginal probabilities of seeing the different symbols as $p(x = X) = \sum_y p(x, y)$ and $p(y = Y) = \sum_x p(x, y)$, where $p(x, y)$ is the joint probability of seeing $x = X$ and $y = Y$. From these we construct the mutual information $I(X : Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$, which is a measure of how much the two variables depend on each other. Then the channel capacity, C , of a discrete memoryless channel, can be defined to be the mutual information maximized over all probability distributions, i.e.,

$$C = \max_{p(X)} I(X : Y) \quad (11.6)$$

The relationship between entropy, conditional entropy, joint entropy, and mutual information is shown in Fig. 11.2. Formally, the Channel Coding theorem then establishes the maximum rate at which reliable communications can be supported given the characteristics of the channel.

Fig. 11.2 Graphical illustration of the relationship between entropy ($H(X)$, $H(Y)$), conditional entropy ($H(X|Y)$ and $H(Y|X)$), joint entropy ($H(X, Y)$) and mutual information ($I(X : Y)$). Formally we have $I(X : Y) = H(X) + H(Y) - H(X|Y)$ or, equivalently, $I(X : Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$. Furthermore, $I(X : Y) = I(Y : X)$ and $I(X : X) = H(X)$. Mutual information is a way to quantify the degree to which two variables depend on each other



Channel Coding Theorem *For any upper bound on the acceptable block error rate, $\epsilon > 0$, and for any rate $R < C$ (where $C = \max_{p(X)} I(X; Y)$ is the channel capacity), there is an encoding/decoding protocol that can guarantee that the probability of block error is less than ϵ for a sufficiently long code. Moreover, for any rate $R > C$, i.e., if the communications rate attempted exceeds the channel capacity, errors are inevitable.*

A proof of the Channel Coding theorem is given in Chapter 8 of Cover & Thomas's "Elements of Information Theory" [117].

11.1.4 Unstated Assumptions Regarding Classical Information

"Information is physical."

– Rolf Landauer

Just as the inventors of classical computer science had attempted to construct a theory of computation that was independent of how computers were implemented, so too did Shannon attempt to construct a theory of information that was supposed to be independent of how symbols were implemented. By building information theory on such a mathematical ideal, Shannon was able to make heroic advances in modeling data compression and communications channels and hence designing superior telecommunications systems. However, accepting this mathematical ideal as reality, causes people to assume (implicitly perhaps) that information has certain eminently reasonable properties. Back in 1948 and for most of the time since then, these assumptions have in fact been so obvious that no-one has ever really questioned them—until now. For example, thinking of information as the mathematical ideal of a stream of symbols invites the following presumptions:

- Information consists of a stream of *distinguishable* symbols
- Information can be compressed to no more than the Shannon bound
- Information does not change upon being read
- Information can be read in part without it affecting the unread parts
- Information can be copied exactly deterministically
- Information can be negated trivially by flipping every bit value

Indeed, the remarkable advances in communications systems since 1948 bear witness to how effective Shannon's theory has been, and how solidly these assumptions have been upheld.

Yet when we reduce the scale of the systems encoding information to individual quantum systems, then the nature of information itself begins to change. Under the right circumstances *every one of the aforementioned plausible statements about information can be made false*. The fundamental reason for this, as Richard Feynman put it, is that "Nature isn't classical dammit!". Indeed it is not. Our preconceptions of the properties that information should possess are intimately tied to the (more

implicit) assumptions for how such information is implemented. Just as computation should be seen as a *physical* process that depends in an essential way on the physical systems being used to enact computations, so it is for quantum information systems too.

11.2 What is Quantum Information?

The concept of quantum information is derived quite readily from that of classical information. Whereas classical information is a sequence of bits quantum information is a sequence of qubits. Entirely new types of phenomena are possible with quantum information that have no counterparts in classical information. For example, the successive qubits in a quantum message need not, and generally are not, orthogonal to one another, nor are they necessarily unentangled from one another. Thus a typical quantum memory register holds within it *quantum* information rather than classical information. As such it will typically hold information in an entangled superposition state, and the strengths of the correlations between bit values can exceed that which is possible classically.

11.2.1 Pure States cf. Mixed States

So far we have been mostly concerned with situations in which we have *complete* knowledge of the state of some n -qubit quantum memory register. That is, there is no uncertainty whatsoever regarding its state. It exists in some superposition of the possible bit string configurations of n bits, weighted by various amplitudes corresponding (via their modulus squared) to the probabilities of obtaining that particular bit string configuration if the memory register were to be read in the computational basis. In other words, the n -qubit register is in a state of the form:

$$|\psi\rangle = c_0|00\dots0\rangle + c_1|00\dots1\rangle + \dots + c_{2^n-1}|11\dots1\rangle \quad (11.7)$$

such that $\sum_{i=0}^{2^n-1} |c_i|^2 = 1$. Such a quantum state is said to be a *pure* state.

There are, however, situations in which we have only *incomplete* knowledge about some quantum state. Such states are called mixed states, as they correspond to weighted mixtures of different pure states.

11.2.2 Mixed States from Partial Knowledge: The Density Operator

One way mixed states can arise is when we only have probabilistic knowledge regarding the composition of a quantum state. Suppose, for example, that we

only known that a quantum system is in one of the (not necessarily orthogonal) states $|\psi_1\rangle, |\psi_2\rangle, \dots, |\psi_N\rangle$ with probabilities p_1, p_2, \dots, p_N respectively such that $\sum_{i=1}^N p_i = 1$. We are therefore a little uncertain of what the state actually is. How are we to characterize the quantum state of such a system?

One way we might learn something about the state is to make some sort of measurement on it. If we performed a measurement, described by the observable \mathcal{O} , on this system, the result we would expect to obtain would be the weighted average of the results we would obtain if the system was in each of the states $|\psi_1\rangle, |\psi_2\rangle, \dots, |\psi_N\rangle$, namely:

$$\langle \mathcal{O} \rangle = \sum_{i=1}^N p_i \langle \psi_i | \mathcal{O} | \psi_i \rangle \quad (11.8)$$

which after some manipulations (see problem 11.12) can be re-written as:

$$\sum_{i=1}^N p_i \langle \psi_i | \mathcal{O} | \psi_i \rangle = \text{tr}\left(\left(\sum_{i=1}^N p_i |\psi_i\rangle\langle\psi_i|\right) \cdot \mathcal{O}\right) = \text{tr}(\rho \cdot \mathcal{O}) \quad (11.9)$$

where “ $\text{tr}(\cdot)$ ” is the sum of the diagonal elements (i.e. the “trace”) of its argument (which is a matrix), and $\rho = \sum_{i=1}^N p_i |\psi_i\rangle\langle\psi_i|$ (which is also a matrix). Notice that ρ contains information only about the statistical mixture of pure states that contribute to the state, and \mathcal{O} contains information only about the observable being measured. Hence, ρ must be a complete characterization of the mixed state.

Density Operator If a quantum system exists in the state $|\psi_1\rangle$ with probability p_1 , $|\psi_2\rangle$ with probability $p_2, \dots, |\psi_N\rangle$ with probability p_N , where in general $\langle\psi_i|\psi_j\rangle \neq 1$ for $i \neq j$, then the best description of its state is given by the density operator:

$$\rho = \sum_{i=1}^N p_i |\psi_i\rangle\langle\psi_i| \quad (11.10)$$

11.2.2.1 Density Operator for a Mixed State

Although you can use density operators to describe pure states, the main motivation for introducing them is to be able to represent *mixed* states, i.e., statistical mixtures of pure states. This allows us to model circumstances in which we only have partial knowledge regarding the state. Specifically, if a quantum system exists in the state $|\psi_1\rangle$ with probability p_1 , $|\psi_2\rangle$ with probability $p_2, \dots, |\psi_N\rangle$ with probability p_N , where in general $\langle\psi_i|\psi_j\rangle \neq 1$ for $i \neq j$, then the best description of its state is given by the density operator:

$$\rho = \sum_{i=1}^N p_i |\psi_i\rangle\langle\psi_i| \quad (11.11)$$

where $\sum_{i=1} p_N = 1$. Here the component states need not be orthogonal with respect to one another, i.e., in general $\langle \psi_i | \psi_j \rangle \neq 0$ for $i \neq j$.

Many people are puzzled about the distinction between a mixed state and a superposition state, so it is worth stating this explicitly. A superposition state is a completely known pure state consisting of a weighted sum of eigenstates, $|\psi\rangle = \sum_i c_i |i\rangle$, which are all orthogonal with respect to one another, i.e., $\langle i | j \rangle = 0$ for all $i \neq j$. In principle, given knowledge of a superposition state, $|\psi\rangle$, one could build a measuring device that always yielded the same predictable result each time you used it to measure state $|\psi\rangle$. For example, if we had a single qubit in the superposition state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ we could rotate a measuring device that measures in the $\{|0\rangle, |1\rangle\}$ basis by 45° and then it would be measuring in the $\{|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\}$ basis, and always yield the result $|+\rangle$.

In contrast, a mixed state $\rho = \sum_j p_j |\phi_j\rangle\langle\phi_j|$ is an incompletely known state in which the component pure states (described by density operators $|\phi_j\rangle\langle\phi_j|$) need not be, and generally are not, orthogonal to one another. The fact that the state is incompletely known means that you can never be sure whether you really are dealing with a $|\phi_1\rangle$, or a $|\phi_2\rangle$, etc. Consequently, even if you know ρ , you cannot pick a measurement basis for a mixed state that is always guaranteed to yield the same predictable outcome.

The following example illustrates how to calculate the density operator of a mixed state that is a combination of three non-orthogonal pure states, $|\psi_1\rangle$, $|\psi_2\rangle$, and $|\psi_3\rangle$, with probabilities $p_1 = \frac{1}{3}$, $p_2 = \frac{2}{5}$ and $p_3 = \frac{4}{15}$ respectively where:

$$|\psi_1\rangle = |0\rangle \quad (11.12)$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (11.13)$$

$$|\psi_3\rangle = \frac{1}{2}|0\rangle + i\frac{\sqrt{3}}{2}|1\rangle \quad (11.14)$$

The corresponding density operator is:

$$\begin{aligned} \rho &= p_1|\psi_1\rangle\langle\psi_1| + p_2|\psi_2\rangle\langle\psi_2| + p_3|\psi_3\rangle\langle\psi_3| \\ &= \frac{1}{3}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \frac{2}{5}\begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix} + \frac{4}{15}\begin{pmatrix} \frac{1}{4} & -i\frac{\sqrt{3}}{4} \\ i\frac{\sqrt{3}}{4} & \frac{3}{4} \end{pmatrix} \\ &= \begin{pmatrix} \frac{3}{5} & -\frac{1}{5} - i\frac{1}{5\sqrt{3}} \\ -\frac{1}{5} + i\frac{1}{5\sqrt{3}} & \frac{2}{5} \end{pmatrix} \end{aligned} \quad (11.15)$$

Note that $\text{tr}(\rho) = 1$ (as for a pure state), but since

$$\rho^2 = \rho \cdot \rho = \begin{pmatrix} \frac{31}{75} & -\frac{1}{5} - i\frac{1}{5\sqrt{3}} \\ -\frac{1}{5} + i\frac{1}{5\sqrt{3}} & \frac{16}{75} \end{pmatrix},$$

$\text{tr}(\rho^2) = \frac{47}{75} < 1$. Seeing $\text{tr}(\rho^2) < 1$ is sufficient to conclude that ρ is a mixed state. This criterion holds true whatever of the dimensions of ρ .

11.2.2.2 Density Operator for a Pure State

Although we don't have to, we can certainly express a pure state in terms of its density operator. As the state is pure we have complete knowledge about it. Hence the ensemble contains exactly one kind of state, namely $|\psi\rangle$, and so the probability of this state being in the ensemble is 1 and all others are 0. Hence the density operator corresponding to pure state $|\psi\rangle = a|0\rangle + b|1\rangle$ is:

$$\rho_{\text{pure}} = |\psi\rangle\langle\psi| \quad (11.16)$$

with no summation. By expanding out the implied bras and kets, we can compute the density matrix explicitly as:

$$\rho_{\text{pure}} = |\psi\rangle\langle\psi| = \begin{pmatrix} a \\ b \end{pmatrix} \cdot (a^* \ b^*) = \begin{pmatrix} |a|^2 & ab^* \\ ba^* & |b|^2 \end{pmatrix} \quad (11.17)$$

where $\langle\psi| \equiv (a^* \ b^*)$ is the bra vector associated with the ket $|\psi\rangle \equiv \begin{pmatrix} a \\ b \end{pmatrix}$. It is obtained by computing the conjugate transpose of the column vector associated with $|\psi\rangle$.

Notice that the sum of the diagonal elements of the density operator is unity, i.e., $\text{tr}(\rho_{\text{pure}}) = 1$. However, as $\rho = |\psi\rangle\langle\psi|$ is actually a pure state (written in density operator formalism) it also happens to be true that $\text{tr}(\rho^2) = 1$ too. Specifically we have,

$$\begin{aligned} \rho^2 &= \begin{pmatrix} |a|^4 + |a|^2|b|^2 & ab^*(|a|^2 + |b|^2) \\ ba^*(|a|^2 + |b|^2) & |b|^4 + |a|^2|b|^2 \end{pmatrix} \\ &= \begin{pmatrix} |a|^2(|a|^2|b|^2) & ab^*(|a|^2 + |b|^2) \\ ba^*(|a|^2 + |b|^2) & |b|^2(|a|^2|b|^2) \end{pmatrix} \\ &= \begin{pmatrix} |a|^2 & ab^* \\ ba^* & |b|^2 \end{pmatrix} = \rho \end{aligned} \quad (11.18)$$

Hence, $\rho^2 = \rho$ and so $\text{tr}(\rho^2) = |a|^2 + |b|^2 = 1$ when ρ is a 1-qubit pure state.

The foregoing results carry over to multi-qubit pure states too. Thus, the density operator associated with a 2-qubit pure state $|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$ is:

$$\rho = |\psi\rangle\langle\psi| = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \cdot (a^* \ b^* \ c^* \ d^*) = \begin{pmatrix} |a|^2 & ab^* & ac^* & ad^* \\ ba^* & |b|^2 & bc^* & bd^* \\ ca^* & cb^* & |c|^2 & cd^* \\ da^* & db^* & dc^* & |d|^2 \end{pmatrix} \quad (11.19)$$

and so on. As you will show in problem ***** for this 2-qubit pure state it is also true that $\rho^2 = \rho$ and $\text{tr}(\rho^2) = (|a|^2 + |b|^2 + |c|^2 + |d|^2)^2 = 1$ (the latter factorization is a hint).

It turns out, whatever the dimensions of ρ , that $\rho^2 = \rho$ and $\text{tr}(\rho^2) = 1$ if and only if ρ is the density operator corresponding to a *pure* state. If the state described by ρ is not pure, but is instead mixed, then $\rho^2 \neq \rho$ and $\text{tr}(\rho^2) < 1$. These properties can be used to decide whether a given state is pure or mixed.

11.2.2.3 The Bloch Ball

In Chap. 1 we introduced the Bloch *sphere* as a way of visualizing single qubit *pure* states. In this picture, the pure states are always points on the *surface* of the Bloch sphere. Since all pure states that differ only by an overall phase factor are indistinguishable, this overall phase factor is not depicted in the Bloch sphere representation. One might wonder where single qubit *mixed* states would reside in this Bloch sphere picture?

The answer is that single qubit mixed states correspond to points *inside* the Bloch sphere, a region we shall henceforth call the Bloch *ball*. After a little algebra, we find that the (x, y, z) coordinates within the Bloch ball corresponding to the mixed state ρ are given by [164]:

$$\begin{aligned} x &= \langle 0|\rho|1\rangle + \langle 1|\rho|0\rangle \\ y &= i\langle 0|\rho|1\rangle - i\langle 1|\rho|0\rangle \\ z &= \langle 0|\rho|0\rangle - \langle 1|\rho|1\rangle \end{aligned} \tag{11.20}$$

with the North Pole corresponding to pure state $|0\rangle$, and the South Pole to pure state $|1\rangle$. Hence, a superposition state such as $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ will have coordinate $(x, y, z) = (1, 0, 0)$ etc. The maximally mixed state is a point, as shown in Fig. 11.3, at the center of the Bloch ball with coordinates $(x, y, z) = (0, 0, 0)$. Non-maximally mixed states lie between the center of the Bloch ball and its surface.

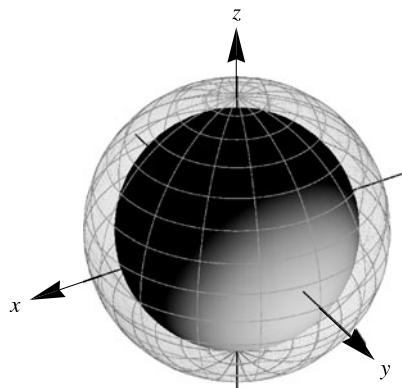


Fig. 11.3 The Bloch Ball can be used to visualize mixed states of a single qubit, which reside on the interior of the Bloch sphere

11.2.2.4 Properties of Density Operators

The quantum mechanical equations based on state vectors, which we have thus far used to describe the evolution and measurement of *pure* states can be re-expressed in the language of density operators. However, the density operator versions apply to the evolution and measurement of both *pure and mixed* states. Consequently, they are more useful, especially when we are dealing with quantum systems for which we have only incomplete knowledge.

- The sum of the diagonal elements of ρ is always 1, i.e., $\text{tr}(\rho(t)) = 1$
- The expected value of an observable $\langle A \rangle = \text{tr}(\rho A)$
- The time evolution of a density operation obeys $i\hbar \frac{d\rho}{dt} = [\mathcal{H}, \rho]$
- The density operator is Hermitian $\rho^\dagger = \rho$
- If ρ corresponds to a pure state $\rho^2 = \rho$
- If ρ corresponds to a pure state $\text{tr}(\rho^2) = 1$
- If ρ corresponds to a pure state the eigenvalues of ρ are either 0 or 1 only
- If ρ corresponds to a mixed state $\frac{1}{d} \leq \text{tr}(\rho^2) < 1$ where d is the dimension of ρ
- A measure of the similarity between two density matrices is given by the fidelity

$$\mathcal{F}(\rho_1, \rho_2) = \left[\text{tr}(\sqrt{\rho_1^{1/2} \rho_2 \rho_1^{1/2}}) \right]^2$$

In Table 11.1 we compare and contrast formulae for performing similar operations on pure states and mixed states. Note that the formulae for mixed states encompass pure states too as a special case, namely, when the density operator takes the form $\rho = |\psi\rangle\langle\psi|$.

11.2.2.5 Non-unique Interpretation of Density Operators

The decomposition of a given density operator into a weighted sum of pure states is *non-unique*. Any decomposition that synthesizes the density operator is as legitimate as any other. This means that there is no unique mixed state to which each density operator corresponds. Moreover, as the expectation value of an observable, \mathcal{O} , is computed from $\text{tr}(\rho\mathcal{O})$, then all these different mixed states (having the same ρ)

Table 11.1 Analogous quantum mechanical formulae for n -qubit pure and mixed states. Note that the mixed state formulae can also be used to describe pure states but not vice versa

Characteristic	Pure state description	Mixed state description
State	$ \psi\rangle = \sum_{j=0}^{2^n-1} c_j j\rangle$	$\rho = \sum_k p_k \phi_k\rangle\langle\phi_k $ where $ \phi_k\rangle$ is an arbitrary n -qubit pure state and $\sum_k p_k = 1$
State evolution	$i\hbar \frac{\partial \psi\rangle}{\partial t} = \mathcal{H} \psi\rangle$	$i\hbar \frac{\partial\rho}{\partial t} = [\mathcal{H}, \rho]$
Component evolution	$\frac{\partial c_j}{\partial t} = -\frac{i}{\hbar} \sum_{\ell=0}^{2^n-1} \mathcal{H}_{j\ell} c_\ell$	$\frac{\partial \rho_{jk}}{\partial t} = -\frac{i}{\hbar} \sum_{\ell=1}^{2^n} [\mathcal{H}_{j\ell} \rho_{\ell k} - \rho_{j\ell} \mathcal{H}_{\ell k}]$
Expected value of observable	$\langle\mathcal{O}\rangle = \langle\psi \mathcal{O} \psi\rangle$	$\langle\mathcal{O}\rangle = \text{tr}(\rho\mathcal{O})$

would produce identical statistical distributions of measurement outcomes whatever observable is used! So there is no experiment we can do that will distinguish between these different mixed states. Operationally, they are all equivalent.

To illustrate the non-uniqueness of the mixed state associated with a given density operator consider the following. Let $|\psi_A\rangle = \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$ and $|\psi_B\rangle = \frac{2}{3}|0\rangle + \frac{\sqrt{5}}{3}|1\rangle$. Then ρ , the density operator corresponding to a mixed state that is $\frac{1}{3}|\psi_A\rangle\langle\psi_A|$ and $\frac{2}{3}|\psi_B\rangle\langle\psi_B|$ can be written as:

$$\begin{aligned}\rho &= \frac{1}{3}|\psi_A\rangle\langle\psi_A| + \frac{2}{3}|\psi_B\rangle\langle\psi_B| \\ &= \begin{pmatrix} 0.37962962962962965 & 0.47560689729737526 \\ 0.47560689729737526 & 0.6203703703703703 \end{pmatrix} \quad (11.21)\end{aligned}$$

However, ρ can be obtained equally well from states $|\phi_A\rangle = a|+\rangle + \sqrt{1-|a|^2}|-\rangle$ and $|\phi_B\rangle = b|+\rangle + \sqrt{1-|b|^2}|-\rangle$ where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ as:

$$\begin{aligned}\rho &= c|\phi_A\rangle\langle\phi_A| + (1-c)|\phi_B\rangle\langle\phi_B| \\ &= \begin{pmatrix} 0.37962962962962965 & 0.47560689729737526 \\ 0.47560689729737526 & 0.6203703703703703 \end{pmatrix} \quad (11.22)\end{aligned}$$

provided $a = -0.875949$, $b = -0.994988$, and $c = 0.064635$. So the question whether ρ is “really” a mixture of the states $|\psi_A\rangle$ and $|\psi_B\rangle$ or a mixture of the states $|\phi_A\rangle$ and $|\phi_B\rangle$ is unanswerable. Each decomposition is as valid as the other.

11.2.3 Mixed States from Partial Ignorance: The Partial Trace

In Sect. 11.2.1 we introduced the concept of the *partial trace* operation. There we explained *what* it was (i.e., the act of ignoring or discarding a subset of the qubits of a multi-partite quantum state) but we did explain *how* to compute it. That is the subject of this section.

The basic idea is that we start off with the quantum mechanical description of a multi-qubit state, and we ask how our description must change if we ignore part of that state. The easiest way to think about this is to partition the set of qubits into two sets A and B and consider a multi-qubit system having density operator ρ_{AB} . In general, ρ_{AB} , will not be a product state, i.e., in general Here the subscript AB signifies that we can arbitrarily

The (i, i') -th element of the reduced density operator, ρ_A obtained by tracing over the second set of qubits B from the state ρ_{AB} is given by:

$$\langle i_A | \rho_A | i'_A \rangle = \text{tr}_B(\rho_{AB}) = \sum_{j_B=0}^{d_B-1} \langle i_A | \langle j_B | \rho_{AB} | i'_A \rangle | j_B \rangle \quad (11.23)$$

where $|i_A\rangle$ and $|i'_A\rangle$ are eigenstates of A subsystem, and $|j_B\rangle$ are eigenstates of the subsystem B (which is a d_B dimensional subspace). Notice that, in the summation, the same eigenstate index j_B is used either side of the ρ_{AB} and the summation is computed over all values for this index. Hence, the reduced density operator ρ_A is obtained by computing each of its possible matrix elements in accordance with (11.23).

Likewise, the (j, j') -th element of the reduced density operator, ρ_B is obtained by tracing over the *first* set of qubits A from the state ρ_{AB} . We have:

$$\langle j_B | \rho_B | j'_B \rangle = \text{tr}_A(\rho_{AB}) = \sum_{i_A=0}^{d_A-1} \langle i_A | \langle j_B | \rho_{AB} | i_A \rangle | j'_B \rangle \quad (11.24)$$

where $|j_B\rangle$ and $|j'_B\rangle$ are eigenstates of the B subsystem, and $|i_A\rangle$ are eigenstates of the subsystem A (which is a d_A dimensional subspace). Notice that, in the summation, the same eigenstate index i_A is used either side of the ρ_{AB} and the summation is computed over all values for this index. Hence, the reduced density operator ρ_B is obtained by computing each of its possible matrix elements in accordance with (11.24).

11.2.3.1 Example: Computing the Partial Trace

For example, consider a pair of non-orthogonal quantum states $|\psi_{ABC}\rangle$ and $|\varphi_{ABC}\rangle$ defined as follows:

$$|\psi_{ABC}\rangle = \frac{1}{2}|000\rangle + \frac{\sqrt{3}}{2}|111\rangle \quad (11.25)$$

$$|\varphi_{ABC}\rangle = \frac{1}{2}|000\rangle + \frac{1}{2}|010\rangle + \frac{1}{2}|011\rangle + \frac{1}{2}|111\rangle \quad (11.26)$$

and imagine these are the components of the density operator weighted to be one third $|\psi_{ABC}\rangle$ and two thirds $|\varphi_{ABC}\rangle$. Thus we have:

$$\begin{aligned} \rho_{ABC} &= \frac{1}{3}|\psi_{ABC}\rangle\langle\psi_{ABC}| + \frac{2}{3}|\varphi_{ABC}\rangle\langle\varphi_{ABC}| \\ &= \begin{pmatrix} \frac{1}{4} & 0 & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & \frac{1}{6} + \frac{1}{4\sqrt{3}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & \frac{1}{6} \\ \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & \frac{1}{6} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{6} + \frac{1}{4\sqrt{3}} & 0 & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & \frac{5}{12} \end{pmatrix} \end{aligned} \quad (11.27)$$

Tracing over any one of the qubits we obtain the three reduced density matrices ρ_{BC} , ρ_{AC} , and ρ_{AB} :

$$\rho_{BC} = \text{tr}_A(\rho_{ABC}) = \begin{pmatrix} \frac{1}{4} & 0 & \frac{1}{6} & \frac{1}{6} \\ 0 & 0 & 0 & 0 \\ \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & 0 & \frac{1}{6} & \frac{7}{12} \end{pmatrix} \quad (11.28)$$

$$\rho_{AC} = \text{tr}_B(\rho_{ABC}) = \begin{pmatrix} \frac{5}{12} & \frac{1}{6} & 0 & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} \\ 0 & 0 & 0 & 0 \\ \frac{1}{6} & \frac{1}{6} & 0 & \frac{5}{12} \end{pmatrix} \quad (11.29)$$

$$\rho_{AB} = \text{tr}_C(\rho_{ABC}) = \begin{pmatrix} \frac{1}{4} & \frac{1}{6} & 0 & 0 \\ \frac{1}{6} & \frac{1}{3} & 0 & \frac{1}{6} \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{6} & 0 & \frac{5}{12} \end{pmatrix} \quad (11.30)$$

Likewise, tracing over any two of the three qubits we obtain the three reduced density matrices ρ_A , ρ_B , and ρ_C :

$$\rho_A = \text{tr}_{BC}(\rho_{ABC}) = \begin{pmatrix} \frac{7}{12} & \frac{1}{6} \\ \frac{1}{6} & \frac{5}{12} \end{pmatrix} \quad (11.31)$$

$$\rho_B = \text{tr}_{AC}(\rho_{ABC}) = \begin{pmatrix} \frac{1}{4} & \frac{1}{6} \\ \frac{1}{6} & \frac{3}{4} \end{pmatrix} \quad (11.32)$$

$$\rho_C = \text{tr}_{AB}(\rho_{ABC}) = \begin{pmatrix} \frac{5}{12} & \frac{1}{6} \\ \frac{1}{6} & \frac{7}{12} \end{pmatrix} \quad (11.33)$$

Thus, the partial trace operation provides the procedure for calculating the quantum state of *part* of a composite quantum system. In general, if the starting state is entangled and pure (say) the restriction of this state to some subset of its qubits, i.e., its partial trace, will, in general, be a mixed state described mathematically by a reduced density operator.

11.2.4 Mixed States as Parts of Larger Pure States: “Purifications”

The foregoing interpretation of the partial trace operation invites the question of whether there is a procedure for going in the opposite direction? That is, starting with a mixed state, which we can think of as the reduced density operator of some larger *pure* state, is there a procedure for finding this larger pure state? The answer is that there is such a procedure. It is called a state *purification* operation, and it works as follows:

Purification of a Mixed State Let $\rho_A = \sum_i p_i |\psi_i\rangle\langle\psi_i|$ be a n -qubit mixed state defined on a Hilbert space \mathcal{H}_A of dimension $d = 2^n$. Our goal is to find a pure state $|\Psi\rangle_{AB}$, defined on a Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$ such that $\text{tr}_B(|\Psi\rangle_{AB}\langle\Psi|_{AB}) = \rho_A$. Such a $|\Psi\rangle_{AB}$ is a purification of the mixed state ρ_A .

1. Rewrite the mixed state ρ_A as:

$$\rho_A = \sum_{i=1}^N p_i |\psi_i\rangle\langle\psi_i| = \sum_{j=1}^d \lambda_j |\phi_j\rangle\langle\phi_j| \quad (11.34)$$

where $\{\lambda_j\}$ are the eigenvalues of ρ_A and $\{|\phi_j\rangle\}$ are the eigenvectors of ρ_A . Note that there are d eigenvalues and eigenvectors, whereas there are N states contributing to the original definition of ρ_A .

2. Pick out just the first N eigenvalues and eigenvectors from the basis $\{|\phi_j\rangle\}$. Then construct the pure state $|\Psi\rangle_{AB}$ defined as:

$$|\Psi_{AB}\rangle = \sum_{i=1}^N \sqrt{p_i} |\psi_i\rangle |\phi_i\rangle \quad (11.35)$$

3. The given $|\Psi\rangle_{AB}$ is a purification of ρ_A since $\text{tr}_B(|\Psi\rangle_{AB}\langle\Psi|_{AB}) = \rho_A$.

11.2.5 Quantifying Mixedness

How do we quantify the degree of mixedness in a state given its description in terms of a density operator? Clearly, our measure of “mixedness” must range from zero (for pure states) to some maximum value (for maximally mixed states). But what measure should we use? In this section we look at some ways to quantify the degree of mixedness of a quantum state.

11.2.5.1 Linear Entropy as a Measure of Mixedness

The first measure of mixedness is related to its deviation from a pure state. In particular, we saw in Sect. 11.2.2.2 that if a state with density matrix ρ is pure $\rho^2 = \rho$ and therefore $\text{tr}(\rho^2) = \text{tr}(\rho) = 1$, whereas if it is mixed, $\frac{1}{d} \leq \text{tr}(\rho^2) < 1$ where d is the dimension of ρ . Hence the deviation of $\text{tr}(\rho^2)$ from 1 can be used as a measure for the mixedness of ρ . This gives us our first measure of mixedness called the *linear entropy* of ρ , which is especially easy to calculate:

$$S_L(\rho) = \frac{d}{d-1} (1 - \text{tr}(\rho^2)) \quad (11.36)$$

where d is the dimension of ρ . Hence, $0 \leq S_L(\rho) \leq 1$: the linear entropy $S_L(\rho) = 0$ whenever ρ is a pure state, and $S_L(\rho) = 1$ whenever ρ is a maximally mixed state.

11.2.5.2 von Neumann Entropy as a Measure of Mixedness

A second measure of mixedness is the von Neumann entropy, $S_V(\rho)$, which is the proper quantum generalization of the Shannon entropy.

To remind you, in classical information theory, the Shannon entropy of a classical source that outputs d distinguishable symbols with corresponding probabilities p_1, p_2, \dots, p_d is given by $H(\{p_i\}) = -\sum_{i=1}^d p_i \log_2 p_i$ where $\sum_{i=1}^d p_i = 1$. This ranges from 0 in the case when all the symbols are the same, to $\log_2 d$ in the case when all the d symbols are equiprobable, and therefore maximally random. One can therefore think of the Shannon entropy as quantifying the degree of randomness in the symbols streaming from a classical source.

What is the analog of Shannon entropy in the quantum context? We can think of a quantum source as outputting d *not necessarily orthogonal* quantum states $|\psi_1\rangle, |\psi_2\rangle, \dots, |\psi_d\rangle$ with corresponding probabilities p_1, p_2, \dots, p_d . Such a source is characterized by the density operator ρ given by:

$$\rho = \sum_{i=1}^d p_i |\psi_i\rangle\langle\psi_i| \quad (11.37)$$

where $\sum_{i=1}^d p_i = 1$.

However, a given density operator can be decomposed into a sum of component states in many different ways, which are all equivalent to one another. In particular, even if the states $|\psi_i\rangle$ are non-orthogonal, we can always *diagonalize* ρ by finding a unitary matrix, U , such that $U\rho U^\dagger$ is a diagonal matrix. Thus, any density operator ρ can also be written as:

$$\rho = \sum_{i=1}^d p_i |\psi_i\rangle\langle\psi_i| = \sum_j \lambda_j |\lambda_j\rangle\langle\lambda_j| \quad (11.38)$$

When so diagonalized, the eigenvalues of ρ , i.e., the λ_j appearing along the main diagonal, are positive real numbers that sum to one, and correspond to the probabilities with which we will see the corresponding eigenvectors of ρ , i.e., $|\lambda_j\rangle$, if ρ were measured in its eigenbasis. As these eigenvectors $|\lambda_j\rangle$ are orthogonal to one another they are distinguishable and we can therefore regard them as classical symbols. Therefore, when viewed in the diagonal basis, we would expect the quantum entropy of the quantum source to coincide with the Shannon entropy of the analogous classical source, i.e., one emitting the “classical” (i.e., perfectly distinguishable) symbols, or equivalently orthonormal states $|\lambda_j\rangle$, with corresponding probabilities λ_j . This allows us to define the entropy of the quantum source (which may or may not output distinguishable symbols) in terms of the Shannon entropy of a corresponding fictitious classical source (which outputs only distinguishable symbols). In particular, we have:

$$S_V(\rho) = -\sum_j \lambda_j \log_2 \lambda_j = H(\{\lambda_j\}) \quad (11.39)$$

where we take $0 \log_2 0 = 0$. Using purely mathematical arguments (i.e. no new physics insights), we can rewrite (11.39) as:

$$S_V(\rho) = -\text{tr}(\rho \log_2 \rho) \quad (11.40)$$

This is the von Neumann entropy of the quantum source described by density operator ρ .

It is apparent from its definition that the von Neumann entropy is bounded as follows:

$$0 \leq S_V(\rho) \leq \log_2 d \quad (11.41)$$

with the von Neumann entropy being 0 for a pure state $\rho = |\psi\rangle\langle\psi|$, and $\log_2 d$ for a maximally mixed state $\rho = \frac{1}{d}\mathbb{1}$, where $\mathbb{1}$ is the identity matrix. Thus, the numerical value of the von Neumann entropy is a measure of the mixedness of the state.

11.3 Entanglement

“No self is of itself alone.”

– Erwin Schrödinger

“Entanglement” describes a correlation between different parts of a quantum system that exceeds anything that is possible classically. It will appear when sub-systems interact in such a way that the resulting state of the whole system cannot be expressed as the direct product of states for its parts. When a quantum system is in such an entangled state, actions performed on one sub-system will have a side-effect on another sub-system even though that sub-system is not acted upon directly. Moreover, provided the sub-systems are separated in such a way that neither is measured, such entanglement will persist regardless of how far apart the sub-systems become. This leads to highly counterintuitive phenomena, which Einstein dubbed “spooky action at a distance”, which we will have more to say about in Chap. 12.

All the known quantum algorithms that display an exponential speedup over their classical counterparts exploit such entanglement-induced side effects in one way or another. In addition, some tasks that are impossible by classical standards, such as teleporting a quantum state, depend upon entanglement in an essential way. Hence, entanglement deserves to be called a “quintessential” quantum phenomenon that plays a major role in making quantum computing more powerful than classical computing, and in enabling quantum information tasks that are impossible in the classical context.

11.3.1 Separable States Versus Entangled States

Formally, the distinction between whether a state is entangled or not entangled rests upon whether its quantum state is separable or not. Therefore, let us examine this question in more mathematical terms.

Suppose we have two independent quantum systems with Hilbert spaces H_A and H_B of dimensions d_A and d_B respectively. There is some complete orthonormal basis for H_A consisting of d_A eigenstates, called $\{|j_A\rangle\}$ say, and some complete orthonormal basis for H_B consisting of d_B eigenstates, $\{|k_B\rangle\}$ say. In other words, any pure state in H_A can be expressed as $|\psi_A\rangle = a_0|0\rangle_A + a_1|1\rangle_A + \dots + a_{d_A-1}|d_A-1\rangle_A$. Likewise, any pure state in H_B can be expressed as $|\psi_B\rangle = b_0|0\rangle_B + b_1|1\rangle_B + \dots + b_{d_B-1}|d_B-1\rangle_B$. And the Hilbert space of the composite system is just the tensor product of the constituent Hilbert spaces $H = H_A \otimes H_B$.

Separable State If a pure (mixed) state, $|\psi^{(AB)}\rangle$ ($\rho^{(AB)}$), of a composite quantum system defined on a Hilbert space $H_A \otimes H_B$ can be written as $|\psi^{(AB)}\rangle = |\psi^{(A)}\rangle \otimes |\psi^{(B)}\rangle$ ($\rho^{(AB)} = \sum_i p_i \rho_i^{(A)} \otimes \rho_i^{(B)}$), then $|\psi^{(AB)}\rangle$ ($\rho^{(AB)}$) is said to be a *separable state*.

The linear entropy $S_L(\rho)$, can also be useful in deciding whether a given density operator ρ corresponds to a separable or entangled state. Specifically, it has been proven that if the linear entropy exceeds a certain threshold, i.e., if $S_L(\rho) \geq d(d-2)/(d-1)^2$, then any such ρ is separable [567].

Entangled State If a state, $|\psi^{(AB)}\rangle$ ($\rho^{(AB)}$), of a composite quantum system defined on a Hilbert space $H_A \otimes H_B$ is not a separable state it is an *entangled state*. Note that a state can be entangled and pure, or entangled and mixed, simultaneously.

As an example, consider the state $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$. Is this state separable or entangled? Well, if it were separable it could be written in the form $|\psi_A\rangle \otimes |\psi_B\rangle$ where $|\psi_A\rangle = a_0|0\rangle + a_1|1\rangle$ and $|\psi_B\rangle = b_0|0\rangle + b_1|1\rangle$. Thus, equating amplitudes and solving we have:

$$\begin{aligned} |\psi_A\rangle \otimes |\psi_B\rangle &= a_0b_0|00\rangle + a_0b_1|01\rangle + a_1b_0|10\rangle + a_1b_1|11\rangle \\ &= \frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle \end{aligned} \quad (11.42)$$

which implies we need to find a solution to the simultaneous equations $a_0b_0 = 0$, $a_0b_1 = \frac{1}{\sqrt{2}}$, $a_1b_0 = -\frac{1}{\sqrt{2}}$, $a_1b_1 = 0$. Unfortunately, these equations admit no such solution and hence the state $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ is not separable. Hence it is entangled.

11.3.2 Signalling Entanglement via Entanglement Witnesses

Given a purported entangled state, ρ , how can we verify that ρ is, in fact, entangled?

One approach is to synthesize several instances of the state ρ via identical preparation procedures and then perform quantum state tomography to reconstruct the density operator for ρ . This is, in one sense, the preferred option since we would obtain *complete* information about ρ —at least to within experimental error.

However, in general, quantum state tomography is an extraordinarily costly procedure. An n -qubit state is described by a $2^n \times 2^n$ dimensional density operator. If we are to determine each element of this density operator empirically, we would need to perform $O(2^{2n})$ different experiments! Thus full quantum state tomography becomes quite impractical for quantum systems having more than a mere handful of qubits.

This difficulty spawned the invention of *entanglement witnesses* [166]. Entanglement witnesses are tools for detecting entanglement that avoid having to perform a complete quantum state tomographic reconstruction of ρ .

The basic idea is to construct an observable operator, W , whose expectation value serves as a “witness” to whether the given state is entangled. If the expectation value of the witness observable W when the system is in state ρ , i.e., $\text{tr}(W\rho) = \langle W \rangle$, is less than some threshold, this provides sufficient evidence that ρ is an entangled state.

Although the fully theory of entanglement witnesses requires an understanding of the superoperator formalism of quantum mechanics, entanglement witnesses need not be that exotic.

11.3.2.1 Example: Entanglement Witness

For example, consider the one-dimensional “Heisenberg chain”. This consists of a one-dimensional loop of spins coupled together in accordance with the Hamiltonian:

$$\mathcal{H} = \sum_{i=1}^N (B\sigma_z^i + J\boldsymbol{\sigma}^i \cdot \boldsymbol{\sigma}^{i+1}) \quad (11.43)$$

where B is the external magnetic field, and a $J < 0$ or $J > 0$ are, respectively, a ferromagnetic or anti-ferromagnetic coupling between the spins. The symbol σ_z^i stands for an the Pauli-Z operator that acts on the i -th qubit, and the vectors $\boldsymbol{\sigma}^i \equiv (\sigma_x^i, \sigma_y^i, \sigma_z^i)$. The one-dimensional chain of spins is made periodic by choosing $\boldsymbol{\sigma}^{N+1} = \boldsymbol{\sigma}^1$.

In the absence of an external magnetic field, i.e., with $B = 0$, the expectation value for the energy, $\langle \mathcal{H} \rangle = \text{tr}(\rho \mathcal{H})$, can be an entanglement witness. Specifically, suppose we have chain consisting of two spins, i.e., $N = 2$. In this case, if the input state is separable, i.e., $\rho_{AB} = \rho_A \otimes \rho_B$, then it is possible to show that the expectation value of the energy is *guaranteed* to be bounded between $-2J \leq \langle \mathcal{H} \rangle \leq +2J$.

However, if the state ρ_{AB} is entangled, $\rho_{AB} \neq \rho_A \otimes \rho_B$, we find that there are entangled states for which $\langle \mathcal{H} \rangle < -2J$. Thus, by measuring expectation value of the energy, $\langle \mathcal{H} \rangle$, we can sometimes decide if the state is entangled. Hence, $\langle \mathcal{H} \rangle$ serves as an entanglement witness.

11.3.3 Signalling Entanglement via the Peres-Horodecki Criterion

An alternative to relying on entanglement witnesses to decide if a state is entangled, is to use the Peres-Horodecki criterion [239, 387]. This criterion uses an operation on a density matrix known as the *partial transpose*.

Definition: Partial Transpose Let ρ be a bi-partite density operator expressed in the form:

$$\rho = \sum_{i,j,k,\ell} \rho_{ij;k\ell} |e_i^A \otimes e_j^B\rangle \langle e_k^A \otimes e_\ell^B| \quad (11.44)$$

where $\{|e_i^A\rangle\}$ is an eigenbasis for sub-space A and $\{|e_j^B\rangle\}$ is an eigenbasis for sub-space B. Then the partial transpose ρ^{T_B} of the density operator ρ is:

$$\rho^{T_B} = \sum_{i,j,k,\ell} \rho_{i\ell;jk} |e_i^A \otimes e_j^B\rangle \langle e_k^A \otimes e_\ell^B| \quad (11.45)$$

Note that, as implied by the definition, the partial transpose depends on the basis chosen but the *eigenvalues* of the partial transposed matrix do not. However, most practical applications of the partial transpose only need to make use of the eigenvalues of the partial transpose matrix.

The partial transpose is important within a test for entanglement known as the Peres-Horodecki criterion [239, 387].

Peres-Horodecki Criterion: a Necessary and Sufficient Test for Entanglement
If a bi-partite state is entangled, its partial transpose always has one or more negative eigenvalues, but if it is separable its partial transpose has no negative eigenvalues.

Thus, given a density operator ρ we can decide whether or not it is entangled by examining the signs of the eigenvalues of its partial transpose.

Note that we can define an analogous partial transpose over the “A” space as follows:

$$\rho^{T_A} = \sum_{i,j,k,\ell} \rho_{kj;i\ell} |e_i^A \otimes e_j^B\rangle \langle e_k^A \otimes e_\ell^B| \quad (11.46)$$

Even though the partial transpose ρ^{T_A} will usually be a different matrix from the partial transpose ρ^{T_B} their eigenvalues will be the same. In applications of the partial transpose it is usually the eigenvalues of the partial transpose that we need rather than the partial transpose itself. If this is the case, whether we use ρ^{T_A} or ρ^{T_B} is immaterial as their eigenvalues are the same.

For example, let us compute ρ^{T_A} and ρ^{T_B} for a general 2-qubit density matrix defined by:

$$\rho = \begin{pmatrix} \rho_{11} & \rho_{12} & \rho_{13} & \rho_{14} \\ \rho_{21} & \rho_{22} & \rho_{23} & \rho_{24} \\ \rho_{31} & \rho_{32} & \rho_{33} & \rho_{34} \\ \rho_{41} & \rho_{42} & \rho_{43} & \rho_{44} \end{pmatrix} \quad (11.47)$$

Computing the partial transposes we obtain:

$$\rho^{T_A} = \begin{pmatrix} \rho_{11} & \rho_{12} & \rho_{31} & \rho_{32} \\ \rho_{21} & \rho_{22} & \rho_{41} & \rho_{42} \\ \rho_{13} & \rho_{14} & \rho_{33} & \rho_{34} \\ \rho_{23} & \rho_{24} & \rho_{43} & \rho_{44} \end{pmatrix} \quad (11.48)$$

$$\rho^{T_B} = \begin{pmatrix} \rho_{11} & \rho_{21} & \rho_{13} & \rho_{23} \\ \rho_{12} & \rho_{22} & \rho_{14} & \rho_{24} \\ \rho_{31} & \rho_{41} & \rho_{33} & \rho_{43} \\ \rho_{32} & \rho_{42} & \rho_{34} & \rho_{44} \end{pmatrix}$$

However, the characteristic polynomials of ρ^{T_A} and ρ^{T_B} are identical, and so the eigenvalues of these matrices must be the same.

Case 1: a Separable Pure State

Let us look at some simple examples. Consider first the case of an unentangled pure state. In this case we have:

$$|\psi_{AB}\rangle = \left(\frac{1}{2}|0\rangle + \sqrt{\frac{3}{4}}|1\rangle \right) \otimes \left(\frac{1}{3}|0\rangle + \sqrt{\frac{8}{9}}|1\rangle \right) \quad (11.49)$$

$$\rho_{AB} = |\psi_{AB}\rangle\langle\psi_{AB}| = \begin{pmatrix} \frac{1}{36} & \frac{1}{9\sqrt{2}} & \frac{1}{12\sqrt{3}} & \frac{1}{3\sqrt{6}} \\ \frac{1}{9\sqrt{2}} & \frac{2}{9} & \frac{1}{3\sqrt{6}} & \frac{2}{3\sqrt{3}} \\ \frac{1}{12\sqrt{3}} & \frac{1}{3\sqrt{6}} & \frac{1}{12} & \frac{1}{3\sqrt{2}} \\ \frac{1}{3\sqrt{6}} & \frac{2}{3\sqrt{3}} & \frac{1}{3\sqrt{2}} & \frac{2}{3} \end{pmatrix} \quad (11.50)$$

$$\rho_{AB}^{T_B} = \begin{pmatrix} \frac{1}{36} & \frac{1}{9\sqrt{2}} & \frac{1}{12\sqrt{3}} & \frac{1}{3\sqrt{6}} \\ \frac{1}{9\sqrt{2}} & \frac{2}{9} & \frac{1}{3\sqrt{6}} & \frac{2}{3\sqrt{3}} \\ \frac{1}{12\sqrt{3}} & \frac{1}{3\sqrt{6}} & \frac{1}{12} & \frac{1}{3\sqrt{2}} \\ \frac{1}{3\sqrt{6}} & \frac{2}{3\sqrt{3}} & \frac{1}{3\sqrt{2}} & \frac{2}{3} \end{pmatrix} \quad (11.51)$$

$$\text{Eigenvalues}(\rho_{AB}^{T_B}) = \{1, 0, 0, 0\} \quad (11.52)$$

As all of the eigenvalues of the partial transpose of $\rho_{AB}^{T_B}$ are positive this guarantees, by the Peres-Horodecki criterion, that ρ_{AB} is separable.

Case 2: an Entangled Pure State

Now let's look what happens when we have an entangled pure state such as $|\psi_{AB}\rangle = (\frac{1}{2}|01\rangle - \sqrt{\frac{3}{4}}|10\rangle)$:

$$|\psi_{AB}\rangle = \left(\frac{1}{2}|01\rangle - \sqrt{\frac{3}{4}}|10\rangle \right) \quad (11.53)$$

$$\rho_{AB} = |\psi_{AB}\rangle\langle\psi_{AB}| = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & -\frac{\sqrt{3}}{4} & 0 \\ 0 & -\frac{\sqrt{3}}{4} & \frac{3}{4} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (11.54)$$

$$\rho_{AB}^{T_B} = \begin{pmatrix} 0 & 0 & 0 & -\frac{\sqrt{3}}{4} \\ 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & \frac{3}{4} & 0 \\ -\frac{\sqrt{3}}{4} & 0 & 0 & 0 \end{pmatrix} \quad (11.55)$$

$$\text{Eigenvalues}(\rho_{AB}^{T_B}) = \left\{ \frac{3}{4}, -\frac{\sqrt{3}}{4}, \frac{\sqrt{3}}{4}, \frac{1}{4} \right\} \quad (11.56)$$

As one of the eigenvalues of the partial transpose of $\rho_{AB}^{T_B}$ is negative, this guarantees by the Peres-Horodecki criterion, that ρ_{AB} is entangled.

Case 3: a Separable Mixed State

The Peres-Horodecki criterion is not limited to deciding whether only pure states are entangled or separable. It also applies to mixed states. For example, the mixed state $\rho_{AB} = \frac{1}{3}\rho_A \otimes \rho_B + \frac{2}{3}\rho'_A \otimes \rho'_B$ is, by construction, separable. The Peres-Horodecki criterion gives us:

$$\rho_A = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \quad (11.57)$$

$$\rho_B = \begin{pmatrix} \frac{2}{3} & -\frac{i}{3} \\ \frac{i}{3} & \frac{1}{3} \end{pmatrix} \quad (11.58)$$

$$\rho'_A = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \quad (11.59)$$

$$\rho'_B = \begin{pmatrix} \frac{1}{8} & \frac{i\sqrt{3}}{8} \\ -\frac{i\sqrt{3}}{8} & \frac{7}{8} \end{pmatrix} \quad (11.60)$$

$$\begin{aligned} \rho_{AB} &= \frac{1}{3}\rho_A \otimes \rho_B + \frac{2}{3}\rho'_A \otimes \rho'_B \\ &= \begin{pmatrix} \frac{11}{72} & -\frac{i}{18} + \frac{i}{8\sqrt{3}} & \frac{1}{24} & \frac{i}{8\sqrt{3}} \\ \frac{i}{18} - \frac{i}{8\sqrt{3}} & \frac{25}{72} & -\frac{i}{8\sqrt{3}} & \frac{7}{24} \\ \frac{1}{24} & \frac{i}{8\sqrt{3}} & \frac{11}{72} & -\frac{i}{18} + \frac{i}{8\sqrt{3}} \\ -\frac{i}{8\sqrt{3}} & \frac{7}{24} & \frac{i}{18} - \frac{i}{8\sqrt{3}} & \frac{25}{72} \end{pmatrix} \end{aligned} \quad (11.61)$$

$$\rho_{AB}^{T_B} = \begin{pmatrix} \frac{11}{72} & \frac{i}{18} - \frac{i}{8\sqrt{3}} & \frac{1}{24} & -\frac{i}{8\sqrt{3}} \\ -\frac{i}{18} + \frac{i}{8\sqrt{3}} & \frac{25}{72} & \frac{i}{8\sqrt{3}} & \frac{7}{24} \\ \frac{1}{24} & -\frac{i}{8\sqrt{3}} & \frac{11}{72} & \frac{i}{18} - \frac{i}{8\sqrt{3}} \\ \frac{i}{8\sqrt{3}} & \frac{7}{24} & -\frac{i}{18} + \frac{i}{8\sqrt{3}} & \frac{25}{72} \end{pmatrix} \quad (11.62)$$

$$\text{Eigenvalues}(\rho_{AB}^{T_B}) = \left\{ \frac{1}{36} \left(15 + \sqrt{95 - 12\sqrt{3}} \right), \frac{1}{36} \left(15 - \sqrt{95 - 12\sqrt{3}} \right), \frac{1}{36} (3 + \sqrt{5}), \frac{1}{36} (3 - \sqrt{5}) \right\} \quad (11.63)$$

As all of the eigenvalues of the partial transpose of $\rho_{AB}^{T_B}$ are positive this guarantees, by the Peres-Horodecki criterion, that ρ_{AB} is separable.

Case 4: an Entangled Mixed State

Finally we consider what happens when the state is entangled and mixed.

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad (11.64)$$

$$|\beta_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \quad (11.65)$$

$$\rho_{AB} = \frac{1}{3}|\beta_{00}\rangle\langle\beta_{00}| + \frac{2}{3}|\beta_{11}\rangle\langle\beta_{11}| = \begin{pmatrix} \frac{1}{6} & 0 & 0 & \frac{1}{6} \\ 0 & \frac{1}{3} & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{6} & 0 & 0 & \frac{1}{6} \end{pmatrix} \quad (11.66)$$

$$\rho_{AB}^{T_B} = \begin{pmatrix} \frac{1}{6} & 0 & 0 & -\frac{1}{3} \\ 0 & \frac{1}{3} & \frac{1}{6} & 0 \\ 0 & \frac{1}{6} & \frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 0 & \frac{1}{6} \end{pmatrix} \quad (11.67)$$

$$\text{Eigenvalues}(\rho_{AB}^{T_B}) = \left\{ \frac{1}{2}, \frac{1}{2}, -\frac{1}{6}, \frac{1}{6} \right\} \quad (11.68)$$

As one of the eigenvalues of the partial transpose of $\rho_{AB}^{T_B}$ is negative, this guarantees by the Peres-Horodecki criterion, that ρ_{AB} is entangled.

11.3.4 Quantifying Entanglement

Rather than merely witnessing of detecting the presence or absence of entanglement, we would prefer to be able to *quantify* the degree of entanglement in a quantum state. Such quantitative methods are necessary if we to have any hope of understanding entanglement properly and how it changes under various physical operations.

Although there is only one effective measure of entanglement in 2-qubit systems—the “tangle”, which we used in Sect. 2.8 to quantify the entangling power of a 2-qubit quantum gate—once we go to three or more qubits the situation becomes extraordinarily complicated. Even at three qubits we start to encounter counterintuitive results such as the possibility of having 3-qubit states that possess 3-way entanglement but for which there is no 2-way entanglement amongst every pair of constituent qubits when considered in isolation!

At present we are stuck with having to wrestle with the notion of entanglement, and having to live with several different and inequivalent ways of quantifying how much entanglement there is in a multi-qubit state. These measures are called entanglement monotones and all share certain desirable properties.

11.3.4.1 Entanglement Monotones

“All science is either physics or stamp collecting.”
– Ernest Rutherford

“Entanglement monotones” are quantitative measures of entanglement of a quantum state, ρ , that increase, monotonically, with the degree of entanglement in the state. Such measures, some of which are summarized in Table 11.2, allow us to compare and contrast the amount of entanglement in different states and hence, to begin to develop a classification for entangled states. We start by stipulating general properties any reasonable measure of entanglement, $E(\rho)$, must possess and then outline some functions that meet these criteria.

Table 11.2 Entanglement monotones

Measure	Explanation
Entanglement of formation $E_F(\rho_{AB}) = \min_{\{\rho_i, \psi_i\rangle\}} \sum_i p_i S(\rho_i^{(A)})$	Quantifies the amount of entanglement needed to synthesize ρ . In essence, it measures how many maximally entangled pairs are needed to synthesize ρ . The minimization is computed over all possible decompositions of ρ_{AB} into sums of pure states making E_F very costly to compute
Entanglement of distillation $E_D(\rho_{AB}) = \lim_{n \rightarrow \infty} m/n$	Quantifies the number of Bell states that can be distilled from ρ per copy of ρ using the optimal purification procedure. Here m is the maximum number of Bell states that can be distilled from n preparations of the state ρ . E_D is also difficult to calculate in practice
Relative entropy of entanglement $E_R(\rho) = \min_{\sigma \in D} \text{tr}(\rho \log \rho - \rho \log \sigma)$	Quantifies the distance of the entangled state ρ from the nearest separable state in the set of all separable density operators D . E_R is relatively easy to compute and happens to exactly equal E_F for pure states of 2-qubit systems
Negativity $E_N(\rho) = 2 \max(0, -\sum_i \lambda_i^{\text{negative}}(\rho^{T_B}))$	Quantifies the entanglement in a state as the degree to which the positive partial transpose separability criterion is violated. If a state is not entangled, the partial transpose of its density operator, ρ^{T_B} , is also a valid density operator, i.e., a positive semi-definite matrix. However, if a state is entangled, the partial transpose of its density operator is not positive semi-definite because it has at least one negative eigenvalue. Hence, negativity quantifies entanglement as the degree to which the positive partial transpose separability criterion is violated. For 2-qubit pure states the negativity equals the concurrence. In the formula for negativity where $\lambda_i^{\text{negative}}(\rho^{T_B})$ is the i -th negative eigenvalue of the partial transpose of ρ

1. For any entanglement measure $E(\rho)$ we require $0 \leq E(\rho) \leq 1$ with $E(\rho) = 0$ if and only if ρ is not entangled, and $E(\rho) = 1$ at least when ρ is the density operator of any maximally entangled state, such as one of the Bell states.
2. The entanglement measure should be immune to local operations, i.e. $E(\rho) = E((U_A \otimes U_B)\rho(U_A \otimes U_B)^\dagger)$.
3. The entanglement measure of the full density operator, i.e., $E(\rho) = E(\sum_i p_i \rho_i)$ cannot be greater than the weighted sum of the entanglement measures of its parts, i.e., $E(\rho) = E(\sum_i p_i \rho_i) \leq \sum_i p_i E(\rho_i)$.

Given the aforementioned desiderata, the following candidates have been identified as acceptable measures of entanglement.

For the case of 2-qubits the different measures of entanglement turn out to be equivalent, and it is therefore simplest to work with the tangle (see Sect. 2.8.1). However, this equivalence does not hold for larger numbers of qubits.

11.3.5 Maximally Entangled Pure States

The most famous maximally entangled pure states are the 2-qubit Bell states:

$$\begin{aligned} |\beta_{00}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\beta_{01}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\beta_{10}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\beta_{11}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \end{aligned} \quad (11.69)$$

The structure of the Bell states invite generalizations in two ways: Either we can extend the pattern we see in the $|\beta_{00}\rangle$ state, and conceive of a two-component superposition having one state with all 0's and the other with all 1's, or we can extend the pattern we see in the $|\beta_{01}\rangle$ state, and conceive of an n -component superposition having a single 1 each in each component at each possible qubit position. This leads to two fundamentally different kinds of maximally entangled states called GHZ and W states respectively. GHZ and W states are defined as follows:

$$\begin{aligned} |\text{GHZ}\rangle &= \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \\ |\text{GHZ}\rangle &= \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle) \\ &\vdots \\ |\text{GHZ}\rangle &= \frac{1}{\sqrt{2}}(|00\dots 0\rangle + |11\dots 1\rangle) \end{aligned} \quad (11.70)$$

and

$$\begin{aligned} |\text{W}\rangle &= \frac{1}{\sqrt{3}}(|001\rangle + |010\rangle + |100\rangle) \\ |\text{W}\rangle &= \frac{1}{\sqrt{4}}(|0001\rangle + |0010\rangle + |0100\rangle + |1000\rangle) \\ &\vdots \\ |\text{W}\rangle &= \frac{1}{\sqrt{n}}(|0\dots 01\rangle + |0\dots 10\rangle + \dots + |1\dots 00\rangle) \end{aligned} \quad (11.71)$$

These two kinds of states are maximally entangled and pure, but are nevertheless fundamentally inequivalent! That is, we cannot inter-convert from GHZ states to W states, or vice versa, using any unitary transformation [253].

Moreover, GHZ and W states behave quite differently under the partial trace operation. For example, tracing over the last qubit in a 3-qubit GHZ state, we obtain:

$$\text{tr}_3(|\text{GHZ}\rangle\langle\text{GHZ}|) = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{pmatrix} = \frac{1}{2}(|00\rangle\langle 00| + |11\rangle\langle 11|) \quad (11.72)$$

which has *no* residual 2-qubit entanglement. However, tracing over the last qubit in a 3-qubit W state results in a state that does have residual 2-qubit entanglement:

$$\begin{aligned} \text{tr}_3(|\text{W}\rangle\langle\text{W}|) &= \begin{pmatrix} \frac{1}{3} & 0 & 0 & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ &= \frac{1}{3} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \frac{2}{3} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ &= \frac{1}{3}|00\rangle\langle 00| + \frac{2}{3}\left(\frac{|01\rangle + |10\rangle}{\sqrt{2}}\right)\left(\frac{\langle 01| + \langle 10|}{\sqrt{2}}\right) \end{aligned} \quad (11.73)$$

The indicated factorization of the reduced density matrix can be interpreted as including a component in $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$, which is one of the Bell states.

11.3.6 Maximally Entangled Mixed States

How does the concept of a maximally entangled state generalize to the case of mixed states? In what sense can a mixed state be said to be maximally entangled?

A superficially reasonable definition of a maximally entangled mixed state is a state that, for a given level of mixedness, attains the highest possible value for entanglement. Unfortunately, it turns out that such a definition is problematic without further qualification. This is because, by the above definition, the identity of the mixed states that are deemed maximally entangled will *change* depending on the measures one chooses with which to quantify the degree of mixedness and quantify the degree of entanglement in the state! This problem appears to be fundamental and unavoidable [525]. Nevertheless, once one pins down the measures for mixedness and entanglement, certain mixed states do pop out as special. These are “frontier” states in a scatter plots of where mixed states lie in an entanglement-mixedness plane.

Given the practical utility of maximally entangled pure states in ideal (i.e., noise-free) quantum information processing, it is possible these maximally entangled

mixed states would find similar application in more noisy quantum information processing, as they possess the maximum amount of entanglement possible for a given degree of mixedness.

Bill Munro and collaborators have identified a class of mixed states that deserve to be called maximally entangled as they lie on the frontier in the tangle (entanglement measure) versus linear-entropy (mixedness measure) plane. The structure of the density matrices corresponding to these maximally entangled mixed states is:

$$\rho_{\text{max-ent-mixed}} = \begin{cases} \begin{pmatrix} \frac{1}{3} & 0 & 0 & \frac{r}{2} \\ 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} & 0 \leq r \leq \frac{2}{3} \\ \begin{pmatrix} \frac{r}{2} & 0 & 0 & \frac{1}{3} \\ \frac{r}{2} & 0 & 0 & \frac{r}{2} \\ 0 & 1-r & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} & \frac{2}{3} \leq r \leq 1 \end{cases} \quad (11.74)$$

For a given value of the linear entropy (mixedness) these density matrices give the highest value of concurrence (entanglement). As tangle and entropy of formation are also both monotonic functions of concurrence, such density matrices also saturate the maximum possible degree of entanglement by these measures too.

11.3.7 The Schmidt Decomposition of a Pure Entangled State

Although we cannot write an entangled state of two quantum systems as the direct product of a state for each system we can, however, write it as a *sum* of such states. That is, if A is d_A -dimensional Hilbert space, and if B is d_B -dimensional Hilbert space, then for any entangled pure state $|\psi_{AB}\rangle$ in the Hilbert space of dimension $d_A \times d_B$, we can always find amplitudes such that:

$$|\psi_{AB}\rangle = \sum_{j=0}^{d_A-1} \sum_{k=0}^{d_B-1} a_{jk} |j_A\rangle \otimes |k_B\rangle \quad (11.75)$$

where $|j_A\rangle$ is a complete eigenbasis (i.e., set of orthonormal eigenvectors) for space A and $|k_B\rangle$ is a complete eigenbasis for space B . Notice, in particular, that to describe the state $|\psi_{AB}\rangle$ it is necessary to use a *double sum* over indices j and k .

The Schmidt decomposition, by contrast, allows us to re-express $|\psi_{AB}\rangle$ as a sum over a *single* index. And, moreover, the number of terms in this single sum is the lesser of d_A and d_B . This is rather counter-intuitive to most people when they first see this. Nevertheless, it is formally correct and allows you to interpret a given state in an interesting new way.

So how does the Schmidt decomposition work? Well it is actually rather simple. Everything hinges on using the *singular value decomposition* of a matrix built from the amplitudes that appear in the double sum description of $|\psi_{AB}\rangle$.

Schmidt Decomposition of a Pure State Given a (generally) entangled pure state of a composite quantum system, $|\psi_{AB}\rangle$, which can be thought of as composed of an n_A -qubit sub-system and an n_B -qubit sub-system, we can compute the Schmidt decomposition of $|\psi_{AB}\rangle$ as follows:

1. Sub-system A is in a $d_A = 2^{n_A}$ dimensional Hilbert space. Likewise, sub-system B is in a $d_B = 2^{n_B}$ dimensional Hilbert space. Let an eigenbasis for A be $\{|j_A\rangle\}_{j=0}^{d_A-1}$ and let an eigenbasis for B be $\{|k_B\rangle\}_{k=0}^{d_B-1}$.
2. Given the decomposition of $|\psi_{AB}\rangle$ in terms of the eigenbases $\{|j_A\rangle\}_{j=0}^{d_A-1}$ and $\{|k_B\rangle\}_{k=0}^{d_B-1}$ as:

$$|\psi_{AB}\rangle = \sum_{j=0}^{d_A-1} \sum_{k=0}^{d_B-1} a_{jk} |j_A\rangle \otimes |k_B\rangle \quad (11.76)$$

Re-group the amplitudes a_{jk} into a $d_A \times d_B$ dimensional array, and call this $\{a_{jk}\}$. That is, take the linear sequence of amplitudes a_{jk} and make a matrix by starting a new row after every d_B amplitudes.

3. Now compute the singular value decomposition (SVD) of the matrix $\{a_{jk}\}$ you just obtained. Specifically, we can write:

$$\text{SVD}(\{a_{jk}\}) = U \cdot \Sigma \cdot V = \{u_{ji}\} \cdot \{\sigma_{ii}\} \cdot \{v_{ik}\} \quad (11.77)$$

where $U = \{u_{ji}\}$ is a $d_A \times \min(d_A, d_B)$ dimensional unitary matrix, $V = \{v_{ik}\}$ is a $\min(d_A, d_B) \times d_B$ dimensional unitary matrix, and $\Sigma = \{\sigma_{ii}\}$ is a $\min(d_A, d_B) \times \min(d_A, d_B)$ diagonal matrix whose elements are the *singular values* of the matrix $\{a_{jk}\}$.

4. Now create new eigenbases as follows:

$$\{|i_A\rangle\}_{i=0}^{\min(d_A-1, d_B-1)} := \sum_{j=0}^{d_A-1} U_{j+1, i+1} |j_A\rangle \quad (11.78)$$

and

$$\{|i_B\rangle\}_{i=0}^{\min(d_A-1, d_B-1)} := \sum_{k=0}^{d_B-1} V_{i+1, k+1} |k_B\rangle \quad (11.79)$$

5. Pick out the subset of the singular values:

$$\{\lambda_i\}_{i=0}^{\min(d_A-1, d_B-1)} := \{\sigma_{ii}\}_{i=0}^{\min(d_A-1, d_B-1)} \quad (11.80)$$

6. Then the (generally entangled) pure state $|\psi_{AB}\rangle$ that is describable as the double sum in (11.75) is equally well describable as the *single* sum:

$$|\psi_{AB}\rangle = \sum_{i=0}^{\min(d_A-1, d_B-1)} \lambda_i |i_A\rangle |i_B\rangle \quad (11.81)$$

which is the Schmidt decomposition of $|\psi_{AB}\rangle$.

11.3.7.1 Example: Schmidt Decomposition

We illustrate the procedure for constructing the Schmidt decomposition using a simple 3-qubit pure state $|\psi_{AB}\rangle$. Here, we assume that A is a $d_A = 2$ dimensional sub-space and B is as $d_B = 4$ dimensional sub-space. To begin, we start with the state $|\psi_{AB}\rangle$ which we have defined to be:

$$\begin{aligned} |\psi_{AB}\rangle &= (-0.1661 - 0.17i)|0_A\rangle|00_B\rangle - (0.2982 + 0.0497i)|0_A\rangle|01_B\rangle \\ &\quad + (0.3471 + 0.3943i)|0_A\rangle|10_B\rangle - (0.2667 + 0.432i)|0_A\rangle|11_B\rangle \\ &\quad - (0.0293 + 0.2317i)|1_A\rangle|00_B\rangle + (0.1217 + 0.2168i)|1_A\rangle|01_B\rangle \\ &\quad + (0.2162 - 0.1238i)|1_A\rangle|10_B\rangle - (0.183 + 0.3263i)|1_A\rangle|11_B\rangle \end{aligned} \quad (11.82)$$

Here we see the eigenbasis for A is $\{|j_A\rangle\} \equiv \{|0_A\rangle, |1_A\rangle\}$ and that of B is $\{|k_B\rangle\} \equiv \{|00_B\rangle, |01_B\rangle, |10_B\rangle, |11_B\rangle\}$. Next we re-group the sequence of amplitudes appearing in (11.82) to form a new matrix $\{a_{jk}\}$. As $d_B = 4$ we start a new row of this matrix after every 4 (i.e., d_B) elements. This gives us the matrix:

$$\{a_{jk}\} = \begin{pmatrix} -0.1661 - 0.17i & -0.2982 - 0.0497i & 0.3471 + 0.3943i & -0.2667 - 0.432i \\ -0.0293 - 0.2317i & 0.1217 + 0.2168i & 0.2162 - 0.1238i & -0.183 - 0.3263i \end{pmatrix} \quad (11.83)$$

Next we compute the singular value decomposition $\text{SVD}(\{a_{jk}\})$ to give:

$$\text{SVD}(\{a_{jk}\}) = U \cdot \Sigma \cdot V \quad (11.84)$$

where

$$\begin{aligned} U &= \begin{pmatrix} 0.8876 & -0.4606 \\ 0.3806 - 0.2594i & 0.7334 - 0.4999i \end{pmatrix} \\ \Sigma &= \begin{pmatrix} 0.9031 & 0 \\ 0 & 0.4295 \end{pmatrix} \\ V &= \begin{pmatrix} -0.109 - 0.2732i & -0.3041 + 0.0775i & 0.4678 + 0.3975i & -0.2455 - 0.6147i \\ 0.3978 - 0.2475i & 0.2754 + 0.5651i & 0.1409 - 0.3826i & 0.3533 - 0.3069i \end{pmatrix} \end{aligned} \quad (11.85)$$

From the SVD we then construct the new bases, $\{|i_A\rangle\}$ and $\{|i_B\rangle\}$ (we use an overbar symbol to distinguish these bases from the earlier ones):

$$|\bar{0}_A\rangle := 0.8876|0_A\rangle + (0.3806 - 0.2594i)|1_A\rangle \quad (11.86)$$

$$|\bar{1}_A\rangle := -0.4606|0_A\rangle + (0.7334 - 0.4999i)|1_A\rangle \quad (11.87)$$

Likewise,

$$\begin{aligned} |\bar{0}_B\rangle &:= (-0.109 - 0.2732i)|00_B\rangle - (0.3041 - 0.0775i)|01_B\rangle \\ &\quad + (0.4678 + 0.3975i)|10_B\rangle - (0.2455 + 0.6147i)|11_B\rangle \end{aligned} \quad (11.88)$$

$$\begin{aligned} |\bar{1}_B\rangle &:= (0.3978 - 0.2475i)|00_B\rangle + (0.2754 + 0.5651i)|01_B\rangle \\ &\quad + (0.1409 - 0.3826i)|10_B\rangle + (0.3533 - 0.3069i)|11_B\rangle \end{aligned} \quad (11.89)$$

where here “ $|\bar{0}_B\rangle$ ” and “ $|\bar{1}_B\rangle$ ” represent 2-qubit states. Notice that we only need $\min(d_A, d_B)$ eigenvectors for each basis even though the dimensions of sub-space A is (in this example) less than that of sub-space B .

Finally, we pick out the Schmidt coefficients from the singular values to give:

$$\lambda_0 = 0.9031 \quad (11.90)$$

$$\lambda_1 = 0.4295 \quad (11.91)$$

Hence our Schmidt decomposition is predicted to be:

$$|\psi_{AB}\rangle = \lambda_0 |\bar{0}_A\rangle |\bar{0}_B\rangle + \lambda_1 |\bar{1}_A\rangle |\bar{1}_B\rangle \quad (11.92)$$

11.3.8 Entanglement Distillation

In most applications of quantum communications and distributed quantum computing it is necessary to establish noise-free maximally entangled pairs of particles, such as pure Bell states, between the ends of a quantum communications channel. Invariably, when one sends quantum particles down real channels those particles will be affected by noise sources in the channel. Thus, what starts off as pure maximally entangled states will not end up as pure maximally entangled states by the time they reach the ends of the channel. This usually causes a failure of the protocol for which the sharing of maximal entanglement was necessary.

The solution is to perform “entanglement distillation” whereby a few maximally entangled bi-partite pure states are obtained from a larger number of non-maximally entangled bi-partite states. Convention has it that if the original states are pure, the process is called “entanglement concentration”, whereas if they are mixed the process is called “entanglement purification”. Either way the principle is the same—one sacrifices some of the non-maximally entangled states in order to distill out a smaller number of maximally entangled ones. There has now been a great deal of research invested in entanglement distillation reflecting its importance as a quantum information processing primitive.

11.3.8.1 Distilling Entanglement from Pure States: Entanglement Concentration

In entanglement concentration we distill out several maximally entangled bi-partite pure states (e.g., states of the form $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$) from a larger number of non-maximally entangled bi-partite pure states (e.g. states of the form $\alpha|00\rangle + \beta|11\rangle$ with $|\alpha| \neq |\beta|$). Thus, entanglement concentration can also be thought of as a kind of error correction wherein several “buggy” Bell states are distilled into fewer perfect Bell states.

Entanglement concentration was first proposed by Charles Bennett, Herbert Bernstein, Sandu Popescu, and Benjamin Schumacher [51], but their scheme was

later improved upon by Phillip Kaye and Michele Mosca [268] and it is the latter version we describe here.

Suppose Alice and Bob share an entangled pair of qubits in the state:

$$|\Psi\rangle = a_0|00\rangle + a_1|01\rangle + a_2|10\rangle + a_3|11\rangle \quad (11.93)$$

where $\sum_{i=0}^3 |a_i|^2 = 1$. Using the Schmidt decomposition of Sect. 11.3.7, such a state can always be re-expressed in the form

$$|\Psi\rangle = \alpha|\psi_0\rangle|\phi_0\rangle + \beta|\psi_1\rangle|\phi_1\rangle \quad (11.94)$$

for positive reals α and β , a $\{|\psi_0\rangle, |\psi_1\rangle\}$ -basis for Alice and a $\{|\phi_0\rangle, |\phi_1\rangle\}$ -basis for Bob. Alice and Bob could simply agree by convention to call their bases $\{\overset{A}{|\psi_0\rangle}, \overset{A}{|\psi_1\rangle}\} \equiv \{|0\rangle, |1\rangle\}$ and $\{\overset{B}{|\phi_0\rangle}, \overset{B}{|\phi_1\rangle}\} \equiv \{|0\rangle, |1\rangle\}$. Thus, whatever the entangled state Alice and Bob share, we can think of it as a “buggy” Bell state:

$$|\Psi\rangle = \alpha \overset{A}{|0\rangle} \overset{B}{|0\rangle} + \beta \overset{A}{|1\rangle} \overset{B}{|1\rangle} \quad (11.95)$$

where the over set letters indicate whether we are talking about Alice’s qubit or Bob’s. If $|\alpha| = |\beta|$ we would be dealing with a maximally entangled state. But in general this is not the case. Yet it is the maximally entangled states we need routinely in quantum information protocols. So the question arises how do we distill out a few maximally entangled Bell states from a greater number of non-maximally entangled ones?

Let us imagine we begin with n of these buggy Bell pairs. Thus, our starting state can be written as $|\Psi\rangle = (\alpha \overset{A}{|0\rangle} \overset{B}{|0\rangle} + \beta \overset{A}{|1\rangle} \overset{B}{|1\rangle})^{\otimes n}$. Expanding the definition gives us a state in which Alice’s and Bob’s qubits are scrambled together. For example, if $n = 2$, $|\Psi\rangle$ is equal to:

$$\begin{aligned} |\Psi\rangle &= (\alpha \overset{A}{|0\rangle} \overset{B}{|0\rangle} + \beta \overset{A}{|1\rangle} \overset{B}{|1\rangle})^{\otimes 2} \\ &= \alpha^2 \overset{AB}{|00\rangle} \overset{AB}{|00\rangle} + \alpha\beta \overset{AB}{|00\rangle} \overset{AB}{|11\rangle} + \alpha\beta \overset{AB}{|11\rangle} \overset{AB}{|00\rangle} + \beta^2 \overset{AB}{|11\rangle} \overset{AB}{|11\rangle} \end{aligned} \quad (11.96)$$

However, it is apparent that a simple bit-permutation applied to the qubits will allow us to group Alice and Bob’s qubits separately. After this permutation of qubits we can see $|\Psi\rangle$ is equivalent to:

$$|\Psi\rangle = \alpha^2 \overset{AA}{|00\rangle} \overset{BB}{|00\rangle} + \alpha\beta \overset{AA}{|01\rangle} \overset{BB}{|01\rangle} + \alpha\beta \overset{AA}{|10\rangle} \overset{BB}{|10\rangle} + \beta^2 \overset{AA}{|11\rangle} \overset{BB}{|11\rangle} \quad (11.97)$$

Generalizing, the state we obtain with n non-maximally entangled states is:

$$|\Psi\rangle = \sum_{j=0}^n \alpha^{n-j} \beta^j \left(\sum_{\text{HammingWeight}(\mathbf{x})=j} |\overset{AA}{\mathbf{x}}\rangle \overset{BB}{\mathbf{x}}\rangle \right) \quad (11.98)$$

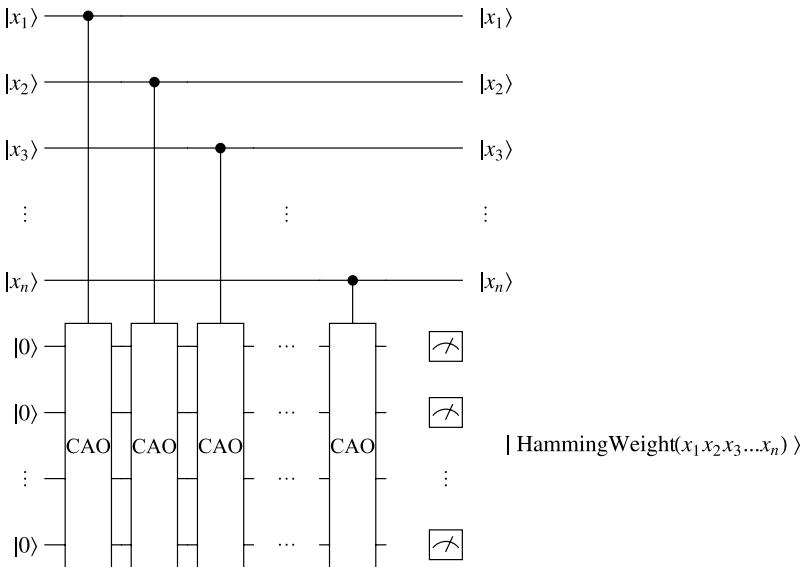


Fig. 11.4 Quantum circuit for measuring the Hamming weight of string of qubits. The quantum state whose Hamming weight we want is in the upper register, and a set of n ancillae qubits is in the lower register. Each qubit that is set to $|1\rangle$ in the upper register adds 1 to the Hamming weight via a controlled-add-one gate. However, each qubit set to $|0\rangle$ adds nothing to the Hamming weight. By initializing the ancillae qubits to $|00\dots 0\rangle$ we accumulate the Hamming weight in the lower register. If the upper register is a superposition state that has eigenstates of different Hamming weights (as we intend it to be) the act of measuring the Hamming weight projects the state of the upper register into a superposition of only those eigenstates consistent with the observed Hamming weight

Now suppose Alice measures the Hamming weight of $|\Psi\rangle$, i.e., she determines how many of her qubits are in state $|1\rangle$. By the structure of the state $|\Psi\rangle$ Bob would be guaranteed to obtain the same result if he were to measure the Hamming weight of his qubits. To measure the Hamming weight, Alice can use a quantum circuit like that shown in Fig. 11.4. This circuit consists of two registers: an upper n -qubit register holding the superposition $|\Psi\rangle$ and a lower n -qubit register holding n ancillae prepared initially in state $|00\dots 0\rangle$. These registers are connected via a cascade of controlled-add-one gates. If the i -th qubit in the upper register is set to $|1\rangle$ it adds 1 to the Hamming weight and 0 otherwise. Via the linearity of quantum mechanics, the circuit produces a superposition of Hamming weights in the lower register. When the Hamming weight in the lower register is measured the upper register is projected into a state whose component eigenstates then have the Hamming weight that was measured in the lower register. Thus, if Alice and Bob each measure the Hamming weight to be $|j\rangle$ they will project the upper register into a state of the form:

$$\frac{1}{\sqrt{\binom{n}{j}}} \sum_{\text{HammingWeight}(\mathbf{x})=j}^{\substack{AA\dots ABB\dots B \\ |\mathbf{x}\rangle \quad |\mathbf{x}\rangle}} \quad (11.99)$$

which is a superposition of $\binom{n}{j}$ bit strings.

Next define a function that maps each of these bits strings (arranged in lexicographic order) into a unique integer from 0 to $\binom{n}{j} - 1$. Specifically, we have:

$$\begin{aligned} f(00\dots 0\underbrace{011\dots 1}_j) &\rightarrow (0)_{10} = 00\dots 0 \\ f(00\dots 0101\underbrace{\dots 1}_{j-1}) &\rightarrow (1)_{10} = 00\dots 1 \\ &\vdots \\ f(\underbrace{11\dots 1}_{j}00\dots 0) &\rightarrow ((\binom{n}{j} - 1)_{10} \end{aligned} \quad (11.100)$$

For example, if there are $n = 6$ qubits with Hamming weight 4, the mapping f would be:

$$\begin{aligned} f(0, 0, 1, 1, 1, 1) &\rightarrow f(15) \rightarrow 0 \\ f(0, 1, 0, 1, 1, 1) &\rightarrow f(23) \rightarrow 1 \\ f(0, 1, 1, 0, 1, 1) &\rightarrow f(27) \rightarrow 2 \\ f(0, 1, 1, 1, 0, 1) &\rightarrow f(29) \rightarrow 3 \\ f(0, 1, 1, 1, 1, 0) &\rightarrow f(30) \rightarrow 4 \\ f(1, 0, 0, 1, 1, 1) &\rightarrow f(39) \rightarrow 5 \\ f(1, 0, 1, 0, 1, 1) &\rightarrow f(43) \rightarrow 6 \\ f(1, 0, 1, 1, 0, 1) &\rightarrow f(45) \rightarrow 7 \\ f(1, 0, 1, 1, 1, 0) &\rightarrow f(46) \rightarrow 8 \\ f(1, 1, 0, 0, 1, 1) &\rightarrow f(51) \rightarrow 9 \\ f(1, 1, 0, 1, 0, 1) &\rightarrow f(53) \rightarrow 10 \\ f(1, 1, 0, 1, 1, 0) &\rightarrow f(54) \rightarrow 11 \\ f(1, 1, 1, 0, 0, 1) &\rightarrow f(57) \rightarrow 12 \\ f(1, 1, 1, 0, 1, 0) &\rightarrow f(58) \rightarrow 13 \\ f(1, 1, 1, 1, 0, 0) &\rightarrow f(60) \rightarrow 14 \end{aligned} \quad (11.101)$$

We further extend f in any way that keeps it reversible and hence implementable as a permutation matrix, i.e. we extend the definition of f so that it maps each of the other bit strings (which do not have Hamming weight j) to unique indices too.

If we define $r = \lceil \log_2 \binom{n}{j} \rceil$ we can write:

$$\begin{aligned}
& \frac{1}{\sqrt{\binom{n}{j}}} \sum_{\text{HammingWeight}(x)=j}^{} \begin{array}{c} AA \dots AB \\ |\mathbf{x}\rangle \quad |\mathbf{x}\rangle \end{array} \\
& \xrightarrow{f} \frac{1}{\sqrt{\binom{n}{j}}} \sum_{\text{HammingWeight}(x)=j}^{} \begin{array}{c} AA \dots A \quad BB \dots B \\ |f(\mathbf{x})\rangle \quad |f(\mathbf{x})\rangle \end{array} \\
& = \frac{1}{\sqrt{\binom{n}{j}}} \sum_{y=0}^{\binom{n}{j}-1} \begin{array}{c} AA \dots A \\ |\underbrace{\mathbf{0}}_{n-r}\rangle \end{array} \begin{array}{c} AA \dots A \\ |\underbrace{\mathbf{y}}_r\rangle \end{array} \begin{array}{c} BB \dots B \\ |\underbrace{\mathbf{0}}_{n-r}\rangle \end{array} \begin{array}{c} BB \dots B \\ |\underbrace{\mathbf{y}}_r\rangle \end{array} \quad (11.102)
\end{aligned}$$

If $\binom{n}{j} = 2^r$, ignoring the first $n - r$ qubits in each register then gives:

$$\frac{1}{\sqrt{2^r}} \sum_{y=0}^{2^r-1} \begin{array}{c} AA \dots A \\ |\underbrace{\mathbf{y}}_r\rangle \end{array} \begin{array}{c} BB \dots B \\ |\underbrace{\mathbf{y}}_r\rangle \end{array} \quad (11.103)$$

which as before can, under a permutation of the qubits, be recognized as r pristine Bell state pairs, and the entanglement in $|\Psi\rangle$ has been concentrated.

Of course, in general $\binom{n}{j} \neq 2^r$. In this case, one can still distill out some perfect Bell state pairs, but their number is not certain a priori. See [268] for details.

Thus, to sum up, the entanglement concentration procedure can be described as follows:

Entanglement Concentration

1. Alice and Bob start off with n copies of a non-maximally entangled state $|\Psi\rangle = (\alpha|00\rangle + \beta|11\rangle)^{\otimes n}$ with $\alpha \neq \beta$, and they each hold one member of each non-maximally entangled pair.
2. Next Alice and Bob perform a qubit-permutation sufficient to group all Alice's qubits together and all Bob's qubits together, creating a state of the form

$$|\Psi\rangle = \sum_{j=0}^n \alpha^{n-j} \beta^j \left(\sum_{\text{HammingWeight}(\mathbf{x})=j}^{} \begin{array}{c} AA \dots A \\ |\mathbf{x}\rangle \end{array} \begin{array}{c} BB \dots B \\ |\mathbf{x}\rangle \end{array} \right) \quad (11.104)$$

3. Alice and Bob each measure the Hamming weight of their set of particles, i.e., they each determine how many of their qubits are in state $|1\rangle$. Given the structure of the state, their results will always agree. If they each determine the Hamming weight is $|j\rangle$, this measurement operation has the effect of projecting Alice and

Bob's joint state into the form

$$\frac{1}{\sqrt{\binom{n}{j}}} \sum_{\text{HammingWeight}(\mathbf{x})=j}^{\text{AA...ABBB...B}} |\mathbf{x}\rangle |\mathbf{x}\rangle \quad (11.105)$$

where the labels “A” and “B” specify whether the qubits are in Alice’s possession or Bob’s possession.

4. Alice and Bob each apply the transformation f to the state obtained in the last step. In the simplest case when $\binom{n}{j} = 2^r$ ignoring the first $n - r$ qubits gives the state $\frac{1}{\sqrt{2^r}} \sum_{y=0}^{2^r-1} |\overset{\text{AA...A}}{\mathbf{y}}\rangle |\overset{\text{BB...B}}{\mathbf{y}}\rangle$.
5. By inverting the qubit permutation performed at step (2) above, this state becomes that of r perfect Bell pairs.
6. The procedure can be extended to deal with the case $\binom{n}{j} \neq 2^r$, and a quantum circuit can be defined which allows the number of perfect Bell pairs distilled out to be measured (see [268]). The expected number of pairs obtainable when $\binom{n}{j} \neq 2^r$ is at least $\sum_{j=0}^n |\alpha|^2 |n-j| (1 - |\alpha|^2)^j \binom{n}{j} (\lfloor \log_2 \binom{n}{j} \rfloor - 1)$.

Entanglement concentration is of practical importance in many quantum communications protocols as well as in distributed quantum computing (see Sect. 15.2). It can be extended to the case of distilling bi-partite maximally entangled pure states from non-maximally entangled mixed states, and is then known as *entanglement purification* [53, 54, 139]. This is distinct from the concept of the purification of a mixed state discussed in Sect. 11.2.4 whereby a mixed state, ρ_B , is re-cast as the partial trace of a pure state, $|\psi_{AB}\rangle$, in a higher dimensional Hilbert space, i.e., state purification finds the $|\psi_{AB}\rangle$ such that $\rho_B = \text{tr}_A(|\psi_{AB}\rangle\langle\psi_{AB}|)$. By contrast, in entanglement purification we distill out a set of maximally entangled bi-partite pure states from a larger number of non-maximally entangled bi-partite mixed states.

11.3.9 Entanglement Swapping

Thus far, the schemes we have looked at for creating entanglement have all worked by causing pairs of initially unentangled qubits to interact *directly* and then separating them spatially. However, it is also possible to entangle two particles that have *never* interacted directly. The trick is to start with two maximally entangled pairs of particles, and to arrange for one member of each pair to be measured in a Bell basis using a device known as a “Bell State Analyzer” (BSA). This sounds fancy, and experimentally it is challenging to build one, but theoretically speaking it requires nothing more than the Bell state synthesizer circuit run in reverse followed by single qubit measurements in the computational basis. The net effect is that we can swap

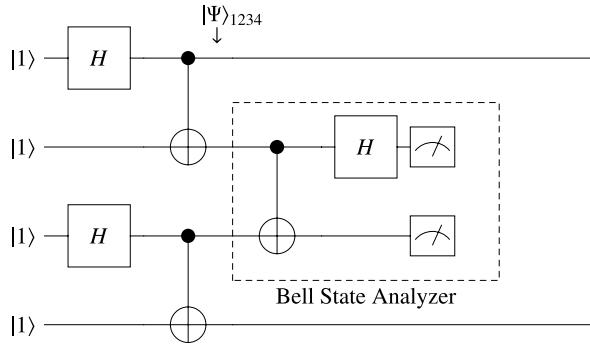


Fig. 11.5 Entanglement swapping provides a means to entangle two parties that have never interacted with one another directly. Alice and Bob each prepare a maximally entangled pair of particles. They each retain one of these particles and pass the other to Cerys. Cerys performs a complete Bell state analysis on the two particles she received, which results in classifying them as being in one of the four Bell states $|\beta_{00}\rangle$, $|\beta_{01}\rangle$, $|\beta_{10}\rangle$, $|\beta_{11}\rangle$. Thereafter, the two particles that remain in Alice and Bob's possession will be maximally entangled in some Bell state the identity of which depends on result of the Bell state analysis. Entanglement swapping is a key ingredient of quantum repeaters, distributed quantum computing, and heralded entangled photon sources

initial entanglement between particles 1 and 2 and initial entanglement between particles 3 and 4 into entanglement between particles 1 and 4, even though particles 1 and 4 never interacted directly. The procedure that does this is therefore called *entanglement swapping* and was originally conceived of by Marek Zukowski, Anton Zeilinger, Michael Horne, and Artur Ekert in 1993 [565]. The scheme is illustrated in Fig. 11.5.

Entanglement swapping works as follows: Alice and Bob each prepare matching maximally entangled pairs of particles. For example, they may each prepare their own Bell singlet pair $|\beta_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$. Such states can be prepared by feeding a $|11\rangle$ state into a Bell state synthesizer circuit, which consists of a single Walsh-Hadamard gate followed by a CNOT gate. Let us say that Alice starts off in possession of qubits 1 and 2, and Bob starts off in possession of qubits 3 and 4. Using these particle labels as subscripts to avoid ambiguity, the input to the entanglement swapping circuit is therefore the state $|11\rangle_{12} \otimes |11\rangle_{34} = |1111\rangle_{1234}$. Upon applying the double Bell state synthesizer circuits as show in Fig. 11.5, the following transformation occurs:

$$\begin{aligned} |\Psi\rangle_{1234} &= (\text{CNOT} \otimes \text{CNOT}) \cdot (H \otimes \mathbb{1} \otimes H \otimes \mathbb{1}) |11\rangle_{12} |11\rangle_{34} \\ &= \left(\frac{1}{\sqrt{2}}(|01\rangle_{12} - |10\rangle_{12}) \right) \otimes \left(\frac{1}{\sqrt{2}}(|01\rangle_{34} - |10\rangle_{34}) \right) \end{aligned} \quad (11.106)$$

However, $|\psi\rangle_{1234}$ can also be re-expressed in the Bell basis by imagining the qubits to be permuted as follows. Swap qubits 2 and 4 (to take the qubit ordering 1234 into 1432) and then swap qubits 3 and 2 (to take the qubit ordering 1432 into 1423). An operator sufficient to perform qubit permutation is $(\mathbb{1}_4 \otimes \text{SWAP}) \cdot (\mathbb{1}_2 \otimes \text{SWAP} \otimes$

$\mathbb{1}_2 \cdot (\mathbb{1}_4 \otimes \text{SWAP}) \cdot (\mathbb{1}_2 \otimes \text{SWAP} \otimes \mathbb{1}_2)$. In the “1423” basis, we can write $|\psi\rangle_{1234}$ as the equivalent $|\psi\rangle_{1423}$ where:

$$|\psi_{1423}\rangle = \frac{1}{2}(-|\beta_{00}\rangle_{14}|\beta_{00}\rangle_{23} + |\beta_{01}\rangle_{14}|\beta_{01}\rangle_{23} + |\beta_{10}\rangle_{14}|\beta_{10}\rangle_{23} - |\beta_{11}\rangle_{14}|\beta_{11}\rangle_{23}) \quad (11.107)$$

Hence, in this Bell basis representation, we can see immediately that if we perform a complete Bell state analysis of qubits 2 and 3 (i.e., if we figure out which Bell state they are in), then qubits 1 and 4 will then be projected into the identical Bell state (up to an overall phase factor), even though qubits 1 and 4 had, at no time, interacted directly.

To perform a complete Bell-basis measurement we need only *invert* the operation that synthesizes the Bell states starting from the computational basis states and then measure the result in the computational basis. In terms of a quantum circuit such an inversion is achieved by reversing the order of the gates and using the inverse (or, since they are unitary, the conjugate transpose) of each gate operator. Therefore, as the Bell state synthesizer is the operator, $\text{CNOT} \cdot (H \otimes \mathbb{1})$, the complete Bell state analyzer is the operator $(H \otimes \mathbb{1})^\dagger \cdot \text{CNOT}^\dagger = (H \otimes \mathbb{1}) \cdot \text{CNOT}$ (as shown in the dashed box in Fig. 11.5). So defined, the Bell state analyzer accepts a Bell state and returns $|00\rangle$, $|01\rangle$, $|10\rangle$, or $|11\rangle$, according to whether the input Bell state was $|\beta_{00}\rangle$, $|\beta_{01}\rangle$, $|\beta_{10}\rangle$, or $|\beta_{11}\rangle$. A *complete* Bell state analyzer has been demonstrated experimentally by Yoon-Ho Kim, Sergei Kulik, and Yanhua Shih in 2001 [278].

Entanglement swapping is a very useful trick in quantum information science. It is a crucial building block in quantum repeaters (used to extend the range of quantum key distribution in optical fibers) [103, 160, 295, 428], in distributed quantum computing [552], and as a means to have a heralded source of entangled photon pairs [565].

11.3.10 Entanglement in “Warm” Bulk Matter

One of the most exciting developments in our understanding of entanglement in recent years has come from the realization that entanglement can persist in macroscopic amounts of matter at room temperature. This came as a complete surprise. Just a few years ago creating and sustaining entangled states of even a handful of quantum particles required exquisitely delicate experiments, and ideal laboratory conditions. Indeed, great suspicion fell on anyone suggesting that entanglement might play a role in the brain and biological structures mainly on the grounds that they were too warm and noisy to sustain such effects. However, old-school thinking about entanglement should no longer be taken as conclusive.

We now know that entanglement can be found in macroscopic systems [19], even relatively warm ones [513], and in fact plays an *essential* role in determining how such matter behaves, such as the anomalously low magnetic susceptibility of certain magnetic systems [85]. This is quite extraordinary. Such developments are very

exciting because they could mark the beginning in an entirely new direction for materials science and solid state physics. Who knows what miracle materials await discovery if entanglement can persist and play a role in shaping their properties at temperatures well above absolute zero.

Similarly, other studies have provided evidence for the existence of quantum effects in certain biological structures. For example, quantum transport is believed to occur in the Fenna-Matthews-Olson (FMO) light harvesting complex of purple bacteria [172, 306]. At low temperatures the excitons created after photon absorption are found to propagate through the FMO complex *coherently*, and in fact, their transport is *enhanced* by the presence of a small amount of noise, perhaps allowing the phenomenon to persist up to biologically relevant temperatures. Likewise, it has been hypothesized that magnetoreception in birds works by interconverting singlet/triplet excited states of the cryptochrome protein [256]. And a recent model of olfaction replaces the standard shape-based theory with the notion that phonon-assisted tunneling is used to sense the vibrational spectra of odorant molecules [500]. All these results are intriguing and may point to more sophisticated ways of harnessing quantum effects in structures that are at relatively high temperatures.

11.4 Compressing Quantum Information

In classical information theory we describe messages as sequences of symbols drawn from some finite alphabet, such that each symbol may appear with a different probability. The obvious quantum analog of this is to treat a source of quantum information as a device that generates a sequence of quantum states, each potentially with some different probability. Thus, quantum states that are known only as some statistical mixture of pure states arise naturally when we extend information theory into the quantum realm.

Whereas Shannon information theory regards a classical source as a device that generates a sequence of classical symbols (i.e., distinguishable states) picked from a finite alphabet according to different probabilities, quantum information theory regards a quantum source as a device that generates a sequence of quantum symbols (i.e., not necessarily distinguishable states) picked from a finite alphabet according to different probabilities. Thus, we find ourselves having to model quantum states that are only specified in exactly the statistical sense mentioned above. Hence, the introduction of density operators is absolutely necessary.

If the quantum states used for the alphabet of symbols are all orthogonal to one another then, in principle, we can measure them without disturbing them, and hence to all intents and purposes they are essentially just classical symbols in disguise. Hence, we would could characterize such a source in terms of its Shannon entropy. In particular, if a source produces a stream of orthogonal (i.e., unambiguously distinguishable) states in which the i -th state occurs with probability p_i the source is characterized by its Shannon entropy $H(\{p_i\}) = - \sum_i p_i \log_2 p_i$.

The situation becomes more interesting if we assume that the quantum states encoding the symbols are not necessarily all orthogonal to one another.

11.4.1 Quantum Sources: The von Neumann Entropy

Let us imagine we have a device for outputting one of d not necessarily orthogonal quantum states at random. In particular, let the device output state $|\psi_i\rangle$ with corresponding probability p_i . The density operator for such a source would be:

$$\rho = \sum_{i=1}^d p_i |\psi_i\rangle\langle\psi_i| \quad (11.108)$$

and we can characterize its entropy using the techniques introduced in Sect. 11.2.5.2. There we saw that the entropy of a quantum source can be related to the Shannon entropy of a corresponding fictitious classical source. Specifically, the von Neumann entropy of a quantum source having density operator ρ is defined via its representation in a diagonal basis as:

$$S_V(\rho) = - \sum_i \lambda_i \log_2 \lambda_i = -\text{tr}(\rho \log_2 \rho) \quad (11.109)$$

As expected, the von Neumann entropy so-defined then equals to the Shannon entropy when the quantum states emitted by the source are unambiguously distinguishable.

The von Neumann entropy has many interesting uses and properties. For example, if ρ is a pure state, $S_V(\rho) = 0$. Hence, the von Neumann entropy can be used to decide whether or not a given density operator corresponds to that of a pure state. In addition, the von Neumann entropy of a state does not change under unitary evolution, i.e., $S_V(\rho) = S_V(U\rho U^\dagger)$, because the von Neumann entropy only depends upon the eigenvalues and these are not changed under unitary evolution. These and other important properties of the von Neumann entropy are summarized in Table 11.3

In analogy to the Shannon noiseless coding theorem wherein the n bit classical messages from a classical source with Shannon entropy $H(\{p_i\})$ can be compressed into at most $nH(\{p_i\})$ classical bits, n qubit quantum messages from a quantum source with von Neumann entropy $S_V(\rho)$ can be compressed into at most $nS_V(\rho)$ qubits. However, this tells us nothing about how to accomplish the compression. That is the domain of quantum data compression.

11.4.2 Schumacher-Jozsa Quantum Data Compression

Suppose Alice chooses real numbers α and β such that $\alpha^2 + \beta^2 = 1$, and creates a quantum message consisting of sequences of the states $|\psi_+\rangle$ and $|\psi_-\rangle$ defined as:

$$|\psi_+\rangle = \alpha|0\rangle + \beta|1\rangle \quad (11.110)$$

$$|\psi_-\rangle = \alpha|0\rangle - \beta|1\rangle \quad (11.111)$$

Table 11.3 Properties of the von Neumann entropy

Property	Formula	Condition
Purity	$S_V(\rho) = 0$	If ρ is a pure state, i.e., $\rho = \psi\rangle\langle\psi $
Invariance	$S_V(\rho) = S_V(U\rho U^\dagger)$	If U is a unitary transformation
Maximum	$S_V(\rho) \leq \log_2 k$	If ρ has k non-zero eigenvalues. Equality holds when all the non-zero eigenvalues are equal
Concavity	$S_V(\sum_i p_i \rho_i) \geq \sum_i p_i S_V(\rho_i)$	Provided $p_i \geq 0$ and $\sum_i p_i = 1$. This result shows that the less we know about how a state is prepared the greater its von Neumann entropy
Boundedness	$S_V(\rho) \leq H(\{p_i\})$	For an ensemble of quantum states $ \psi_i\rangle$ occurring with probabilities p_i , and having density operator $\rho = \sum_i p_i \psi_i\rangle\langle\psi_i $ its von Neumann entropy is never greater than the Shannon entropy of the corresponding classical ensemble. Equality holds when all the quantum states are orthogonal and hence unambiguously distinguishable
Subadditivity	$S_V(\rho_{AB}) \leq S_V(\rho_A) + S_V(\rho_B)$	Equality holds when $\rho_{AB} = \rho_A \otimes \rho_B$. That is, the von Neumann entropies of independent systems add, but will be lowered if they are correlated
Strong subadditivity	$S_V(\rho_{ABC}) + S_V(\rho_B) \leq S_V(\rho_{AB}) + S_V(\rho_{BC})$	For two systems AB and BC having a common subsystem B the sum of the von Neumann entropies of their union and intersection is less than the sum of their von Neumann entropies
Araki-Lieb inequality	$S_V(\rho_{AB}) \geq S_V(\rho_A) - S_V(\rho_B) $	A bipartite state ρ_{AB} can be completely known (zero entropy) even though its parts are not, such as when $S_V(\rho_A) = S_V(\rho_B) \neq 0$

The overlap $\langle\psi_+|\psi_-\rangle = 2\alpha^2 - 1$ is non-zero, and hence $|\psi_+\rangle$ and $|\psi_-\rangle$ are non-orthogonal, for most values of α . This means that the quantum ‘‘symbols’’ in Alice’s message are not entirely distinguishable for most choices of α . There is the potential, therefore, for some added redundancy in messages encoded using (non-orthogonal) quantum symbols that is not present in messages encoded using (orthogonal) classical symbols. Ultimately, this is what allows quantum messages to be compressed beyond the Shannon bound.

If the two states, $|\psi_+\rangle$ and $|\psi_-\rangle$, appear with equal probability, the von Neumann entropy of Alice’s source is:

$$S(\rho) = -\alpha^2 \log_2 \alpha^2 - \beta^2 \log_2 \beta^2 \quad (11.112)$$

Thus, if the states are orthogonal (which occurs when $\alpha^2 = \beta^2 = \frac{1}{2}$) the von Neumann entropy of the source reduces to exactly the Shannon entropy.

As shown by Mitsumori et al. [356] we can compress our quantum message in blocks of three qubits at a time.

$$|B_{\mathbf{L}}\rangle = |\psi_{L_1}\rangle \otimes |\psi_{L_2}\rangle \otimes |\psi_{L_3}\rangle \quad (11.113)$$

$\mathbf{L} = (L_1, L_2, L_3)$ and $L_i \in \{+, -\}$.

Index L corresponds to one of eight possible configurations for the 3-qubit block, namely $|\psi_+\rangle|\psi_+\rangle|\psi_+\rangle$, $|\psi_+\rangle|\psi_+\rangle|\psi_-\rangle$, $|\psi_+\rangle|\psi_-\rangle|\psi_+\rangle$, ..., $|\psi_-\rangle|\psi_-\rangle|\psi_-\rangle$.

Alice applies the “compressor” operation, U , which is defined via its action on computational basis states as follows:

$$\begin{aligned} U := & |000\rangle \rightarrow |000\rangle \\ & |001\rangle \rightarrow |001\rangle \\ & |010\rangle \rightarrow |010\rangle \\ & |011\rangle \rightarrow |100\rangle \\ & |100\rangle \rightarrow |011\rangle \\ & |101\rangle \rightarrow |101\rangle \\ & |110\rangle \rightarrow |110\rangle \\ & |111\rangle \rightarrow |111\rangle \end{aligned}$$

The state of a block of three qubits after this compressor has been applied is as follows:

$$U|B_{\mathbf{L}}\rangle = \alpha^2 \sqrt{1 + 2\beta^2} |0\rangle |\mu_{\mathbf{L}}\rangle + \beta^2 \sqrt{1 + 2\alpha^2} |\nu_{\mathbf{L}}\rangle \quad (11.115)$$

where

$$|\mu_{\mathbf{L}}\rangle = \frac{1}{1 + 2\beta^2} (\alpha|00\rangle + \beta_1|11\rangle + \beta_2|10\rangle + \beta_3|01\rangle) \quad (11.116)$$

$$|\nu_{\mathbf{L}}\rangle = \frac{1}{\beta^2 \sqrt{1 + 2\alpha^2}} [\alpha(\beta_1\beta_2|10\rangle + \beta_1\beta_3|01\rangle + \beta_2\beta_3|00\rangle) + \beta_1\beta_2\beta_3|11\rangle] \quad (11.117)$$

where $\beta_i = L_i \beta$ which will either be $+\beta$ or $-\beta$.

Next Alice measures the first qubit of the compressed state in the computational basis, to obtain the value $|0\rangle$ or $|1\rangle$ [356]. What happens next depends on whether Alice wants to pursue a “Discard-on-Fail” (see Fig. 11.6) or an “Augment-on-Fail” (see Fig. 11.7) quantum data compression protocol. Let us take a look at each of these protocols in turn.

11.4.3 “Discard-on-Fail” Quantum Data Compression Protocol

Discard-on-Fail Quantum Data Compression

- Partition the data in blocks of three qubits at a time, apply the compressor, U , to each block, i.e., $|B_{\mathbf{L}}\rangle \rightarrow U|B_{\mathbf{L}}\rangle$.

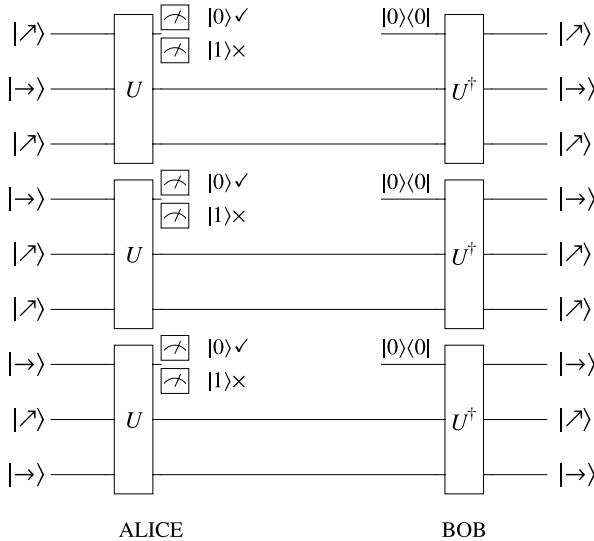


Fig. 11.6 First quantum data compression protocol. Alice encodes a sequence of non-orthogonal qubits in blocks of three qubits using the compressor U . She reads the first qubit obtaining the result $|0\rangle$ or $|1\rangle$. If Alice obtains $|0\rangle$ she will have prepared the second and third qubits in the same block in the state $|\mu_L\rangle$, and she sends these qubits to Bob. Upon receipt, Bob augments these qubits with a new first qubit prepared in state $|0\rangle$ and sends all three qubits through the decompressor U^\dagger . The output triplet of qubits is now restored close to their original values even though only two qubits (rather than three) passed through the channel between Alice and Bob. If, instead, when Alice had measured the first qubit Alice she had found it in state $|1\rangle$ she would have regarded this as a “failure” and would have sent nothing to Bob

2. Alice measures the first qubit in each block output from the compressor, and obtains $|0\rangle$ or $|1\rangle$.
3. If Alice obtains $|0\rangle$ she retains the measured qubit and passes the remaining two qubits, now in state $\rho_L^{(1)} = |\mu_L\rangle\langle\mu_L|$, to Bob. This event occurs with probability $p = \alpha^4(1 + 2\beta^2)$. If Alice obtains $|1\rangle$ she regards this as a “failure” and sends nothing to Bob. This event, which Bob sees a data drop out in the stream from Alice, occurs with probability $1 - p$.
4. If Bob does receive qubits from Alice, he prepares a new qubit in the state $|0\rangle\langle 0|$ to create the state $(|0\rangle\langle 0| \otimes \rho_L^{(1)})$, and then feeds this expanded state into the 3-qubit decompressor, U^\dagger . This operation produces the state

$$\Phi_L^{(1)} = U^\dagger (|0\rangle\langle 0| \otimes \rho_L^{(1)}) U \quad (11.118)$$

5. The result is that for each block, Bob either receives nothing from Alice or a pair of qubits which he can expand and decompress. Hence, the fidelity of the overall quantum data compression process is

$$F^{(1)} = \sum_L \frac{1}{8} \langle B_L | \Phi_L^{(1)} | B_L \rangle = \alpha^8(1 + 2\beta^2)^2 \quad (11.119)$$

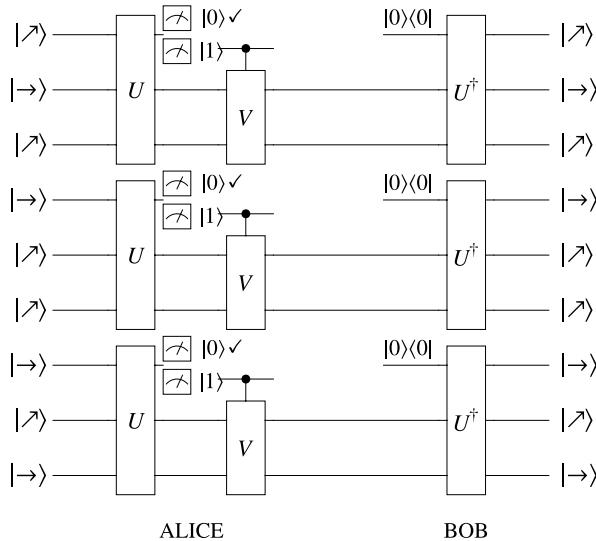


Fig. 11.7 Second quantum data compression protocol. Alice encodes a sequence of non-orthogonal qubits in blocks of three qubits using the compressor U . She reads the first qubit obtaining the result $|0\rangle$ or $|1\rangle$. If Alice obtains $|0\rangle$ she will have prepared the second and third qubits in the same block in the state $|\mu_L\rangle$, and she sends these qubits to Bob. If, however, Alice had obtained $|1\rangle$ when she had measured the first qubit, she would have modified the state of the second and third qubits before passing them to Bob. Upon receipt, Bob augments the qubits transmitted from Alice with a new first qubit prepared in state $|0\rangle$ and sends all three qubits through the decompressor U^\dagger . The output triplet of qubits is now restored close to their original values even though only two qubits (rather than three) passed through the channel between Alice and Bob

11.4.4 “Augment-on-Fail” Quantum Data Compression Protocol

Augment-on-Fail Quantum Data Compression

1. Partition the data in blocks of three qubits at a time, apply the compressor, U , to each block, i.e., $|B_L\rangle \rightarrow U|B_L\rangle$.
2. Alice measures the first qubit in each block output from the compressor, and obtains $|0\rangle$ or $|1\rangle$.
3. If Alice obtains $|0\rangle$ she retains the measured qubit and passes the remaining two qubits to Bob. If Alice obtains $|1\rangle$ she applies a unitary operation V to the two unmeasured qubits and then sends them to Bob.
4. When Bob receives a pair of qubits from Alice, he prepares a new qubit in the state $|0\rangle$ to create the state $|0\rangle|\mu_L\rangle$, and then feeds this expanded state into the 3-qubit decompressor, U^\dagger .
5. The result is that for each block, Bob either receives nothing from Alice or a pair of qubits which he can expand and decompress.

The fidelity of the “augment-on-fail” quantum data compression protocol exceeds that of the “discard-on-fail” quantum data compression protocol. However,

the “augment-on-fail” protocol is more challenging to implement in physical hardware due to the conditional correction that Alice must apply to the unmeasured qubits in each block prior to their transmission to Bob.

11.4.5 Quantum Circuit for Schumacher-Jozsa Compressor

The final step in understanding quantum data compression is to construct explicit quantum circuits for the compressor, U , and the decompressor, U^\dagger .

First, we can make our life easier by recognizing that once we know an efficient quantum circuit for U we know an efficient quantum circuit for U^\dagger too. To see this, consider a unitary matrix, U , which can be factored in terms of a dot product of unitary matrices A and B i.e., $U = A \cdot B$. This implies that the unitary matrix U^\dagger can be factored as $U^\dagger = U^{-1} = (A \cdot B)^{-1} = B^{-1} \cdot A^{-1} = B^\dagger \cdot A^\dagger$. Thus given a quantum circuit for the compressor, U , we can obtain a quantum circuit for the decompressor, U^\dagger , by inverting and reversing the gates in the quantum circuit for U . Hence, we need only find a quantum circuit for just the compressor U .

In order to realize the truth table (i.e., basis transformation) we want U to have, the matrix for U must take the form:

$$U := \begin{pmatrix} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ 000 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 001 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 010 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 011 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 100 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 101 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 110 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 111 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (11.120)$$

which is a permutation matrix, similar to that of a TOFFOLI gate:

$$\text{TOFFOLI:}= \begin{pmatrix} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ 000 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 001 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 010 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 011 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 100 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 101 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 110 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 111 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (11.121)$$

except that the “NOT” part is shifted up the diagonal. This suggests that we can obtain U from TOFFOLI by shifting the “NOT” part using the permutation matrix Q :

$$Q := \begin{pmatrix} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ 000 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 001 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 010 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 011 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 100 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 101 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 110 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 111 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (11.122)$$

applied three times to TOFFOLI. Thus we obtain our first clue about how to construct U from the factorization:

$$U = Q \cdot Q \cdot Q \cdot \text{TOFFOLI} \cdot Q^\dagger \cdot Q^\dagger \cdot Q^\dagger \quad (11.123)$$

Next, we need to reduce TOFFOLI and Q to simpler forms. The TOFFOLI gate is well-studied and we already know of an efficient quantum circuit for implementing TOFFOLI using 1-qubit gates and CNOT gates (see Sect. 2.5.7). However, the Q gate is a new (and pretty useful) gate in the toolbox of the quantum circuit designer. So how do we factor Q into more familiar quantum gates?

The trick is to realize that for the general n -qubit case:

$$Q_{2^n} = \text{QFT}_{2^n}^{-1} \cdot T_{2^n} \cdot \text{QFT}_{2^n} \quad (11.124)$$

where T_{2^n} is defined by:

$$T_{2^n} = \bigotimes_{k=n-1}^0 \begin{pmatrix} 1 & 0 \\ 0 & \exp(-\frac{2\pi i}{2^n} k) \end{pmatrix} \quad (11.125)$$

Consequently, in our 3-qubit example case, Q^3 reduces to:

$$Q^3 = \text{QFT}^{-1} \cdot T^3 \cdot \text{QFT} \quad (11.126)$$

Once, we recognize this basic structure further reductions become pretty easy to spot:

$$\begin{aligned} U &= Q^3 \cdot \text{TOFFOLI} \cdot Q^{\dagger 3} \\ &= \text{QFT}^{-1} \cdot T^3 \cdot \text{QFT} \cdot \text{TOFFOLI} \cdot (\text{QFT}^{-1} \cdot T^3 \cdot \text{QFT})^{-1} \\ &= \text{QFT}^{-1} \cdot T^3 \cdot \text{QFT} \cdot \text{TOFFOLI} \cdot \text{QFT}^{-1} \cdot T^{\dagger 3} \cdot \text{QFT} \end{aligned} \quad (11.127)$$

which can be further simplified by recognizing that $T^3 = Z \otimes R_z(\frac{\pi}{2}) \otimes R_z(-\frac{3\pi}{4})$.

Hence, we have succeeded in factorizing the compressor U in terms of TOFFOLI and Q gates, which in turn can both be reduced explicitly to 1-qubit and CNOT gates. Hence, our quantum circuit for the compressor, U , for the 3-qubit example block-size used, is shown in Fig. 11.8.

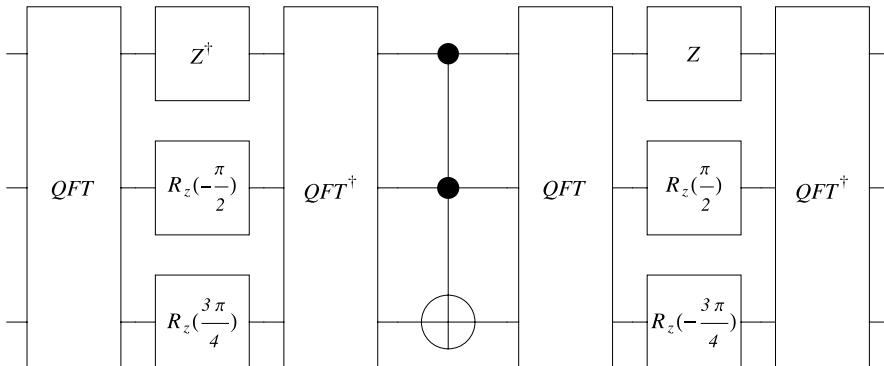


Fig. 11.8 Quantum circuit for the data compressor, U , used in the Schumacher-Jozsa quantum data compression protocols. The example given is appropriate for a block size of three qubits. The quantum circuit for the decompressor, U^\dagger , uses the inverse versions of the same gates applied in reverse order

11.4.6 Is Exponential Compression Possible?

A final thought on information compression in the quantum context is worthwhile. Let us compare the storage capacity of the Library of Congress to that a single qubit. Imagine, for example, that we translate each book in the Library of Congress into a bit string and concatenate them together. Then the entire Library, the entire repository of humankind's literary work product, is equivalent to some (very long) binary string. Let us call this string, s say. Ok ... perhaps such an important bit string deserves a more grandiose letter. You've convinced me—let's call it Σ instead.

Now let's imagine affixing a period to the front of Σ to make $. \Sigma$. Having done so ". Σ " can be regarded as a binary *fraction* $0.j_1 j_2 \dots j_n$. This represents a real number between 0 and 1 specifically $0 \leq \phi = j_1 2^{-1} + j_2 2^{-2} + \dots + j_n 2^{-n} \leq 1$. Thus, in principle, we could imagine creating a single qubit state of the form

$$|\psi\rangle_\Sigma = |0\rangle + \exp(i\phi)|1\rangle \quad (11.128)$$

and so this single qubit state contains (in some sense) the entire body of human knowledge! So, on the face of it, it may appear possible to compress information into a single qubit by an exponential factor. Unfortunately, this is not possible. To encode all the bits needed to specify the complete contents of the Library of Congress would require a physically unrealistic precision in setting the angle ϕ . Moreover, any single attempt to perform a measurement on $|\psi\rangle_\Sigma$, or any transformed version thereof, will only reveal at most one bit of information. It is neither practically possible to cram the Library or Congress into a single qubit, nor to extract more than one bit of information from a single qubit state.

11.5 Superdense Coding

We know from Sect. 11.1.2 that if Alice wants to send Bob a *classical* message over a *classical* communications channel, the maximum extent to which she can compress her message is set by Shannon’s Source Coding Theorem. This states that, if Alice wants negligible risk of information loss, a message comprising a string of n bits in which symbol 0 occurs with probability p_0 , and symbol 1 occurs with probability $p_1 = 1 - p_0$ cannot be compressed into less than $nH(\{p_0, p_1\})$ bits, where $0 \leq H(\{p_0, p_1\}) \leq 1$ is the Shannon entropy of the source. The question of interest is whether Alice can compress her classical message beyond this Shannon bound if she is able to send it over a *quantum* communications channel?

At first sight it seems impossible for Alice to do any better than what is allowed by the Source Coding Theorem. Even if we allow Alice to use quantum states to encode her classical bits, the fact that we require those quantum states to be unambiguously distinguishable, consistent with our commonsense view of what it means to be a classical “symbol”, forces Alice to have to use *orthogonal* quantum states to do the encoding. Thus, Alice could choose quantum state $|0\rangle$ to represent a classical bit 0, and quantum state $|1\rangle$ to represent classical bit 1. However, if Alice does this, the resulting von Neumann entropy, $S_V(\rho) = -\text{tr}(\rho \log_2 \rho)$, of her “quantum” source, described by density operator $\rho = p_0|0\rangle\langle 0| + p_1|1\rangle\langle 1|$, will be identical to the Shannon entropy of her equivalent classical source, having a probability distribution $\{p_0, p_1\} = \{p_0, 1 - p_0\}$ over the “symbols” 0 and 1. Hence, the maximum compression that is allowed quantum mechanically, i.e., $nS_V(\rho)$ qubits per n qubit message, will be identical to the maximum compression Alice can achieve classically, i.e., $nH(\{p_0, p_1\})$ bits per n -bit message. It would seem, therefore, that Alice can realize no benefit whatsoever from having access to a quantum channel over which to send her classical message.

It turns out, however, that there *is* a way of using a quantum communications channel to compress a stream of classical bits—at *communications time*—beyond that allowed by Shannon’s Source Coding Theorem. The trick is to allow for the possibility of creating, distributing and storing certain entangled qubits (or “ebits” as they are called) over the quantum channel, prior to any “message” communications taking place. Then, when a classical message of n -bits needs to be communicated, it can be encoded in only $n/2$ qubits, sent over the quantum channel, and the measured jointly with some of the previously shared ebits already at the destination end of the channel in a such a ways as to re-constitute as the full classical message.

In fact, one could take a maximally compressed classical message, e.g., as given by a turbo code or low density parity check code, and then *further* compress this maximally compressed classical message into quantum message, at communications time, by an additional factor of two! As the result is, at communications time, a quantum message needing only half as many qubits as the (perhaps already maximally compressed) classical message, this trick is called “superdense coding” and is only possible using quantum information resources.

It is important to note that this scheme does not violate Shannon’s Source Coding Theorem because it requires certain quantum states to be created, distributed,

and stored across the quantum communications channel prior to any actual classical message being sent. When one takes account of the communication resources needed to distribute these shared prior states, and add it to the communications resources required to transmit the quantum-encoding of the classical message itself, the net efficiency is again identical to the Shannon bound. However, in many practical circumstances, it is possible to create, distribute, and store the ebits at leisure, so that an urgent classical message can be transmitted at double density at some critical communications time. That is the main benefit of superdense coding.

To understand how superdense coding works, we must first discuss the Bell states and how it is possible to interconvert between them by acting on only one end of a Bell state.

11.5.1 Bell States

The starting point for superdense coding is to begin with 2-qubit maximally entangled states such as the Bell states.

$$\begin{aligned} |\beta_{00}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\beta_{01}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\beta_{10}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\beta_{11}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \end{aligned} \quad (11.129)$$

which can be summarized as:

$$|\beta_{xy}\rangle = \frac{1}{\sqrt{2}}(|0, y\rangle + (-1)^x|1, 1-y\rangle) \quad (11.130)$$

Each of these Bell states can be synthesized from a different starting computational basis state in a quantum circuit consisting of a single Walsh-Hadamard gate and a CNOT. Specifically, we have:

$$|\beta_{xy}\rangle = \text{CNOT} \cdot (H \otimes \mathbb{1})|xy\rangle \quad (11.131)$$

where x and y can each be 0 or 1.

For superdense coding Alice is going to create Bell states in this manner, store one member of each pair, and transmit the other member to Bob, which he also indexes and stores. Provided neither qubit is measured the entanglement between the qubits in each Bell state is preserved. This shared prior entanglement becomes the resource upon which we will draw to achieve superdense coding of a subsequent classical message.

11.5.2 Interconversion Between Bell States by Local Actions

The Bell states have the curious property that they can be converted into one another by performing single qubit operations on just one of the qubits in each Bell pair. Moreover, this capability persists even if the two qubits in a Bell state become separated spatially over an arbitrarily large distance, provided neither of them is measured during the separation process.

To see this, suppose Alice and Bob each hold one member of a Bell state. The single qubit operation Alice needs to perform on her qubit, in order to convert the joint state into some other Bell state are as follows:

$$\begin{aligned} |\beta_{00}\rangle &\xrightarrow{\mathbb{1} \otimes \mathbb{1}} |\beta_{00}\rangle \\ |\beta_{00}\rangle &\xrightarrow{X \otimes \mathbb{1}} |\beta_{01}\rangle \\ |\beta_{00}\rangle &\xrightarrow{Z \otimes \mathbb{1}} |\beta_{10}\rangle \\ |\beta_{00}\rangle &\xrightarrow{Z \cdot X \otimes \mathbb{1}} |\beta_{11}\rangle \end{aligned} \tag{11.132}$$

$$\begin{aligned} |\beta_{01}\rangle &\xrightarrow{X \otimes \mathbb{1}} |\beta_{00}\rangle \\ |\beta_{01}\rangle &\xrightarrow{\mathbb{1} \otimes \mathbb{1}} |\beta_{01}\rangle \\ |\beta_{01}\rangle &\xrightarrow{Z \cdot X \otimes \mathbb{1}} |\beta_{10}\rangle \\ |\beta_{01}\rangle &\xrightarrow{Z \otimes \mathbb{1}} |\beta_{11}\rangle \end{aligned} \tag{11.133}$$

$$\begin{aligned} |\beta_{10}\rangle &\xrightarrow{Z \otimes \mathbb{1}} |\beta_{00}\rangle \\ |\beta_{10}\rangle &\xrightarrow{X \cdot Z \otimes \mathbb{1}} |\beta_{01}\rangle \\ |\beta_{10}\rangle &\xrightarrow{\mathbb{1} \otimes \mathbb{1}} |\beta_{10}\rangle \\ |\beta_{10}\rangle &\xrightarrow{Z \cdot X \cdot Z \otimes \mathbb{1}} |\beta_{11}\rangle \end{aligned} \tag{11.134}$$

$$\begin{aligned} |\beta_{11}\rangle &\xrightarrow{X \cdot Z \otimes \mathbb{1}} |\beta_{00}\rangle \\ |\beta_{11}\rangle &\xrightarrow{Z \otimes \mathbb{1}} |\beta_{01}\rangle \\ |\beta_{11}\rangle &\xrightarrow{Z \cdot X \cdot Z \otimes \mathbb{1}} |\beta_{10}\rangle \\ |\beta_{11}\rangle &\xrightarrow{\mathbb{1} \otimes \mathbb{1}} |\beta_{11}\rangle \end{aligned} \tag{11.135}$$

11.5.3 Superdense Coding Protocol

We now have all the pieces needed to understand superdense coding. The protocol is surprisingly straight forward.

Superdense Coding Suppose Alice wishes to send Bob a classical message comprising a string of bits. If Alice and Bob have a quantum channel, and quantum memories available to them, they can halve the required number of communicative acts needed at the time the message is sent, but exploiting entanglement resources created, shared, and stored, at an earlier time. The superdense coding protocol works as follows:

1. Before any information-bearing message is communicated, Alice creates several pairs of entangled qubits (i.e., ebits), each in the state $|\beta_{00}\rangle$, indexes and stores one member of each pair and passes the other member of the same pair to Bob.
2. Upon receipt Bob indexes and stores each ebit he receives from Alice. The result is that Alice and Bob come to possess matching members of pairs of entangled qubits each in the state $|\beta_{00}\rangle$ stored at matching index locations in some quantum memory.
3. Subsequently, when Alice wants to send Bob a two bit classical message, presented as the quantum state $|x\rangle|y\rangle$, she performs one of four possible operations on the next indexed ebit in her possession. By acting on her end of an entangled pair of qubits, Alice is able to convert the joint state of the entangled pair into any of the four Bell states. In particular, if $|x\rangle|y\rangle = |0\rangle|0\rangle$ Alice applies $\mathbb{1}$ (the identity) to her ebit. If $|x\rangle|y\rangle = |0\rangle|1\rangle$ she applies X (the Pauli- X gate) to her ebit. If $|x\rangle|y\rangle = |1\rangle|0\rangle$ she applies Z (the Pauli- Z gate) to her ebit. Finally, if $|x\rangle|y\rangle = |1\rangle|1\rangle$ she applies $Z \cdot X$ to her ebit. These operations transform the entangled state (initially $|\beta_{00}\rangle$) shared between Alice and Bob as follows:

$$\begin{aligned}
 & |00\rangle|\beta_{00}\rangle \xrightarrow{1 \otimes 1 \otimes 1 \otimes 1} |00\rangle|\beta_{00}\rangle \\
 & |01\rangle|\beta_{00}\rangle \xrightarrow{1 \otimes 1 \otimes X \otimes 1} |01\rangle|\beta_{01}\rangle \\
 & |10\rangle|\beta_{00}\rangle \xrightarrow{1 \otimes 1 \otimes Z \otimes 1} |10\rangle|\beta_{10}\rangle \\
 & |11\rangle|\beta_{00}\rangle \xrightarrow{1 \otimes 1 \otimes Z \cdot X \otimes 1} |11\rangle|\beta_{11}\rangle
 \end{aligned} \tag{11.136}$$

4. Alice then sends her “treated” ebit to Bob over the quantum communications channel.
5. Upon receipt, Bob performs a joint Bell state analysis on the ebit he receives from Alice together with the correspondingly indexed ebit from his quantum memory.
6. The Bell state analysis allows Bob to determine unambiguously which Bell state he has ($|\beta_{00}\rangle$, $|\beta_{01}\rangle$, $|\beta_{10}\rangle$, or $|\beta_{11}\rangle$) and hence what bit value pair Alice intended to send. Thus, if Alice and Bob share prior entanglement, then to send a two-bit message subsequently, Alice need only send a single “treated” ebit to Bob.

It is important to appreciate that superdense coding does not result in a *net* reduction in the communications resources needed to transmit n classical bits. However, it does allow us to time-shift when channel capacity is available. In essence, superdense coding can use an under-utilized channel at one time to share and store successive members of EPR pairs so that, at a later time, a classical n bit message

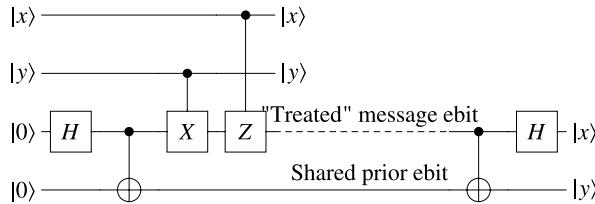


Fig. 11.9 Quantum circuit for superdense coding. Alice (using qubits 3 and 4 in the figure) prepares maximally entangled pairs of qubits (called “ebits”), keeps one member of each pair, and passes the other to Bob (qubit 4 in the figure). Subsequently, if Alice wants to send the bits xy encoded in the quantum state $|x\rangle|y\rangle$, she performs conditional operations on her retained ebit. This causes the entangled state shared between Alice and Bob to be set in the Bell state $|\beta_{xy}\rangle$. Next Alice transmits her “treated” ebit to Bob (qubit 3 in the figure). Upon receipt, Bob performs a complete Bell state analysis which allows him to determine which Bell state qubits 3 and 4 are in. This tells him what bit values Alice had intended to send. Thus, at communications time, Alice need only send one qubit to achieve the effect of sending two classical bits. Overall, superdense coding does not do any better than classical communications because of the communicative acts needed to establish the shared prior entanglement. Nevertheless, it does allow channel capacity available at one time to effectively be time-shifted to a later time

can be transmitted using the transmission of only $n/2$ qubits, consuming one EPR pair per qubit transmitted. Thus, the extra factor of two compression of the classical message can only be achieved for as long as the supply of EPR entangled particles lasts. However, this added factor of two additional compression is even possible if the classical message has already been maximally compressed (classically) using a turbo code or low density parity check code.

A quantum circuit for superdense coding is shown in Fig. 11.9

11.6 Cloning Quantum Information

One of the most useful aspects of classical information is our ability to copy, or “clone”, it reliably without any noticeable error. A photocopier, for example, can reproduce sheets of papers that are almost indistinguishable from the original. Digital computer files can be copied with even higher fidelity, in fact, perfectly. The ability to make perfect copies of classical data is also the curse of the entertainment and software industries because it also allows bootleggers to make illicit copies of digital music files, movies, and computer programs. As quantum information applications become more widespread it therefore behooves us to understand what can and cannot be done in terms of copying quantum information.

11.6.1 Historical Roots and Importance of Quantum Cloning

“I was the referee who approved the publication of Nick Herbert’s FLASH paper, knowing perfectly well that it was wrong. I explain why my decision was the correct one, [...]”

– Asher Peres [389]

The roots of quantum cloning can be traced back to a controversial paper written by Nick Herbert in 1981 describing an idea for a superluminal communicator based on the presumption that it is possible to make perfect copies (or clones) of an unknown quantum state. In 2002 Asher Peres revealed that he and Gian Carlo Ghirardi had been the “anonymous” reviewers of Herbert’s FLASH paper and that Ghirardi had recommended its rejection on the grounds that the linear nature of quantum mechanics meant that the supposed copying process could not exist. Peres likewise realized the paper was flawed but nevertheless recommended its publication in the hopes of stimulating others to find the flaw and thereby draw more attention to the emerging field of quantum information theory.

It turned out Peres was correct. Soon after Herbert’s paper was published William Wootters and Wojciech Zurek published a paper in Nature entitled “A Single Quantum Cannot be Cloned”, which basically re-discovered Ghirardi’s argument opposing Herbert’s paper [547]. Around the same time Dennis Dieks published a paper arguing that the claims of superluminal communications in Herbert’s paper were also flawed [142]. Thus the publication of the FLASH paper, and the reaction to it, went a long way towards stimulating more careful analyses of the properties of quantum information.

Recently, a more pragmatic motivation for studying quantum cloning has arisen from the need to understand how well an unscrupulous eavesdropper might be able to tap a quantum communications channel, whilst remaining undetected. If exact deterministic quantum cloning of unknown quantum states were possible (which luckily it isn’t), then an eavesdropper would be able to tap a quantum channel, forward perfect copies of the qubits to the intended recipient, and examine the copies they made at leisure. Fortunately, as we will show below, such exact deterministic quantum copying is physically impossible. Nevertheless, the practical question is how well can an eavesdropper do? How much information from a quantum channel can they extract without their presence being detected? With what fidelity *can* they copy unknown quantum states? And if they cannot copy states deterministically, can they do so probabilistically? These questions and others demonstrate the practical need to understand what physics permits one to do in terms of cloning quantum information.

11.6.2 Impossibility of Exact Deterministic Quantum Cloning

As in the classical case, an ideal universal quantum copy machine, or ideal universal quantum “cloning” machine as it is sometimes called, would be able to make a *perfect* copy of *any* quantum state it was handed. In particular, the action of an ideal universal quantum cloning machine, U_{clone} , on an arbitrary pure state $|\psi\rangle$ would be described as:

$$|\psi\rangle_A |0\rangle_B \xrightarrow{U_{\text{clone}}} |\psi\rangle_A |\psi\rangle_B \quad (11.137)$$

which we read as “particle A starts off in state $|\psi\rangle$, and particle B starts off in state $|0\rangle$, and after cloning the state of particle A, i.e., $|\psi\rangle$, is replicated on particle B.” This is more clearly seen to be a cloning procedure by suppressing the particle labels as in $|\psi\rangle|0\rangle \xrightarrow{U_{\text{clone}}} |\psi\rangle|\psi\rangle$.

Likewise, the ideal behavior of a quantum cloner when handed an arbitrary mixed state, ρ , would be:

$$\rho_A \otimes |0\rangle_B \langle 0|_B \xrightarrow{U_{\text{clone}}} \rho_A \otimes \rho_B \quad (11.138)$$

which we read as “particle A starts off in state ρ , and particle B starts off in state $|0\rangle\langle 0|$, and after cloning the state of particle A, i.e., ρ , is replicated on particle B.” As above, this is more clearly seen to be a cloning procedure by suppressing the particle labels as in $\rho \otimes |0\rangle\langle 0| \xrightarrow{U_{\text{clone}}} \rho \otimes \rho$.

The question is, does Nature permit such an ideal exact deterministic quantum cloning operation? To proceed, let us assume that U_{clone} is a perfect quantum cloning machine, i.e., a unitary operation such that whatever quantum state is given as input, two perfect copies of it are returned after U_{clone} has acted. In particular, U_{clone} will clone (say) the computational basis states perfectly. Thus, we would have:

$$\begin{aligned} |0\rangle|0\rangle &\xrightarrow{U_{\text{clone}}} |0\rangle|0\rangle \\ |1\rangle|0\rangle &\xrightarrow{U_{\text{clone}}} |1\rangle|1\rangle \end{aligned} \quad (11.139)$$

So far so good. But now let's assume the same machine was handed the states $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ instead, which are rotated with respect to the computational basis states. In this case, a *proper* quantum cloning machine is required to act as follows:

$$\begin{aligned} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle &\xrightarrow{U_{\text{clone}}} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\rangle \\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|0\rangle &\xrightarrow{U_{\text{clone}}} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\rangle \end{aligned} \quad (11.140)$$

But this not what our *supposed* quantum cloning machine U_{clone} does! If U_{clone} clones the computational basis states ($\{|0\rangle, |1\rangle\}$) correctly then, *by the linearity of quantum mechanics*, U_{clone} will transform the input states $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ as follows:

$$\begin{aligned} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle &\xrightarrow{U_{\text{clone}}} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|0\rangle &\xrightarrow{U_{\text{clone}}} \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \end{aligned} \quad (11.141)$$

In neither case is the output a product state of clones of the input state. Hence, if U_{clone} is a unitary procedure that clones computational basis states perfectly,

then it is guaranteed to clone states non-orthogonal to these imperfectly, and vice versa. This echoes the argument Ghirardi and Wootters and Zurek found against the FLASH paper. Hence, U_{clone} cannot be an ideal universal quantum cloning machine as we had supposed, and in fact the foregoing argument proves that ideal universal quantum cloning is *physically impossible* using any unitary operation whatsoever! Thus we arrive at the so-called “no-cloning” theorem for quantum information.

No Cloning Theorem *There is no deterministic quantum procedure by which an unknown pure quantum state can be cloned exactly.*

11.6.3 Universal Approximate Quantum Cloning

Although the quantum no-cloning theorem proves that it is impossible to clone an unknown quantum state perfectly deterministically it leaves open the possibility of cloning an unknown quantum state *approximately* deterministically, or perfectly *non-deterministically*. We will consider approximate deterministic cloning first.

If we are able to make an approximate clone, our main concerns are going to be how good an approximation can we obtain; whether the quality of the approximation can be made independent of the state we are trying to clone; and whether the resulting approximate clones can be used freely in subsequent quantum computations as proxies for the state that was cloned. The latter concern arises because if the approximate clones are entangled, then it may not matter how good they are individually, because using one of them could mess up the other one. This last point is often neglected but is, in fact, crucial to the whom concept of the utility of the clones.

These concerns were well appreciated by Vladimir Bužek and Mark Hillery. In 1996 they devised the first quantum cloning machine that produced high quality clones, whose fidelities were input independent, and which were practical to use in lieu of the original state in subsequent quantum computations [93]. Their elegant scheme for cloning a single qubit can be described as follows.

Imagine a 3-qubit quantum memory register with the qubits labeled A , B , and C . Qubit A is to hold the qubit whose state we wish to clone, and the outputs of qubits B and C are to hold the approximate clones. The quantum cloning machine will be unitary operation, \tilde{U}_{clone} , able to perform at least the following transformation on the computational basis states of qubit A , i.e., $|0\rangle_A$ and $|1\rangle_A$, augmented with a pair of ancillae prepared in the state $|00\rangle_{BC}$:

$$\begin{aligned} |0\rangle_A |0\rangle_B |0\rangle_C &\xrightarrow{\tilde{U}_{\text{clone}}} \sqrt{\frac{2}{3}}|000\rangle_{ABC} + \frac{1}{\sqrt{3}}|1\rangle_A \left[\frac{1}{\sqrt{2}}(|01\rangle_{BC} + |10\rangle_{BC}) \right] \\ |1\rangle_A |0\rangle_B |0\rangle_C &\xrightarrow{\tilde{U}_{\text{clone}}} \sqrt{\frac{2}{3}}|111\rangle_{ABC} + \frac{1}{\sqrt{3}}|0\rangle_A \left[\frac{1}{\sqrt{2}}(|01\rangle_{BC} + |10\rangle_{BC}) \right] \end{aligned} \quad (11.142)$$

Now imagine what the approximate quantum cloning transformation, \tilde{U}_{clone} , does to an arbitrary superposition state on qubit A , i.e., $|\psi\rangle_A = \alpha|0\rangle_A + \beta|1\rangle_A$.

A little algebra shows that \tilde{U}_{clone} will transform a general superposition of qubit A as:

$$\begin{aligned} |\Psi_{ABC}\rangle &= \tilde{U}_{\text{clone}}(\alpha|0\rangle_A + \beta|1\rangle_A)|0\rangle_B|0\rangle_C \\ &= \sqrt{\frac{2}{3}}\alpha|000\rangle + \frac{\beta}{\sqrt{6}}|001\rangle + \frac{\beta}{\sqrt{6}}|010\rangle + \frac{\alpha}{\sqrt{6}}|101\rangle \\ &\quad + \frac{\alpha}{\sqrt{6}}|110\rangle + \sqrt{\frac{2}{3}}\beta|111\rangle \end{aligned} \quad (11.143)$$

where we have dropped the qubit labels in the final output state. We can write $|\Psi_{ABC}\rangle$ equivalently as the density operator:

$$\rho_{ABC} = |\Psi_{ABC}\rangle\langle\Psi_{ABC}| = \begin{pmatrix} \frac{2}{3}|\alpha|^2 & \frac{1}{3}\alpha\beta^* & \frac{1}{3}\alpha\beta^* & 0 & 0 & \frac{1}{3}|\alpha|^2 & \frac{1}{3}|\alpha|^2 & \frac{2}{3}\alpha\beta^* \\ \frac{1}{3}\beta\alpha^* & \frac{1}{6}|\beta|^2 & \frac{1}{6}|\beta|^2 & 0 & 0 & \frac{1}{6}\beta\alpha^* & \frac{1}{6}\beta\alpha^* & \frac{1}{3}|\beta|^2 \\ \frac{1}{3}\beta\alpha^* & \frac{1}{6}|\beta|^2 & \frac{1}{6}|\beta|^2 & 0 & 0 & \frac{1}{6}\beta\alpha^* & \frac{1}{6}\beta\alpha^* & \frac{1}{3}|\beta|^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3}|\alpha|^2 & \frac{1}{6}\alpha\beta^* & \frac{1}{6}\alpha\beta^* & 0 & 0 & \frac{1}{6}|\alpha|^2 & \frac{1}{6}|\alpha|^2 & \frac{1}{3}\alpha\beta^* \\ \frac{1}{3}|\alpha|^2 & \frac{1}{6}\alpha\beta^* & \frac{1}{6}\alpha\beta^* & 0 & 0 & \frac{1}{6}|\alpha|^2 & \frac{1}{6}|\alpha|^2 & \frac{1}{3}\alpha\beta^* \\ \frac{2}{3}\beta\alpha^* & \frac{1}{3}|\beta|^2 & \frac{1}{3}|\beta|^2 & 0 & 0 & \frac{1}{3}\beta\alpha^* & \frac{1}{3}\beta\alpha^* & \frac{2}{3}|\beta|^2 \end{pmatrix} \quad (11.144)$$

This density operator ρ_{ABC} is therefore the output from our quantum cloning machine.

Next we determine the state of the clones individually by tracing out the unwanted qubits to obtain:

$$\rho_A = \text{tr}_{BC}(\rho_{ABC}) = \begin{pmatrix} \frac{2}{3}|\alpha|^2 + \frac{1}{3}|\beta|^2 & \frac{1}{3}\beta\alpha^* \\ \frac{1}{3}\alpha\beta^* & \frac{1}{3}|\alpha|^2 + \frac{2}{3}|\beta|^2 \end{pmatrix} \quad (11.145)$$

$$\begin{aligned} \rho_B &= \text{tr}_{AC}(\rho_{ABC}) = \begin{pmatrix} \frac{5}{6}|\alpha|^2 + \frac{1}{6}|\beta|^2 & \frac{2}{3}\alpha\beta^* \\ \frac{2}{3}\beta\alpha^* & \frac{1}{6}|\alpha|^2 + \frac{5}{6}|\beta|^2 \end{pmatrix} \\ &= \frac{5}{6}|\psi\rangle\langle\psi| + \frac{1}{6}|\psi^\perp\rangle\langle\psi^\perp| \end{aligned} \quad (11.146)$$

$$\begin{aligned} \rho_C &= \text{tr}_{AB}(\rho_{ABC}) = \begin{pmatrix} \frac{5}{6}|\alpha|^2 + \frac{1}{6}|\beta|^2 & \frac{2}{3}\alpha\beta^* \\ \frac{2}{3}\beta\alpha^* & \frac{1}{6}|\alpha|^2 + \frac{5}{6}|\beta|^2 \end{pmatrix} \\ &= \frac{5}{6}|\psi\rangle\langle\psi| + \frac{1}{6}|\psi^\perp\rangle\langle\psi^\perp| \end{aligned} \quad (11.147)$$

where $|\psi^\perp\rangle = \alpha^*|1\rangle - \beta^*|0\rangle$ is a state orthogonal to $|\psi\rangle$, which is the antipodal point to $|\psi\rangle$ on the Bloch sphere. Thus, we see that the reduced density operators for the clones contain the state being cloned plus some extra stuff we did not want.

To assess how close the clones are to the original state, we compute the fidelity of the clones, i.e., $\rho_B = \rho_C = \frac{5}{6}|\psi\rangle\langle\psi| + \frac{1}{6}|\psi^\perp\rangle\langle\psi^\perp|$, with respect to the original state, i.e., $\rho_{\text{ideal}} = |\psi\rangle\langle\psi|$. The formula for the fidelity with which one density operator, ρ , approximates another, σ , was given in Sect. 11.2.2.4 as:

$$\mathcal{F}(\rho, \sigma) = \left[\text{tr}\left(\sqrt{\sqrt{\rho} \sigma \sqrt{\rho}}\right) \right]^2 \quad (11.148)$$

Plugging the relevant density operators into this formula for fidelity we have:

$$\begin{aligned} \rho_{\text{ideal}} &= |\psi\rangle\langle\psi| = (\alpha|0\rangle + \beta|1\rangle)(\alpha^*\langle 0| + \beta^*\langle 1|) \\ &= \begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \beta\alpha^* & |\beta|^2 \end{pmatrix} \end{aligned} \quad (11.149)$$

and

$$\rho_B = \rho_C = \begin{pmatrix} \frac{5}{6}|\alpha|^2 + \frac{1}{6}|\beta|^2 & \frac{2}{3}\alpha\beta^* \\ \frac{2}{3}\beta\alpha^* & \frac{1}{6}|\alpha|^2 + \frac{5}{6}|\beta|^2 \end{pmatrix} \quad (11.150)$$

which gives $\mathcal{F}(\rho_{\text{ideal}}, \rho_B) = \mathcal{F}(\rho_{\text{ideal}}, \rho_C)$ as:

$$\begin{aligned} \mathcal{F}(\rho_{\text{ideal}}, \rho_B) &= \left[\text{tr}(\sqrt{\rho_{\text{ideal}}} \cdot \rho_B \cdot \sqrt{\rho_{\text{ideal}}}) \right]^2 \\ &= \left[\text{tr}(\rho_{\text{ideal}} \cdot \rho_B \cdot \rho_{\text{ideal}}) \right]^2 \\ &= \left[\text{tr}\left(\begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \beta\alpha^* & |\beta|^2 \end{pmatrix} \cdot \begin{pmatrix} \frac{5}{6}|\alpha|^2 + \frac{1}{6}|\beta|^2 & \frac{2}{3}\alpha\beta^* \\ \frac{2}{3}\beta\alpha^* & \frac{1}{6}|\alpha|^2 + \frac{5}{6}|\beta|^2 \end{pmatrix} \right. \right. \\ &\quad \left. \left. \cdot \begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \beta\alpha^* & |\beta|^2 \end{pmatrix} \right) \right]^2 \\ &= \frac{5}{6} \end{aligned} \quad (11.151)$$

where we used the fact that, as ρ_{ideal} is pure, $\sqrt{\rho_{\text{ideal}}} = \rho_{\text{ideal}}$. The same result holds for the second clone ρ_C . In both cases the fidelity with which quantum cloning is achieved is $\frac{5}{6}$. Moreover, as the fidelity we obtain is a numerical constant and does not involve α or β it must, therefore, be *independent* of the input state being cloned. So our cloning transform is a state independent cloner. However, whereas the original state was pure, the clones are mixed. We can understand how these two states are related in terms of the Bloch sphere/Bloch ball picture of a qubit (see Sect. 11.2.2.3). The pure state $|\psi\rangle$ is represented by a point on the surface of the Bloch sphere. If you imagine a vector drawn from center of the Bloch sphere to the point representing $|\psi\rangle$, then the clone is the mixed state obtained by shrinking the length of this vector radially without changing its direction. This may help you to visualize the physical meaning of an approximate quantum clone.

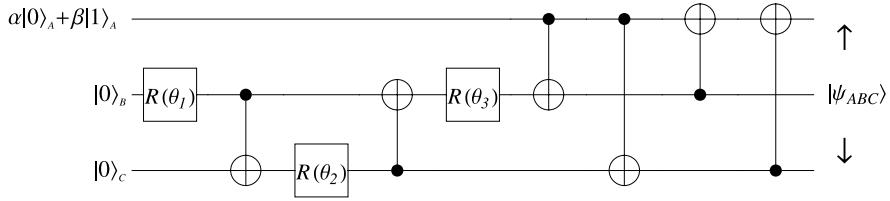


Fig. 11.10 Quantum circuit for cloning an unknown quantum state $|\psi\rangle_A = \alpha|0\rangle_A + \beta|1\rangle_A$. The clones appear in the output on qubits B and C . Their states are given by tracing out the other two qubits. That is, $\rho_B = \text{tr}_{AC}(|\psi_{ABC}\rangle\langle\psi_{ABC}|)$, and $\rho_C = \text{tr}_{AB}(|\psi_{ABC}\rangle\langle\psi_{ABC}|)$. Note that $\rho_B = \rho_C = \frac{5}{6}|\psi_A\rangle\langle\psi_A| + \frac{1}{6}|\psi_A^\perp\rangle\langle\psi_A^\perp|$, showing that the fidelity of the copies with respect to the original state is $\frac{5}{6}$

11.6.4 Circuit for Quantum Cloning

A quantum circuit that accomplishes our desired cloning transformation $|\psi\rangle_A|0\rangle_B|0\rangle_C \xrightarrow{\tilde{U}_{\text{clone}}} |\Psi_{ABC}\rangle$ is shown in Fig. 11.10. Here the 1-qubit gate $R(\theta)$ is defined to be:

$$R(\theta) := \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (11.152)$$

and the particular angles used are set at:

$$\begin{aligned} \theta_1 &= \frac{\pi}{8} \\ \theta_2 &= -\arcsin \sqrt{\frac{1}{6}(3 - 2\sqrt{2})} \\ \theta_3 &= \frac{\pi}{8} \end{aligned} \quad (11.153)$$

With these angle values, the quantum cloning circuit induces a (fixed) unitary transformation described by the matrix:

$$\left(\begin{array}{cccccccc} \sqrt{\frac{2}{3}} & 0 & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & 0 & \sqrt{\frac{2}{3}} \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & \sqrt{\frac{2}{3}} & 0 \\ 0 & \sqrt{\frac{2}{3}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{\frac{2}{3}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & \sqrt{\frac{2}{3}} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & 0 & \sqrt{\frac{2}{3}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{\frac{2}{3}} & 0 & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \end{array} \right) \quad (11.154)$$

As a check, it is easy to verify that this circuit transforms the basis states as follows:

$$\begin{aligned}\tilde{U}_{\text{clone}}|000\rangle &= \sqrt{\frac{2}{3}}|000\rangle + \frac{1}{\sqrt{6}}|101\rangle + \frac{1}{\sqrt{6}}|110\rangle \\ \tilde{U}_{\text{clone}}|100\rangle &= \frac{1}{\sqrt{6}}|001\rangle + \frac{1}{\sqrt{6}}|010\rangle + \sqrt{\frac{2}{3}}|111\rangle\end{aligned}\quad (11.155)$$

and hence transforms a superposition state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ as:

$$\begin{aligned}\tilde{U}_{\text{clone}}|\psi\rangle|0\rangle|0\rangle &= \sqrt{\frac{2}{3}}\alpha|000\rangle + \frac{\beta}{\sqrt{6}}|001\rangle + \frac{\beta}{\sqrt{6}}|010\rangle + \frac{\alpha}{\sqrt{6}}|101\rangle \\ &\quad + \frac{\alpha}{\sqrt{6}}|110\rangle + \sqrt{\frac{2}{3}}\beta|111\rangle\end{aligned}\quad (11.156)$$

which is exactly what is called for in (11.143).

11.6.5 Usability of the Quantum Clones

In an ideal universal cloning machine, the output clones would be *perfect* copies of the unknown state $|\psi\rangle$, and they would be unentangled from each other and the top qubit in the cloning circuit shown in Fig. 11.10. If these conditions hold, then the clones would clearly be useful as they could serve as perfect proxies for the state $|\psi\rangle$ in subsequent quantum computations. Unfortunately, the clones we obtain are neither perfect copies of the original state nor are they unentangled from each other and the top qubit of the cloning circuit. It is not immediately clear, therefore that cloning has achieved anything practically useful because, if the clones are entangled, operations performed on one of them might mess up the other. Furthermore, is a fidelity of $\frac{5}{6}$ really high enough to allow us to compute expectation values of observables that will be close enough to the true values to be useful? We will now address these issues by showing that although the clones are indeed entangled, they are nevertheless usable in subsequent quantum computations.

In an *ideal* cloning machine, an input state of the form $|\psi\rangle_A|0\rangle_B|0\rangle_C$ to be mapped into an output state of the form $|?\rangle_A|\psi\rangle_B|\psi\rangle_C$. Here perfect clones appear on qubits B and C , and they are unentangled from each other and from qubit A . Alas, we know from the no-cloning theorem, that quantum mechanics does not allow such perfection. Nevertheless, we can produce approximate clones on qubits B and C but these are no longer guaranteed to be unentangled from each other and unentangled from qubit A . If they are entangled then, potentially, subsequent operations on one clone could perturb the other clone (not to mention the ancilla). So we need to understand whether or not the clones are entangled.

11.6.5.1 Are the Clones Entangled?

First let us determine whether or not the clones are entangled with each other. That is, we test whether the joint density operator of the clones, ρ_{BC} , is separable or inseparable. To test this, we can use the Peres-Horodecki criterion of Sect. 11.3.3. As you will recall this test is based on checking whether there is at least one negative eigenvalue in the partial transpose of the density operator whose entanglement status is sought—in our case ρ_{BC} .

Starting with $\rho_{ABC} = |\Psi_{ABC}\rangle\langle\Psi_{ABC}|$ we obtain ρ_{BC} by tracing over qubit A (the top qubit in the circuit shown in Fig. 11.10), to obtain:

$$\rho_{BC} = \text{tr}_A(\rho_{ABC}) = \begin{pmatrix} \frac{2}{3}|\alpha|^2 & \frac{1}{3}\alpha\beta^* & \frac{1}{3}\alpha\beta^* & 0 \\ \frac{1}{3}\beta\alpha^* & \frac{1}{6} & \frac{1}{6} & \frac{1}{3}\alpha\beta^* \\ \frac{1}{3}\beta\alpha^* & \frac{1}{6} & \frac{1}{6} & \frac{1}{3}\alpha\beta^* \\ 0 & \frac{1}{3}\beta\alpha^* & \frac{1}{3}\beta\alpha^* & \frac{2}{3}|\beta|^2 \end{pmatrix} \quad (11.157)$$

Then, we compute the partial transpose of ρ_{BC} taken over the space “B” i.e., the space corresponding to the first of the two qubits in ρ_{BC} . This gives

$$\rho_{BC}^{T_B} = \begin{pmatrix} \frac{2}{3}|\alpha|^2 & \frac{1}{3}\alpha\beta^* & \frac{1}{3}\beta\alpha^* & \frac{1}{6} \\ \frac{1}{3}\beta\alpha^* & \frac{1}{6} & 0 & \frac{1}{3}\beta\alpha^* \\ \frac{1}{3}\alpha\beta^* & 0 & \frac{1}{6} & \frac{1}{3}\alpha\beta^* \\ \frac{1}{6} & \frac{1}{3}\alpha\beta^* & \frac{1}{3}\beta\alpha^* & \frac{2}{3}|\beta|^2 \end{pmatrix} \quad (11.158)$$

The eigenvalues of the partial transpose $\rho_{BC}^{T_B}$ can be obtained from the characteristic polynomial³ of the partial transpose $\rho_{BC}^{T_B}$, i.e., as the roots of:

$$\det(\rho_{BC}^{T_B} - \lambda\mathbb{1}) = \frac{(6\lambda - 1)^2(36\lambda^2 - 24\lambda - 1)}{1296} = 0 \quad (11.159)$$

Amazingly, after simplifying $\det(\rho_{BC}^{T_B} - \lambda\mathbb{1})$ by using the fact that $|\alpha|^2 + |\beta|^2 = 1$ and $|\alpha| \leq 1$, the resulting characteristic polynomial does not contain any mention of α and β ! This means that the eigenvalues of $\rho_{BC}^{T_B}$ are independent of the state being cloned, and are in fact equal to $\frac{1}{6}, \frac{1}{6}, \frac{1}{6}(2 - \sqrt{5})$, and $\frac{1}{6}(2 + \sqrt{5})$. As $\sqrt{5} > 2$, we see that the third eigenvalue is assuredly negative. Thus, by the Peres-Horodecki criterion ρ_{BC} , which is the joint state of the clones, must be an entangled.⁴ Rats!

³The characteristic polynomial of a square matrix U is the left hand side of the equation $\det(U - \lambda\mathbb{1}) = 0$ where $\mathbb{1}$ is the identity matrix. The roots of the characteristic polynomial are the eigenvalues of the matrix U .

⁴N.B. If we had computed, instead, the partial transpose over the space “C” i.e., the space corresponding to the second of the two qubits in ρ_{BC} , we would have obtained a different matrix for the partial transpose, $\rho_{BC}^{T_C}$, but its eigenvalues would have been the same as those of $\rho_{BC}^{T_B}$, and therefore one would have still been negative.

11.6.5.2 How Entangled are the Clones?

Just how entangled are the clones? To quantify the degree to which the clones are entangled we can compute the tangle of ρ_{BC} . Tangle, as a measure of entanglement for pure states, was introduced in Sect. 2.8.1. However, it generalizes readily to the case of mixed states. Define $\tilde{\rho}$ as the “spin-flipped” version of a density operator ρ :

$$\tilde{\rho} = (Y \otimes Y) \cdot \rho \cdot (Y \otimes Y) \quad (11.160)$$

then the tangle of ρ , $\text{tangle}(\rho)$, is related to the eigenvalues of the operator $\rho \cdot \tilde{\rho}$. Specifically, if the four eigenvalues of $\rho \cdot \tilde{\rho}$ are arranged in decreasing order so that $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4$, then:

$$\text{tangle}(\rho) = [\max(\sqrt{\lambda_1} - \sqrt{\lambda_2} - \sqrt{\lambda_3} - \sqrt{\lambda_4}, 0)]^2 \quad (11.161)$$

For the density operator, ρ_{BC} , the spin-flipped version is:

$$\begin{aligned} \widetilde{\rho_{BC}} &= (Y \otimes Y) \cdot \rho_{BC} \cdot (Y \otimes Y) \\ &= \begin{pmatrix} \frac{2}{3}|\beta|^2 & -\frac{1}{3}\alpha\beta^* & -\frac{1}{3}\alpha\beta^* & 0 \\ -\frac{1}{3}\beta\alpha^* & \frac{1}{6} & \frac{1}{6} & -\frac{1}{3}\alpha\beta^* \\ -\frac{1}{3}\beta\alpha^* & \frac{1}{6} & \frac{1}{6} & -\frac{1}{3}\alpha\beta^* \\ 0 & -\frac{1}{3}\beta\alpha^* & -\frac{1}{3}\beta\alpha^* & \frac{2}{3}|\alpha|^2 \end{pmatrix} \end{aligned} \quad (11.162)$$

and so the eigenvalues of $\rho_{BC} \cdot \widetilde{\rho_{BC}}$ are the roots of the corresponding characteristic polynomial:

$$\det(\rho_{BC} \cdot \widetilde{\rho_{BC}} - \lambda \mathbb{1}) = \lambda^3 \left(\lambda - \frac{1}{9} \right) = 0 \quad (11.163)$$

Amazingly again, after simplifying $\det(\rho_{BC} \cdot \widetilde{\rho_{BC}} - \lambda \mathbb{1})$ by using the fact that $|\alpha|^2 + |\beta|^2 = 1$ and $|\alpha| \leq 1$, the resulting characteristic polynomial does not contain any mention of α and β . This means that the eigenvalues of $\rho_{BC} \cdot \widetilde{\rho_{BC}}$ are independent of the state being cloned, and are in fact equal to 0, 0, 0, and $\frac{1}{9}$. Thus, arranging the eigenvalues in decreasing order so that $\lambda_1 = \frac{1}{9}$, $\lambda_2 = \lambda_3 = \lambda_4 = 0$, and taking square roots, the tangle is then given by:

$$\begin{aligned} \text{tangle}(\rho_{BC}) &= [\max(\sqrt{\lambda_1} - \sqrt{\lambda_2} - \sqrt{\lambda_3} - \sqrt{\lambda_4}, 0)]^2 \\ &= \left[\max\left(\sqrt{\frac{1}{9}} - \sqrt{0} - \sqrt{0} - \sqrt{0}, 0\right) \right]^2 = \frac{1}{9} \end{aligned} \quad (11.164)$$

This is actually not that bad. A maximally entangled 2-qubit state has a tangle of 1, so $\text{tangle}(\rho_{BC}) = \frac{1}{9}$ is fairly small. So the clones are far from being maximally entangled. However, the fact that the clones are entangled at all could spell trouble because when one uses one of the clones, the operations performed on it, could change the other clone. Hence, we might wonder whether we can use the two clones

freely in subsequent quantum computations. Furthermore, the fidelity of the clones, $\frac{5}{6}$ is noticeably less than 1. Is this good enough to learn anything trustworthy about ρ_{ideal} by subsequent observations on the clones ρ_B and ρ_C ? These issues are resolved in the next two sections.

11.6.5.3 Expectation Value of an Observable Based on Ideal State

To assess how useful the clones really are, we need to examine how the expectation value of a general operator Ω , when the system is in a clone state ρ_B or ρ_C , differs from the expectation value of the same operator when the system is in the original state $\rho_{\text{ideal}} = |\psi\rangle\langle\psi|$.

The ideal state is just the original state $|\psi\rangle$ we are trying to clone. Thus we have:

$$\rho_{\text{ideal}} = |\psi\rangle\langle\psi| = \begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \beta\alpha^* & |\beta|^2 \end{pmatrix} \quad (11.165)$$

Without loss of generality, the general form for an arbitrary 1-qubit observable operator, Ω , can be defined symbolically as:

$$\Omega = \begin{pmatrix} p & z \\ z^* & q \end{pmatrix} \quad (11.166)$$

where p and q are *real* numbers and z is (in general) a *complex* number. Any 1-qubit observable operator has to adhere to this form to be hermitian.

Now we can compute the expectation value of the observable Ω when the system is in state ρ_{ideal} . Using the formula given in Table 11.1 for computing the expectation value of an observable of a state defined by a density operator we have:

$$\langle \Omega \rangle = \text{tr}(\rho_{\text{ideal}}\Omega) = (p\alpha + z\beta)\alpha^* + (q\beta + \alpha z^*)\beta^* \quad (11.167)$$

This result is therefore our “gold standard” against which the quality of our clones can be judged.

11.6.5.4 Expectation Value of an Observable Based on a Clone

Now let us re-derive the expectation value $\langle \Omega \rangle$ this time using our clones. We want to tell two things. First, given that the clone is imperfect, what is the relationship between an operator expectation value for a clone state compared to that of the ideal state? Second, given that the clones are entangled, can we still use both clones in determining expectation values or does the use of one of them, render the other useless?

The state of each single clone is given by tracing over the other two qubits in the output state $|\Psi_{ABC}\rangle\langle\Psi_{ABC}|$. We calculated the reduced density matrices of the clones

in (11.146) and (11.147). We found that:

$$\begin{aligned}\rho_B &= \text{tr}_{AC}(\rho_{ABC}) = \begin{pmatrix} \frac{5}{6}|\alpha|^2 + \frac{1}{6}|\beta|^2 & \frac{2}{3}\alpha\beta^* \\ \frac{2}{3}\beta\alpha^* & \frac{1}{6}|\alpha|^2 + \frac{5}{6}|\beta|^2 \end{pmatrix} \\ \rho_C &= \text{tr}_{AB}(\rho_{ABC}) = \begin{pmatrix} \frac{5}{6}|\alpha|^2 + \frac{1}{6}|\beta|^2 & \frac{2}{3}\alpha\beta^* \\ \frac{2}{3}\beta\alpha^* & \frac{1}{6}|\alpha|^2 + \frac{5}{6}|\beta|^2 \end{pmatrix}\end{aligned}\quad (11.168)$$

These reduced density matrices for the clones are telling. These are the states we will appear to have regardless of what happens to the other clone. So provided we can milk some useful information out of ρ_B and ρ_C we do not need to worry further about the fact that the clones are actually entangled. So can we extract useful information?

Well surprisingly, although the clones ρ_B and ρ_C are only *approximations* to the ideal state ρ_{ideal} we can, in principle, use them to obtain the *exact* expectation values for any operator, \mathcal{Q} ! This is remarkable. The trick is to write “1” in the form “ $|\alpha|^2 + |\beta|^2$ ” to see that the following identity holds:

$$\rho_B = \rho_C = \frac{2}{3}\rho_{\text{ideal}} + \left(\frac{|\alpha|^2}{6} + \frac{|\beta|^2}{6} \right) \mathbb{1} \quad (11.169)$$

where $\mathbb{1}$ is the identity matrix. It then follows that:

$$\langle \mathcal{Q} \rangle = \text{tr}(\rho_{\text{ideal}} \cdot \mathcal{Q}) = \frac{3}{2} \left(\text{tr}(\rho_B \cdot \mathcal{Q}) - \frac{1}{6} \text{tr}(\mathcal{Q}) \right) \quad (11.170)$$

Thus we can use the clones to obtain the exact value of any observable, even though they are only approximations to the ideal clone, and even though they are entangled. I find this really a most amazing result!

11.6.6 Universal Probabilistic Quantum Cloning

Recall that the no-cloning theorem proves the impossibility of cloning an unknown state exactly deterministically. Yet it does not preclude the possibility of cloning an unknown state *approximately* deterministically, or cloning one exactly *non-deterministically*. In the preceding sections we showed that approximate deterministic quantum cloning machines are feasible. These are quantum circuits that use only unitary quantum gates to produce approximate clones that are described by reduced density matrices corresponding to mixed states. So even if the input state is pure the approximate clone is mixed.

In this section we show that the alternative strategy of exact albeit non-deterministic cloning machines are also feasible. We call such devices “probabilistic cloning machines” because they might not produce clones every time they run but

when they do the fidelity of those clones is higher than what can be achieved deterministically. The quantum circuits corresponding to probabilistic cloning machines use measurements, in addition to unitary gates, to achieve the desired state transformation. The success of the exact probabilistic cloning procedure is signalled by obtaining a specific outcome for these measurements.

The first design for a probabilistic cloning machine is due to Lu-Ming Duan and Guang-Can Guo [152]. They showed that if states are selected secretly from a set $\{|\psi_1\rangle, |\psi_2\rangle, \dots, |\psi_n\rangle\}$ they can be cloned exactly probabilistically if and only if the $\{|\psi_i\rangle\}$ are linearly independent. In other words, probabilistic cloning does not work for arbitrary states—they must be linearly independent—but their precise identity does not need to be known so long as the promise holds that they are linearly independent. If this condition holds, then Duan and Guo showed that there exists a unitary operation U and measurement M such that the following transformation is possible:

$$|\psi_i\rangle|\Sigma\rangle \xrightarrow{U \& M} |\psi_i\rangle|\psi_i\rangle \quad (11.171)$$

Here the measurement M means that the transformation is non-unitary overall, which is what allows it to appear to circumvent the no-cloning theorem.

To obtain such a transformation we need to design a unitary transformation and a measurement that does the trick. We begin by imaging there are three sub-spaces to our system A , B , and C . Sub-space A holds the state to be cloned. Sub-space B will hold the clone. And sub-space C will hold ancillae states that we intend to measure.

We can begin by defining an orthonormal set of $(n + 1)$ states of a so-called measurement probe $\{|P_0\rangle, |P_1\rangle, \dots, |P_n\rangle\}$. These states can serve as an unambiguous measurement basis provided $\langle P_i | P_j \rangle = 0$ for $i \neq j$ and $\langle P_i | P_i \rangle = 1$. Given such basis states, and a state selected secretly from our linearly independent set $\{|\psi_i\rangle\}$, probabilistic cloning works by creating a unitary evolution of the form:

$$|\psi_i\rangle_A|\Sigma\rangle_B|P_0\rangle_C \xrightarrow{U} \sqrt{p_i}|\psi_i\rangle_A|\psi_i\rangle_B|P_0\rangle_C + \sum_{j=1}^n c_{ij}|\Phi_j\rangle_{AB}|P_j\rangle_C \quad (11.172)$$

followed by a measurement of sub-system C in the $\{|P_1\rangle, |P_2\rangle, \dots, |P_n\rangle\}$ basis. In this transformation $|\Phi_1\rangle_{AB}, |\Phi_2\rangle_{AB}, \dots, |\Phi_n\rangle_{AB}$ are n normalized states of sub-systems A and B combined, but they are not necessarily orthogonal. Given the structure of the state produced under the action of U on an input $|\psi_i\rangle_A|\Sigma\rangle_B|P_0\rangle_C$ we can see immediately that exact cloning will be achieved whenever the measurement on sub-space C in the $\{|P_0\rangle, |P_1\rangle, \dots, |P_n\rangle\}$ -basis yields the result $|P_0\rangle$. Moreover, this event will occur with probability p_i , which we can think of as the “cloning efficiency”.

The simplest case is when we want to clone one of only two linearly independent states $\{|\psi_1\rangle, |\psi_2\rangle\}$. In this case Duan and Guo show that the cloning efficiencies p_1 and p_2 must satisfy the inequality:

$$\frac{1}{2}(p_1 + p_2) \leq \frac{1}{1 + \langle \psi_1 | \psi_2 \rangle} \quad (11.173)$$

This result can be generalized to bound all the probabilities p_1, p_2, \dots, p_n based on a certain matrix having to be positive semi-definite.

Optimal probabilistic cloning is closely related to the task of optimal unambiguous quantum state discrimination [109, 248, 254, 386].

11.6.7 Broadcasting Quantum Information

Extending the notion of quantum cloning to mixed states requires a little thought, because a complication arises that we do not have in the case of pure states. Given that we don't directly “see” the quantum state produced by cloning, but rather only experience it through the statistical properties it displays, we might wonder whether our goal is to clone a given mixed state literally, or merely produce clones that replicate the statistical properties of the given mixed state? This distinction can be best appreciated in terms of the two possible ways we could set up the notion of cloning for mixed states. These are usually distinguished by contrasting them as “cloning” versus “broadcasting”.

$$\rho \otimes |\Phi\rangle\langle\Phi| \xrightarrow{\text{Cloner}} \rho \otimes \rho \quad (11.174)$$

$$\rho \otimes |\Phi\rangle\langle\Phi| \xrightarrow{\text{Broadcaster}} \rho_{AB} : \text{tr}_A(\rho_{AB}) = \rho \text{ and } \text{tr}_B(\rho_{AB}) = \rho \quad (11.175)$$

The problem is that there are many density operators that can mimic the statistical behavior of the true clones. Hence, merely obtaining output density operators that display the same statistical properties as the true clones, is not entirely enough to allow us to conclude we really have true clones.

11.7 Negating Quantum Information

“The process of optimal quantum cloning is closely connected to another impossible operation in quantum mechanics, the so-called universal NOT gate for qubits.”

– Nicolas Cerf [99]

An ideal classical NOT gate, NOT, is able to negate any bit it is handed even if the bit value is unknown. That is, if $b \in \{0, 1\}$, $\text{NOT}b = 1 - b = \neg b$ regardless of value of b .

Similarly, an *ideal* universal⁵ quantum NOT gate (if it existed) would be able to negate any 1-qubit state it is handed. That is, for $|\psi\rangle = a|0\rangle + b|1\rangle$,

$$U_{\text{NOT}}^{\text{ideal}}|\psi\rangle = b^*|0\rangle - a^*|1\rangle \equiv |\psi^\perp\rangle \quad (11.176)$$

⁵Here “universal” means “input state independent.”

In terms of the Bloch sphere, $|\psi^\perp\rangle$ is the antipodal point to $|\psi\rangle$ on the opposite side of the Bloch sphere along a straight line through its center. Hence $|\psi\rangle$ and $|\psi^\perp\rangle$ are orthogonal quantum states, i.e., $\langle\psi|\psi^\perp\rangle = 0$.

Unfortunately, such an *ideal* universal quantum NOT operation requires that $U_{\text{NOT}}^{(\text{ideal})}$ be described by an anti-unitary matrix, whereas deterministic quantum gates are always described by unitary matrices. Hence it is impossible to achieve $U_{\text{NOT}}^{(\text{ideal})}$ exactly deterministically as purely a rotation on the Bloch sphere. Nevertheless, as in the case of quantum cloning, we can define a universal quantum NOT as the best approximation to the ideal NOT operation on qubits.

11.7.1 Universal Quantum Negation Circuit

Surprisingly, as the alert reader will have noticed, the desired negated state $|\psi^\perp\rangle$ happens to be produced as an “unwanted” side effect of using a universal quantum cloning circuit! In (11.146) and (11.147) we see the negated state appears as the “distortion” that prevents the clone for being exact. Specifically, we have:

$$\begin{aligned}\rho_B &= \frac{5}{6}|\psi\rangle\langle\psi| + \frac{1}{6}|\psi^\perp\rangle\langle\psi^\perp| \\ \rho_C &= \frac{5}{6}|\psi\rangle\langle\psi| + \frac{1}{6}|\psi^\perp\rangle\langle\psi^\perp|\end{aligned}\tag{11.177}$$

However, although we did not show this earlier, the contribution of the negated state, $|\psi^\perp\rangle$, to the top qubit A , turns out to be even greater. We can see this by factoring the reduced density operator ρ_A in terms of $|\psi\rangle$ and $|\psi^\perp\rangle$ as follows:

$$\begin{aligned}\rho_A &= \text{tr}_{BC}(\rho_{ABC}) = \begin{pmatrix} \frac{2}{3}|\alpha|^2 + \frac{1}{3}|\beta|^2 & \frac{1}{3}\beta\alpha^* \\ \frac{1}{3}\alpha\beta^* & \frac{1}{3}|\alpha|^2 + \frac{2}{3}|\beta|^2 \end{pmatrix} \\ &= Y \cdot \left(\frac{1}{3}|\psi\rangle\langle\psi| + \frac{1}{3}|\psi^\perp\rangle\langle\psi^\perp| \right) \cdot Y\end{aligned}\tag{11.178}$$

where

$$|\psi\rangle\langle\psi| = \begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \beta\alpha^* & |\beta|^2 \end{pmatrix}\tag{11.179}$$

$$|\psi^\perp\rangle\langle\psi^\perp| = \begin{pmatrix} |\beta|^2 & -\alpha\beta^* \\ -\beta\alpha^* & |\alpha|^2 \end{pmatrix}\tag{11.180}$$

In fact, it turns out that the *optimal* universal negating circuit is exactly the same as the optimal universal cloning circuit! The only difference, when we want to use the cloning circuit as a negating circuit, is that we pay attention to a different output qubit, namely the top qubit that contains ρ_A . Thus, a circuit for universal quantum negation is shown in Fig. 11.11.

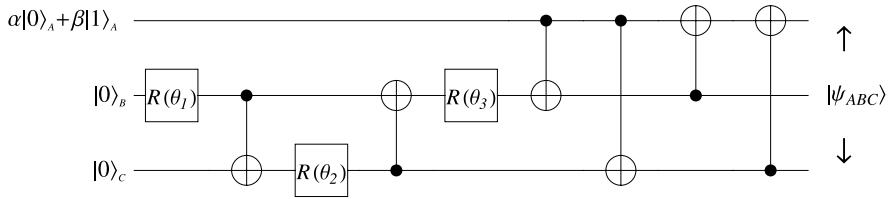


Fig. 11.11 Quantum circuit for universal quantum negation. In an ideal universal quantum negation circuit an unknown quantum state $|\psi\rangle_A = \alpha|0\rangle_A + \beta|1\rangle_A$ would be transformed into $\beta^*|0\rangle_A - \alpha^*|1\rangle_A$. This is not possible deterministically using unitary gates. Instead the best we can do is given by monitoring the output of the top qubit (A). The reduced density matrix of this qubit, $\rho_A = \text{tr}_{BC}(|\psi_{ABC}\rangle\langle\psi_{ABC}|)$ gives the best approximation to the negated state. This shows the fidelity of the negated state with respect to the ideal negated state is $\frac{1}{6}$

11.7.2 Expectation Value of an Observable Based on the Negated State

We can ask a similar question for universal negation that we asked for universal cloning: is the negated state, ρ_A , close enough to the ideal negated state, $|\psi^\perp\rangle\langle\psi^\perp|$, to be on use in subsequent quantum computations?

Using (11.178) we can express the ideal negated state, $\rho_{\text{ideal}}^{\text{UNOT}} = |\psi^\perp\rangle\langle\psi^\perp|$, in terms of the original state and the output on the top qubit of the quantum cloning (or equally, quantum “negating”) circuit:

$$\rho_{\text{ideal}}^{\text{UNOT}} = \frac{3}{2} \left(Y \cdot \rho_A \cdot Y - \frac{1}{3} |\psi\rangle\langle\psi| \right) \quad (11.181)$$

where $|\psi\rangle\langle\psi| = \rho_{\text{ideal}}^{\text{CLONE}}$. So for any observable operator \mathcal{Q} we would have:

$$\langle \mathcal{Q} \rangle = \text{tr}(\rho_{\text{ideal}}^{\text{UNOT}} \cdot \mathcal{Q}) = \frac{3}{2} \left(\text{tr}(Y \cdot \rho_A \cdot Y \cdot \mathcal{Q}) - \frac{1}{3} \text{tr}(|\psi\rangle\langle\psi| \cdot \mathcal{Q}) \right) \quad (11.182)$$

So we can obtain the exact expectation value of an operator on the true negated state, by using the approximation to the negated state on qubit A in conjunction with $\rho_{\text{ideal}}^{\text{CLONE}}$.

11.8 Summary

In Shannon’s view, information is equated to the representation of knowledge rather than the content of the knowledge per se. This view of “information” is alien to many people when they first encounter it. However, it turns out to be very useful in practice because it allows us to make concrete predictions on such matters as the degree to which an information bearing message can be compressed while ensuring the original message is recoverable, and the amount of redundancy to build into a communication to ensure it can be transmitted reliably through a noisy channel.

In the quantum context, the notion of information is extended in the obvious way by replacing classical streams of bits with quantum streams of qubits (possibly in non-orthogonal states). We found that the probability distribution by which we characterize a classical source is replaced by the density operator by which we characterize a corresponding quantum source. We introduced a new kind of entropy, the von Neumann entropy, which matches the Shannon entropy only when the quantum states are orthogonal and hence unambiguously distinguishable (like classical symbols). But when the quantum states are non-orthogonal, the von Neumann entropy exceeds the Shannon entropy. This allows certain operations on quantum information to exceed the bounds for corresponding operations on classical information. For example, we can compress quantum messages comprising non-orthogonal states over some probability distribution to a degree that is greater than that of classical messages over symbols that occur with the same probability distribution. We gave examples of two variants of such quantum compression protocols—discard-on-fail and augment-on-fail. More interestingly, we also found that we can use quantum information to compress a classical message by a factor of two beyond the Shannon bound at communication time provided we have already established and stored matching pairs of entangled qubits between the two ends of the communications channel. Thus, overall, Shannon's bound is not exceeded. However, at communication time, we can temporarily appear to exceed the Shannon by a factor of two for as long as the supply of matching entangled pairs remain.

Some operations that we take for granted on classical information are not so easy with quantum information. For example, whereas we can copy classical information perfectly deterministically, we cannot do so for quantum information in an unknown quantum state. Similarly, whereas we can negate classical information perfectly deterministically, we cannot negate quantum information in an unknown state. In both cases, however, we can find approximate quantum protocols that do as well as Nature allows. Surprisingly, we can use the approximate clones and approximate negated states to obtain *exact* expectation values of observable operators based on them. So in this sense, they are almost as useful as having perfect clones and perfect negated states.

The main difference between quantum information and classical information is the ability of the former to use non-orthogonal states to represent symbols, and for those non-orthogonal states to be entangled. Neither of these options exists for classical information, and this difference is the root of the dissimilarities between quantum and classical information. We introduced the formalism of density operators to describe quantum sources. We showed how the partial trace was used to describe a part of a composite quantum system. We highlighted the difference between pure and mixed states and focussed on the maximally entangled variants of both kinds of quantum states. We introduced a measure of the degree of entanglement in a quantum state via the tangle, and showed that deciding whether or not a quantum was entangled could be answered using so-called entanglement witnesses or the Peres-Horodecki criterion.

11.9 Exercises

11.1 Calculate the density matrices for the following ensembles.

1. An ensemble of quantum states that are all $\frac{1}{2}|0\rangle - \frac{\sqrt{3}}{2}|1\rangle$.
2. An ensemble of quantum states that are all $\frac{1}{3\sqrt{3}}|01\rangle + \frac{1}{3}\sqrt{\frac{26}{3}}|10\rangle$.
3. An ensemble of quantum states that are $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ with probability 0.3, $\frac{1}{2}|0\rangle - \frac{\sqrt{3}}{2}|1\rangle$ with probability 0.4, and $|0\rangle$ with probability 0.3.

11.2 Compute the density operator for an ensemble that is 30% $|\psi_1\rangle = \frac{1}{\sqrt{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle$ and 70% $|\psi_2\rangle = \frac{2}{3}|0\rangle + \frac{\sqrt{5}}{3}|1\rangle$, and write its elements as decimal numbers. Now compute the density operators for the following ensembles:

1. An ensemble that is 50% $|\psi_1\rangle = 0.680082|0\rangle + 0.733136|1\rangle$ and 50% $|\psi_2\rangle = 0.599759|0\rangle + 0.800181|1\rangle$.
2. An ensemble that is 25% $|\psi_1\rangle = 0.568532|0\rangle + 0.822661|1\rangle$ and 75% $|\psi_2\rangle = 0.66363|0\rangle + 0.748061|1\rangle$.

What do you notice? Can you devise any experimental test to distinguish between these ensembles? Justify your answer.

11.3 What test on a density operator, ρ , tells you whether the state is pure or mixed? According to this test, does the density operator given by

$$\rho = \begin{pmatrix} \frac{1}{9} & 0 & -\frac{2}{9} & \frac{2}{9} \\ 0 & 0 & 0 & 0 \\ -\frac{2}{9} & 0 & \frac{4}{9} & -\frac{4}{9} \\ \frac{2}{9} & 0 & -\frac{4}{9} & \frac{4}{9} \end{pmatrix}$$

correspond to that of a pure state or a mixed state?

11.4 Under what conditions is a 2-qubit state said to be *separable*? Your definition should cover both pure states and mixed states.

11.5 Under what conditions is a 2-qubit density operator said to be that of a *pure* state?

11.6 Which of the following simultaneous conditions of a quantum state are possible? There may be more than one correct answer.

1. A state can be simultaneously pure and mixed.
2. A state can be simultaneously separable and entangled.
3. A state can be simultaneously entangled and mixed.
4. A state can simultaneously mixed and separable.

5. A state can be simultaneously entangled and pure.

11.7 What is the linear entropy of the density operator, ρ , defined by:

$$\rho = \begin{pmatrix} \frac{1}{8} & 0 & 0 & -\frac{\sqrt{3}}{8} \\ 0 & \frac{3}{8} & \frac{1}{8} & 0 \\ 0 & \frac{1}{8} & \frac{1}{8} & 0 \\ -\frac{\sqrt{3}}{8} & 0 & 0 & \frac{3}{8} \end{pmatrix} \quad (11.183)$$

Is linear entropy a good measure of the mixedness or the entanglement within a state? Explain your answer.

11.8 Exhibit a 2-qubit (i.e., 4×4) density operator having a linear entropy less than $\frac{8}{9}$ which is entangled.

11.9 Exhibit a 2-qubit (i.e., 4×4) density operator having a linear entropy less than $\frac{8}{9}$ which is separable.

11.10 What test based on the linear entropy of a density operator, ρ , tells you whether the state is entangled or separable? According to this test, does the density operator given by

$$\rho = \begin{pmatrix} 0.375003 & 0.0403853 & 0.0634155 & 0.00682943 \\ 0.0403853 & 0.126466 & 0.00682943 & 0.0213862 \\ 0.0634155 & 0.00682943 & 0.372806 & 0.0401487 \\ 0.00682943 & 0.0213862 & 0.0401487 & 0.125725 \end{pmatrix} \quad (11.184)$$

correspond to an entangled state or a separable state?

11.11 What is the von Neumann entropy of a mixed state described by a density operator ρ ? Is the von Neumann entropy a good measure of the mixedness or entanglement within a state? Calculate the von Neumann entropies of the following density operators:

1. The maximally mixed state

$$\rho = \begin{pmatrix} \frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & \frac{1}{4} \end{pmatrix}$$

2. The typical mixed state

$$\rho = \begin{pmatrix} 0.314815 & -0.165635i & 0 & 0.166667 \\ 0.165635i & 0.372685 & -0.165359 & 0 \\ 0 & -0.165359 & 0.145833 & 0 \\ 0.166667 & 0 & 0 & 0.166667 \end{pmatrix}$$

3. The maximally entangled mixed state

$$\rho = \begin{pmatrix} \frac{1}{3} & 0 & 0 & \frac{1}{10} \\ 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{10} & 0 & 0 & \frac{1}{3} \end{pmatrix}$$

11.12 Prove that the expectation value of an observable, \mathcal{O} , for a quantum system in state ρ , given by (11.8) can be re-expressed in trace form as given by (11.9). That is, prove $\langle \mathcal{O} \rangle = \sum_{i=1}^N p_i \langle \psi_i | \mathcal{O} | \psi_i \rangle = \text{tr}(\rho \cdot \mathcal{O})$ where \mathcal{O} is an hermitian matrix, and ρ is a density operator.

11.13 It is possible to inter-convert between Bell states by applying single qubit operation to one member of a Bell state pair. What Bell state transformations do the following 1-qubit gates bring about?

$$\begin{array}{lll} |\beta_{00}\rangle & \xrightarrow{R_y(-\pi) \otimes \mathbb{1}} & ??? \\ |\beta_{01}\rangle & \xrightarrow{Ph(\pi/2) \cdot R_y(\pi) \cdot R_z(\pi) \otimes \mathbb{1}} & ??? \\ |\beta_{10}\rangle & \xrightarrow{Ph(\pi/2) \cdot R_y(-\pi) \cdot R_z(\pi) \otimes \mathbb{1}} & ??? \\ |\beta_{11}\rangle & \xrightarrow{Ph(-\pi/2) \cdot R_z(-\pi) \otimes \mathbb{1}} & ??? \end{array}$$

11.14 Recall that the quantum No-Cloning theorem asserts that “An unknown quantum state cannot be cloned”. Thus, it is supposed to be impossible to find a unitary transformation that can accomplish the transformation $|\psi\rangle|0\rangle \rightarrow |\psi\rangle|\psi\rangle$ for $|\psi\rangle$ unknown. However, you see an article that challenges the veracity of the No-Cloning theorem based on the following argument:

- (a) A bit, by definition, can be only 0 or 1.
- (b) If you are given a bit but not told its value, then the bit is, by definition, unknown to you. So let’s call the bit value b , but leave the value unspecified.
- (c) Conceptually, you could use the bit value b to control the settings of a device such as a Pockels cell (see Chap. 13), that outputs a horizontally polarized photon if $b = 0$ and a vertically polarized photon if $b = 1$. Thus, without loss of generality, we can convert our unknown bit to an unknown quantum state, which we can represent as $|b\rangle$, without ever revealing the value of b .
- (d) Now imagine augmenting the output from the Pockels cell, the *unknown* state $|b\rangle$, with another photon in a *known* state $|0\rangle$ (horizontally polarized photon) and push them through some optical apparatus that implements a CNOT gate, i.e., compute $\text{CNOT}|b\rangle|0\rangle$. Clearly, b has to be either 0 or 1 so the only two cases we need to consider are $\text{CNOT}|0\rangle|0\rangle = |0\rangle|0\rangle$ and $\text{CNOT}|1\rangle|0\rangle = |1\rangle|1\rangle$.
- (e) Either way, the unknown quantum state $|b\rangle$ has been successfully cloned!
- (f) Therefore, the No-Cloning theorem must be wrong, because here we have successfully cloned an unknown quantum state $|b\rangle$!

What is wrong with this argument? Why does it not disprove the No-Cloning theorem? Justify your answer by critiquing each step in the aforementioned argument.

11.15 Given the density matrix:

$$\rho = \begin{pmatrix} \frac{4}{49} & -\frac{6}{35} & \frac{4\sqrt{314}}{735} & -\frac{4i}{21} \\ -\frac{6}{35} & \frac{9}{25} & -\frac{2\sqrt{314}}{175} & \frac{2i}{5} \\ \frac{4\sqrt{314}}{735} & -\frac{2\sqrt{314}}{175} & \frac{1256}{11025} & -\frac{4i\sqrt{314}}{315} \\ -\frac{4i}{21} & \frac{2i}{5} & \frac{4i\sqrt{314}}{315} & \frac{4}{9} \end{pmatrix} \quad (11.185)$$

prove that its two partial transposes, ρ^{T_A} and ρ^{T_B} , have the same set of eigenvalues.

11.16 What are the density matrices corresponding to the four pure Bell states, $|\beta_{00}\rangle$, $|\beta_{01}\rangle$, $|\beta_{10}\rangle$, or $|\beta_{11}\rangle$ as defined in (11.69)? Are they the same or different? Now compute the reduced density matrices obtained by tracing over each of the qubits in each of these Bell states. Are these reduced density matrices the same or different? If your results are different, use them to find a single qubit observable, $\Omega = \begin{pmatrix} a & c \\ c^* & b \end{pmatrix}$, which is able to distinguish between the four Bell states. Alternatively, if your results are the same, use them to prove no such observable exists.

11.17 One way to measure the similarity between a pair of density matrices, σ and ρ , is via their fidelity:

$$\mathcal{F}(\sigma, \rho) = \left[\text{tr}(\sqrt{\sqrt{\sigma} \cdot \rho \cdot \sqrt{\sigma}}) \right]^2 \quad (11.186)$$

Show that if σ is the density matrix of an arbitrary single qubit pure state, i.e., if $\sigma = |\psi\rangle\langle\psi|$ where $\psi = a|0\rangle + \sqrt{1-a^2}|1\rangle$ (with $|a| \leq 1$ and $a \in \mathbb{C}$), and $\rho = p|0\rangle\langle 0| + (1-p)|1\rangle\langle 1|$ (with $0 \leq p \leq 1$ and $p \in \mathbb{R}$) then the fidelity $\mathcal{F}(\sigma, \rho)$ can be written as:

$$\mathcal{F}(\sigma, \rho) = \langle\psi|\rho|\psi\rangle = 1 - p - (1 - 2p)|a|^2 \quad (11.187)$$

Notice that if $p = \frac{1}{2}$ the fidelity is then independent of a . What is so special about the state $\rho = p|0\rangle\langle 0| + (1-p)|1\rangle\langle 1|$ when $p = \frac{1}{2}$? Why should the fidelity between ρ when $p = \frac{1}{2}$ and any pure state be independent of the form of that pure state?

11.18 Consider the pair of entangled states $|\psi_W\rangle$ and $|\psi_{GHZ}\rangle$ defined on the three qubits A , B , and C as follows:

$$\begin{aligned} |\psi_W\rangle &= \frac{1}{\sqrt{3}}(|001\rangle + |010\rangle + |100\rangle); & \rho_{ABC}^W &= |\psi_W\rangle\langle\psi_W| \\ |\psi_{GHZ}\rangle &= \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle); & \rho_{ABC}^{GHZ} &= |\psi_{GHZ}\rangle\langle\psi_{GHZ}| \end{aligned} \quad (11.188)$$

Prove the following:

- (a) The states $|\psi_W\rangle$ and $|\psi_{GHZ}\rangle$ are orthogonal, i.e., $\langle\psi_W|\psi_{GHZ}\rangle = 0$. What does this tell you about the degree to which $|\psi_W\rangle$ is similar to $|\psi_{GHZ}\rangle$?
- (b) $|\psi_W\rangle$ and $|\psi_{GHZ}\rangle$ are both entangled states.
- (c) The 2-qubit sub-systems of $|\psi_W\rangle$ are identical, i.e. ignoring indices, we obtain the same state whether we trace over the first, second, or third qubit:

$$\rho_{BC}^W = \text{tr}_A(|\psi_W\rangle\langle\psi_W|) = \rho_{AC}^W = \text{tr}_B(|\psi_W\rangle\langle\psi_W|) = \rho_{AB}^W = \text{tr}_C(|\psi_W\rangle\langle\psi_W|) \quad (11.189)$$

- (d) The 2-qubit sub-systems of $|\psi_{GHZ}\rangle$ are identical, i.e. ignoring indices, we obtain the same state whether we trace over the first, second, or third qubit:

$$\begin{aligned} \rho_{BC}^{GHZ} &= \text{tr}_A(|\psi_{GHZ}\rangle\langle\psi_{GHZ}|) = \rho_{AC}^{GHZ} = \text{tr}_B(|\psi_{GHZ}\rangle\langle\psi_{GHZ}|) \\ &= \rho_{AB}^{GHZ} = \text{tr}_C(|\psi_{GHZ}\rangle\langle\psi_{GHZ}|) \end{aligned} \quad (11.190)$$

- (e) For any pair of indices $\{x, y\} \subset \{A, B, C\}$, the fidelity between the reduced density matrices ρ_{xy}^W and ρ_{xy}^{GHZ} is $\frac{1}{6}$. That is, prove $\mathcal{F}(\rho_{xy}^W, \rho_{xy}^{GHZ}) = \frac{1}{6}$.
- (f) The 1-qubit sub-systems of $|\psi_W\rangle$ are identical, i.e. ignoring indices, we obtain the same state whether we trace over the second and third, first and third, or first and second qubits:

$$\rho_A^W = \text{tr}_{BC}(|\psi_W\rangle\langle\psi_W|) = \rho_B^W = \text{tr}_{AC}(|\psi_W\rangle\langle\psi_W|) = \rho_C^W = \text{tr}_{AB}(|\psi_W\rangle\langle\psi_W|) \quad (11.191)$$

- (g) The 1-qubit sub-systems of $|\psi_{GHZ}\rangle$ are identical, i.e. ignoring indices, we obtain the same state whether we trace over the second and third, first and third, or first and second qubits:

$$\begin{aligned} \rho_A^{GHZ} &= \text{tr}_{BC}(|\psi_{GHZ}\rangle\langle\psi_{GHZ}|) = \rho_B^{GHZ} = \text{tr}_{AC}(|\psi_{GHZ}\rangle\langle\psi_{GHZ}|) = \rho_C^{GHZ} \\ &= \text{tr}_{AB}(|\psi_{GHZ}\rangle\langle\psi_{GHZ}|) \end{aligned} \quad (11.192)$$

- (h) For any index $x \in \{A, B, C\}$, the fidelity between the reduced density matrices ρ_x^W and ρ_x^{GHZ} is $\frac{1}{6}(3 + 2\sqrt{2})$. That is prove, $\mathcal{F}(\rho_x^W, \rho_x^{GHZ}) = \frac{1}{6}(3 + 2\sqrt{2})$.
- (i) What is the fidelity between the original pair of states $|\psi_W\rangle$ and $|\psi_{GHZ}\rangle$ in comparison to the fidelities of its 2-qubit and 1-qubit sub-systems?

11.19 Consider the state $|\psi_W\rangle$ defined in (11.188). Use the Schmidt decomposition to “automatically” discover the (trivial) factorization of $|\psi_W\rangle$ in the form:

$$|\psi_W\rangle = \sqrt{\frac{2}{3}}|0\rangle \otimes \left(\frac{|01\rangle + |10\rangle}{\sqrt{2}} \right) + \frac{1}{\sqrt{3}}|1\rangle|00\rangle \quad (11.193)$$

11.20 Consider the 3-qubit state $|\psi_{ABC}\rangle = \frac{1}{\sqrt{2}}(|\psi_W\rangle + |\psi_{GHZ}\rangle)$ where $|\psi_W\rangle$ and $|\psi_{GHZ}\rangle$ are defined as in (11.188). Suppose you wish to write $|\psi_{ABC}\rangle$ in the form:

$$|\psi_{ABC}\rangle = \sum_{i=0}^{\min(d_A-1, d_{BC}-1)} \lambda_i |i_A\rangle |i_{BC}\rangle \quad (11.194)$$

(single index summation) where A is a 2-dimensional subspace, and BC is a 4-dimensional subspace. Demonstrate how to apply the Schmidt decomposition to find suitable values for the Schmidt coefficients (λ_i) and the eigenvectors ($\{|i_A\rangle\}$ and $\{|i_{BC}\rangle\}$). Verify that your solution yields a Schmidt decomposition for $|\psi_{ABC}\rangle$ of the form:

$$|\psi_{ABC}\rangle = \lambda_1 |1_A\rangle |1_{BC}\rangle + \lambda_2 |2_A\rangle |2_{BC}\rangle \quad (11.195)$$

where:

$$\begin{aligned} \lambda_1 &= \sqrt{\frac{1}{2} + \frac{\sqrt{7}}{12}} \\ \lambda_2 &= \frac{1}{2} \sqrt{\frac{1}{3}(6 - \sqrt{7})} \\ |1_A\rangle &= -\sqrt{\frac{1}{2} + \frac{1}{2\sqrt{7}}} |0\rangle - \sqrt{\frac{1}{14}(7 - \sqrt{7})} |1\rangle \\ |2_A\rangle &= -\sqrt{\frac{1}{14}(7 - \sqrt{7})} |0\rangle + \sqrt{\frac{1}{2} + \frac{1}{2\sqrt{7}}} |1\rangle \\ |1_{BC}\rangle &= -\sqrt{\frac{17}{58} + \frac{43}{58\sqrt{7}}} |00\rangle - \sqrt{\frac{1}{203}(35 - \sqrt{7})} |01\rangle \\ &\quad - \sqrt{\frac{1}{203}(35 - \sqrt{7})} |10\rangle - \sqrt{\frac{3}{406}(49 - 13\sqrt{7})} |11\rangle \\ |2_{BC}\rangle &= \sqrt{\frac{1}{406}(119 - 43\sqrt{7})} |00\rangle - \sqrt{\frac{5}{29} + \frac{1}{29\sqrt{7}}} |01\rangle \\ &\quad - \sqrt{\frac{5}{29} + \frac{1}{29\sqrt{7}}} |10\rangle + \sqrt{\frac{3}{406}(49 + 13\sqrt{7})} |11\rangle \end{aligned} \quad (11.196)$$

11.21 Alice and Bob wish to perform a quantum mechanical experiment over a distance of 400 km. The experiment requires that Alice and Bob have corresponding members of maximally entangled pairs of particles. However, if they transmit a particle over 100 km they can no longer guarantee its state is pristine. How, in principle, can Alice and Bob establish the required entangled pairs of particles over a distance of 400 km? Explain, by describing the sequence of state changes, how

they could use this scheme to establish shared pairs of particles each in the state $\beta_{01} = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$.

11.22 Suppose Alice and Bob have access to ideal quantum memories, i.e., they are able to store quantum information without any loss of fidelity indefinitely. In addition, assume Alice and Bob are connected by a fiber optic communications network, which can support both quantum and classical communications but is shared with other users. This network is idle for approximately 20% of the time, under-utilized for 70% of the time and at peak congestion for 10% of the time. Explain how Alice and Bob can exploit quantum information to boost their effective communications capacity at times of peak congestion. At such times, by what factor can they, in principle, increase their effective communications rate? Can this enhanced communications rate be maintained indefinitely? Explain your answer.

11.23 The states that have the maximal possible amount of entanglement for a given amount of mixedness (as measured by linear entropy) can be written in the form:

$$\rho = \begin{cases} \begin{pmatrix} \frac{1}{3} & 0 & 0 & \frac{r}{2} \\ 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{r}{2} & 0 & 0 & \frac{1}{3} \end{pmatrix} & 0 \leq r \leq \frac{2}{3} \\ \begin{pmatrix} \frac{r}{2} & 0 & 0 & \frac{r}{2} \\ 0 & 1-r & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{r}{2} & 0 & 0 & \frac{r}{2} \end{pmatrix} & \frac{2}{3} < r \leq 1 \end{cases} \quad (11.197)$$

Show when $0 \leq r \leq \frac{2}{3}$ that ρ can be factored in the form:

$$\rho = p_1|\psi_1\rangle\langle\psi_1| + p_2|\psi_2\rangle\langle\psi_2| + (1 - (p_1 + p_2))|\psi_3\rangle\langle\psi_3| \quad (11.198)$$

where

$$\begin{aligned} p_1 &= \frac{1}{12}(4 - 9r^2) \\ p_2 &= \frac{1}{3} \\ |\psi_1\rangle &= |00\rangle \\ |\psi_2\rangle &= |01\rangle \end{aligned} \quad (11.199)$$

$$|\psi_3\rangle = \frac{3r}{\sqrt{4+9r^2}}|00\rangle + \frac{2}{\sqrt{4+9r^2}}|11\rangle$$

Likewise, show when $\frac{2}{3} < r \leq 1$ that ρ can be factored in the form:

$$\rho = (1 - r)|\psi_1\rangle\langle\psi_1| + r|\psi_2\rangle\langle\psi_2| \quad (11.200)$$

where

$$\begin{aligned} |\psi_1\rangle &= |01\rangle \\ |\psi_2\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \end{aligned} \quad (11.201)$$

11.24 We can always regard a mixed state as the reduced density matrix of a larger pure state within some sub-system of interest. The procedure for finding such an encompassing pure state is called “purification of a mixed state”, and was described in this Chapter. Review the purification procedure and apply it to show that the state

$$\begin{aligned} |\psi_{AB}\rangle &= \left(\frac{1}{4}\sqrt{\frac{7}{6}} - \frac{1}{4}\sqrt{\frac{3}{2}}\right)|0000\rangle + \left(\frac{1}{4}\sqrt{\frac{7}{6}} + \frac{1}{4}\sqrt{\frac{3}{2}}\right)|0011\rangle + \frac{1}{\sqrt{3}}|0101\rangle \\ &\quad - \frac{1}{\sqrt{6}}|1100\rangle + \frac{1}{\sqrt{6}}|1111\rangle \end{aligned} \quad (11.202)$$

is a purification of the mixed state

$$\rho_A = \begin{pmatrix} \frac{1}{3} & 0 & 0 & \frac{1}{4} \\ 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{4} & 0 & 0 & \frac{1}{3} \end{pmatrix} \quad (11.203)$$

Note that a state such as ρ_A has the maximum possible value of entanglement for the degree of mixedness (as measured by linear entropy) in ρ_A . Verify that $|\psi_{AB}\rangle$ is a purification of ρ_A by showing $\text{tr}_B(|\psi_{AB}\rangle\langle\psi_{AB}|) = \rho_A$, where sub-space A corresponds to the first and second qubits, and sub-space B corresponds to the third and fourth qubits.

11.25 Show how to construct the purification:

$$|\psi_{AB}\rangle = \frac{1}{2}\sqrt{\frac{3}{2}}|0001\rangle + \frac{1}{2\sqrt{2}}|0100\rangle + \frac{1}{2\sqrt{2}}|0111\rangle + \frac{1}{2}\sqrt{\frac{3}{2}}|1101\rangle \quad (11.204)$$

of the mixed state:

$$\rho_A = \begin{pmatrix} \frac{3}{8} & 0 & 0 & \frac{3}{8} \\ 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{3}{8} & 0 & 0 & \frac{3}{8} \end{pmatrix} \quad (11.205)$$

Note that a state such as ρ_A has the maximum possible value of entanglement for the degree of mixedness (as measured by linear entropy) in ρ_A . Verify that $|\psi_{AB}\rangle$ is a purification of ρ_A by showing $\text{tr}_B(|\psi_{AB}\rangle\langle\psi_{AB}|) = \rho_A$.

11.26 Use the Peres-Horodecki criterion to decide whether each of the following states is or is not entangled:

1.

$$|\psi\rangle = \frac{1}{\sqrt{6}} |00\rangle + \frac{1}{\sqrt{3}} |01\rangle + \frac{1}{4} |10\rangle + \frac{1}{4}\sqrt{7} |11\rangle \quad (11.206)$$

2.

$$|\psi\rangle = \frac{1}{5}\sqrt{3} |00\rangle + \frac{1}{5}\sqrt{6} |01\rangle + \frac{4}{5\sqrt{3}} |10\rangle + \frac{4}{5}\sqrt{\frac{2}{3}} |11\rangle \quad (11.207)$$

3.

$$\rho = \begin{pmatrix} \frac{5}{14} & 0 & 0 & \frac{5}{14} \\ 0 & \frac{2}{7} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{5}{14} & 0 & 0 & \frac{5}{14} \end{pmatrix} \quad (11.208)$$

4.

$$\begin{pmatrix} \frac{1}{90}(15 - 4\sqrt{3}) & -\frac{i}{15\sqrt{3}} & -\frac{1}{90}i(-15 + 4\sqrt{3}) & \frac{1}{15\sqrt{3}} \\ \frac{i}{15\sqrt{3}} & \frac{1}{15\sqrt{3}} & -\frac{1}{15\sqrt{3}} & \frac{2i}{15\sqrt{3}} \\ \frac{1}{90}i(-15 + 4\sqrt{3}) & -\frac{1}{15\sqrt{3}} & \frac{5}{6} - \frac{2}{3\sqrt{3}} & -\frac{i}{3\sqrt{3}} \\ \frac{1}{15\sqrt{3}} & -\frac{2i}{15\sqrt{3}} & \frac{i}{3\sqrt{3}} & \frac{2}{3\sqrt{3}} \end{pmatrix} \quad (11.209)$$

5.

$$\rho = \begin{pmatrix} \frac{2}{3}(1 - \frac{2}{\sqrt{5}}) & 0 & 0 & \frac{1}{15}\sqrt{2}(-5 + 2\sqrt{5}) \\ 0 & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & 0 \\ 0 & -\frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0 \\ \frac{1}{15}\sqrt{2}(-5 + 2\sqrt{5}) & 0 & 0 & \frac{1}{15}(5 - 2\sqrt{5}) \end{pmatrix} \quad (11.210)$$

11.27 There are many possible entanglement monotones that can be used to quantify the degree of entanglement within a quantum state. Two popular ones for 2-qubit states are “negativity” and “concurrence” (which in turn is just the square root of the tangle). Look up the definitions of negativity and concurrence (i.e., tangle) and then answer the following questions:

1. Compute the negativity and concurrence for each of the quantum states listed in Problem 11.26.
2. What do you notice about the values of negativity and concurrence when the states are pure?
3. What do you notice about the values of negativity and concurrence when the states are determined, e.g., by the Peres-Horodecki criterion, to be separable?

Chapter 12

Quantum Teleportation

“We report free-space implementation of quantum teleportation over 16 km”
– Jin et al.¹

In science fiction stories, teleportation is usually depicted as a routine means of relocating an object by a process of dissociation, information transmission, and reconstitution. When all goes well the original object is scanned and disassembled at one place only to shimmer reassuringly back into existence at another. For dramatic effect, occasional blunders corrupt the object en route or leave it suspended in some nebulous state. Hapless bit-part actors seem especially prone to malfunctions.

Such accounts of teleportation are convenient literary devices for moving action heroes around the Universe and for introducing paradoxes of identity into story lines. But to what extent is teleportation consistent with known physical laws? In particular, does quantum information offer any new possibilities? In this chapter we look at the *scientific* basis for teleportation.

12.1 Uncertainty Principle and “Impossibility” of Teleportation

Until recently no serious attention had been paid to the physical principles on which true teleportation might be based. The presumption of most scientists, if they had any, was that teleportation was impossible because it would require some sort of scanning, or measurement, operation in order to extract a precise description of the state of all the particles in a system. At the very least this would seem to necessitate having to learn, simultaneously, the positions and momenta of all the particles from which the object was made.

¹Source: “Experimental Free-space Quantum Teleportation” by Xian-Min Jin, Ji-Gang Ren, Bin Yang, Zhen-Huan Yi, Fei Zhou, Xiao-Fan Xu, Shao-Kai Wang, Dong Yang, Yuan-Feng Hu, Shuo Jiang, Tao Yang, Hao Yin, Kai Chen, Cheng-Zhi Peng & Jian-Wei Pan, *Nature Photonics*, Volume 4 (2010) pp. 376–381.

Unfortunately, such a measurement is provably impossible! The Heisenberg Uncertainty Principle shows that whenever we try to measure a pair of observables whose corresponding observable operators do not commute, the product of the uncertainties in the expected values of the two operators is greater than a definite minimum value. This is the case for position and momentum observables, because the position observable, X , does not commute with the momentum observable, P , and in fact $[X, P] = i\hbar$ from which one can deduce (as we will show below) that $\Delta X \Delta P \geq \frac{\hbar}{2}$. Consequently, teleportation seemed doomed to fail because you could never obtain *complete* information about the original object sufficient to re-synthesize it perfectly elsewhere.

12.1.1 Heisenberg Uncertainty Principle

To understand where the Heisenberg Uncertainty Principle comes from, consider any pair of observables represented by Hermitian operators, A , and B . We are interested in quantifying the uncertainties with which we can know the values of these observables simultaneously. We characterize these uncertainties via their mean square deviations. Starting with the operators:

$$\begin{aligned}\Delta A &= A - \langle A \rangle \\ \Delta B &= B - \langle B \rangle\end{aligned}\tag{12.1}$$

as the deviations of A and B from their true means, squaring gives us:

$$\begin{aligned}\langle (\Delta A)^2 \rangle &= \langle A^2 \rangle - \langle A \rangle^2 \\ \langle (\Delta B)^2 \rangle &= \langle B^2 \rangle - \langle B \rangle^2\end{aligned}\tag{12.2}$$

as the mean square deviations. These quantities characterize how uncertain we are in the value of observables A and B .

Next, to obtain our desired formula we can use the Cauchy-Schwarz inequality. For vectors u and v the Cauchy-Schwarz inequality tells us how their inner products are related, namely:

$$\langle u|u\rangle\langle v|v\rangle \geq |\langle u|v\rangle|^2\tag{12.3}$$

Setting $|u\rangle = \langle \psi|(\Delta A)^\dagger$ and $|v\rangle = \Delta A|\psi\rangle$ the Cauchy-Schwarz inequality implies:

$$\underbrace{\langle \psi|(\Delta A)^\dagger}_{\langle u|} \underbrace{(\Delta A)|\psi\rangle}_{|u\rangle} \underbrace{\langle \psi|(\Delta B)^\dagger}_{\langle v|} \underbrace{(\Delta B)|\psi\rangle}_{|v\rangle} \geq \underbrace{|\langle \psi|(\Delta A)^\dagger(\Delta B)|\psi\rangle|^2}_{\langle u|} \underbrace{|\psi\rangle}_{|v\rangle}\tag{12.4}$$

However, as A is Hermitian and $\langle A \rangle$ is a real number, $\Delta A = A - \langle A \rangle$ must be Hermitian too and so $(\Delta A)^\dagger(\Delta B) = \Delta A \Delta B$. Hence, we obtain

$$\langle (\Delta A)^2 \rangle \langle (\Delta B)^2 \rangle \geq |\langle \Delta A \Delta B \rangle|^2\tag{12.5}$$

So far so good, but to make progress we now need to say something about $\Delta A \Delta B$. To do so, let us split it into two equal terms and insert zero written in just the right way. Namely,

$$\begin{aligned}\Delta A \Delta B &= \frac{1}{2} \Delta A \Delta B + \frac{1}{2} \Delta A \Delta B \\ &= \frac{1}{2} (\Delta A \Delta B - \Delta B \Delta A) + \frac{1}{2} (\Delta A \Delta B + \Delta B \Delta A) \\ &= \frac{1}{2} [\Delta A, \Delta B] + \frac{1}{2} \{\Delta A, \Delta B\}\end{aligned}\tag{12.6}$$

which shows $\Delta A \Delta B$ can be written as the sum of commutator $[\Delta A, \Delta B]$ and an anti-commutator $\{\Delta A, \Delta B\}$. The significance of this is that the commutator of two Hermitian matrices is itself anti-Hermitian, i.e., $[\Delta A, \Delta B]^\dagger = -[\Delta A, \Delta B]$, and the expectation value of an anti-Hermitian operator is purely *imaginary*. Conversely, the anti-commutator of two Hermitian matrices is itself Hermitian, i.e., $\{\Delta A, \Delta B\}^\dagger = \{\Delta A, \Delta B\}$, and the expectation value of an Hermitian operator is purely *real*. Thus, from (12.5) and (12.6) we see that:

$$\begin{aligned}\langle(\Delta A)^2\rangle\langle(\Delta B)^2\rangle &\geq |\langle\Delta A \Delta B\rangle|^2 \geq \left|\left\langle\frac{1}{2}[\Delta A, \Delta B] + \frac{1}{2}\{\Delta A, \Delta B\}\right\rangle\right|^2 \\ &\geq \frac{1}{4}|\langle[\Delta A, \Delta B]\rangle|^2 + \frac{1}{4}|\langle\{\Delta A, \Delta B\}\rangle|^2 \\ &\geq \frac{1}{4}|\langle[A, B]\rangle|^2 + \frac{1}{4}|\langle\{\Delta A, \Delta B\}\rangle|^2 \geq \frac{1}{4}|\langle[A, B]\rangle|^2\end{aligned}\tag{12.7}$$

where we have used $[\Delta A, \Delta B] = [A, B]$. Thus, we arrive at the Heisenberg Uncertainty Principle:

Heisenberg’s Uncertainty Principle For any two hermitian operators, the product of the uncertainties in their values always satisfies the inequality:

$$\langle(\Delta A)^2\rangle\langle(\Delta B)^2\rangle \geq \frac{1}{4}|\langle[A, B]\rangle|^2\tag{12.8}$$

Thus, if teleportation requires that an object be scanned to ascertain (say) the position and momentum of all the particles which comprise it, then as the observables for position and momentum do not commute, i.e., as $[X, P] = i\hbar$, we have (setting $A = X$ and $B = P$) $\langle(\Delta X)^2\rangle\langle(\Delta P)^2\rangle \geq \frac{1}{4}|i\hbar|^2$, which implies the more famous Heisenberg Uncertainty Relation $\Delta X \Delta P \geq \frac{\hbar}{2}$. It is therefore, as a matter of physical *principle*, quite impossible to determine, simultaneously, the exact position and exact momentum of all the particles in an object. Hence, the Heisenberg Uncertainty Principle appeared to rule the possibility of true physical teleportation given that the (presumed) scanning step it must involve is physically impossible.

12.2 Principles of True Teleportation

The situation changed in 1993 when, in a paper whose author list reads like a “Who’s Who?” of quantum information theory, Charles Bennett, Gilles Brassard, Claude Crepeau, Richard Jozsa, Asher Peres, and William Wootters, showed how to exploit entangled states and non-local influences, to circumvent the limitations of the Heisenberg Uncertainty Principle and teleport an arbitrary—even unknown—quantum state between two locations in such a manner that the state did not traverse the intervening distance [49]. The technique transfers the quantum state of the particle to be teleported to another remote particle without the original particle having to traverse the intervening distance. However, in the process, the quantum state of the original particle is necessarily destroyed and that of the receiving particle becomes a perfect reincarnation of the original. Quantum teleportation is therefore distinct from “faxing”, which would leave the original intact and transmit an approximate copy of it over the intervening distance. It is also distinct from “cloning”, which would leave the original intact and create a perfect copy. Obviously, the notion of teleporting a quantum state of a simple particle is considerably less ambitious than teleporting an entire human being from one place to another, but it is a start and has been demonstrated experimentally to increasing degrees of sophistication [38, 67, 68, 279, 340, 374, 418, 503].

We emphasize that quantum physics dictates that the state of the original particle has to be destroyed during the teleportation operation, otherwise teleportation would produce a perfect copy of the original (unknown) quantum state and this would violate the “no cloning” theorem (see Sect. 11.6). This marks a slight distinction from science fiction accounts of teleportation wherein defective teleporters are apt to create perfect clones.

Note also, that in quantum teleportation it is not the *particle* that is teleported, but rather its quantum *state*. However, if the original particle holding the state, is of exactly the same type as the particle onto which that state will be teleported then, as elementary particles such as electrons have identical properties, the net effect of transferring a quantum state from one electron (say) to another remote electron will appear, to all intents and purposes *as if* the electron itself had been teleported. Charles Bennett, one of the inventors of quantum teleportation, made the humorous collage shown in Fig. 12.1 of himself with co-inventor Richard Jozsa using photographs taken as they passed through the (real) Tokyo Teleport station.

12.2.1 Local Versus Non-local Interactions

As we shall see shortly, quantum teleportation is very much dependent on certain so-called non-local physical effects. So we need to take a brief detour to consider what this means.

A *local interaction* is one that involves direct contact, or employs an intermediary that is in direct contact. The forces with which we are familiar in everyday life, such



Having persuaded their co-authors to go first to establish safety and efficacy, Jozsa and Bennett depart Japan using the new facility.

Fig. 12.1 Photographic collage courtesy of Claude Crepeau showing Richard Jozsa and Charles Bennett at the Tokyo Teleport station. Crepeau, Jozsa, and Bennett were three of the inventors of quantum teleportation. Photograph provided courtesy of Charles Bennett

as friction and gravity, are local interactions. With friction, the physical contact between two bodies is really mediated by an electromagnetic field, which in turn comes about by the action of an intermediary, the carrier of the electromagnetic force, called the photon. Photons travel at the speed of light, which although fast is still finite. Consequently, electromagnetic influences cannot propagate faster than the speed of light in a vacuum. Moreover, electromagnetic forces tend to weaken the farther you go from the source.

Locality does not necessarily imply “nearby,” however. Gravity, for example, is a force that exerts its influence over astronomically large distances. Nevertheless, gravity is still regarded as a local interaction because it is mediated by particles, called gravitons, which travel between gravitating objects. It too drops off in strength as the distance between the gravitating objects increases and cannot travel faster than the speed of light.

An important corollary of local interactions is the following: if two events occur in regions of spacetime such that no signal, not even one traveling at the speed of light, could ever reach one region from the other, these two events ought to be completely independent of one another. Why? Because if no signal could ever travel from one region to the other, how could what happens in one region ever be communicated to the other? In fact, special relativity has a special name for two such regions: it says that they are “spacelike separated.”

In short, local interactions can be characterized by three criteria: they are mediated by another entity, such as a particle or field; they propagate no faster than the speed of light; and their strength drops off with distance. Thus the assumption of “locality” allows one to infer that events in spacelike separated regions ought to be independent of one another.

Scientists have shown that all the known forces in the Universe, the electromagnetic, the gravitational, the strong, and the weak forces are all *local*, in this sense. One might think, therefore, that is an end to it, and that reality must be local. After all, if *all* the known forces are local, what is left to be non-local?

Well, what is left is the “collapse of the state vector.” State vectors, as we discussed earlier, provide the mathematical description of quantum systems. When we make measurements, the state vectors collapse into eigenstates, at least according to the Copenhagen interpretation of quantum theory. Now the intriguing point is that there is nothing in quantum theory that explains, mediates, or determines the exact mechanism of the collapse. In particular, the collapse of a state vector involves no *forces* of any kind. This lack of reliance upon a force of any kind, provides quantum theory with an “out”; a way to evade the strictures of locality.

How exactly would a non-local influence be defined? We can just negate each criterion for a local interaction to say that a *non-local interaction* is an interaction that is *not* mediated by anything, is *not* limited to acting at the speed of light, and does *not* drop off in strength with distance. Thus non-local interactions would appear to be magic! The question is—do they exist?

12.2.2 Non-locality: Einstein’s “Spooky Action at a Distance”

“That one body may act upon another at a distance through a vacuum without the mediation of anything else . . . is to me so great an absurdity, that I believe no man, who has in philosophical matters a competent faculty for thinking, can ever fall into.”

– Isaac Newton²

Many scientists have an instinctive distaste for non-local interactions. Certainly, they would seem to be in direct conflict with Einstein’s Theory of Special Relativity which says that nothing can travel faster than the speed of light. Indeed, it was the discrepancy between the predictions of relativity and quantum theory concerning the correlations between events in spacelike separated regions that led Albert Einstein, Boris Podolsky, and Nathan Rosen to point out an effect (thereafter known as the EPR effect) whereby one part of an entangled quantum system appears to instantaneously influence another.

To Einstein, Podolsky and Rosen such non-local influences seemed implausible, and they sought to use their seeming absurdity to prove that quantum mechanics gave only an incomplete account of physical reality. In particular, as Special Relativity held that nothing could travel faster than light, they believed that the correlations in measurement outcomes of experiments measuring both members of greatly separated entangled particles were more plausibly explained by hypothesizing that the pairs of particles were not really entangled at all but rather had fixed values of all their measurable attributes from the outset. Thus the experimental outcomes were really being determined by “hidden variables”. It was out ignorance of these hidden

²Source: [369].

variables that made it appear that the states became definite upon being measured rather than the existence of any instantaneous, unmediated, arbitrarily far separated, “non-local” interactions.

12.2.3 Bell’s Inequality

Now here comes the twist. It could be argued that it is simply a matter of philosophical *taste* as to whether you believe the quantum account or the hidden variable account of how the two entangled photons come to have correlated polarization states upon being measured. But what if there were some experimentally testable difference between the predictions of the two theories—then perhaps a physical experiment could resolve a philosophical question?

In the 1960s John Bell, an Irish physicist on leave from CERN (the European Center for Nuclear Research) showed that there was an empirically testable difference between the predictions of any hidden variable theory and the predictions of quantum mechanics. The test relies upon the statistics obtained when collecting data on the outcomes of pairs of polarization measurements on spacelike separated entangled particles when the polarizers are oriented at certain angles to one another.

Just what would we see if we performed a set of pairs of polarization measurements? For clarity let us suppose that the pair of photons exist in an entangled state such that both polarizations are guaranteed to be the same but are otherwise indefinite until they are measured.

Let’s call our experimenters Alice and Bob, and let’s suppose that they agree to orient their polarizers in the same direction. Thus the angle between their polarizers is 0° . What would Alice and Bob discover? Well, since the entangled particles we are dealing with are perfectly correlated, every time Alice observes a “vertical” Bob also observes a “vertical”. And every time Alice observes a “horizontal”, Bob also observes a “horizontal”. The fraction of times that they agree on the measurement outcomes is 1, i.e., always.

Now let’s imagine what would happen if Bob rotated his polarizer through 90° . Now what looks like “vertical” to Bob is actually seen as “horizontal” by Alice. So now when Alice and Bob perform polarization measurements on respective pairs of correlated photons, their results will be perfectly *anti*-correlated. Every time Alice sees “vertical” Bob sees “horizontal” and vice versa. The fraction of times that they agree on the outcomes will be 0, i.e., never.

So far so good. Now suppose Bob rotates his polarizer back towards Alice’s vertical so that Bob’s polarizer now makes an angle of θ_{12} to Alice’s vertical. This is where things get interesting. Suppose Alice measures her photon to be “vertical”. Thus the twin photon will be “vertical” (in Alice’s basis) too. To Bob however, the photon he receives will appear to be a superposition of his “horizontal” and “vertical” orientations. As a result the outcome of Bob’s polarization measurement is not certain: sometimes when Bob measures a photon that Alice sees as vertical Bob will obtain “vertical” too. But at other times when Bob measures a photon

Alice sees as “vertical” Bob will see “horizontal”. The net effect is that the fraction of times Alice and Bob agree is now somewhere between 0 (never) and 1 (always) the exact number being dependent on the angle, θ_{12} , between Bob’s and Alice’s polarizers.

The question is, what degree of correlation would we expect to see in the outcomes of the polarization measurements made by Alice and Bob? To answer this quantitatively, suppose that Alice and Bob’s polarizers are oriented in parallel planes so that what Alice thinks of as “vertical” is at angle θ_1 degrees with respect to some reference line, and what Bob thinks of as “vertical” is at θ_2 degrees with respect to the same reference line. Hence, the angle between Alice and Bob’s vertical axes is $\theta_{12} = \theta_2 - \theta_1$.

When Alice and Bob make polarization measurements on successive pairs of entangled photons they each obtain either “vertical” or “horizontal” in their respective frames. This means that the quantum state of the joint system can be written as a superposition over product states of polarization outcomes, i.e., $\{|\psi_{xy}\rangle\} \equiv \{|\psi_x\rangle \otimes |\psi_y\rangle\}$, where x and y are vertical or horizontal polarizations as perceived by Alice and Bob respectively. Using geometric arguments we can determine the projection of Bob’s basis vectors onto Alice’s basis vectors allowing us to write:

$$\begin{aligned} |\Psi\rangle &= \frac{1}{2} \cos^2 \theta_{12} |\psi_{v_1 v_2}\rangle + \frac{1}{2} \sin^2 \theta_{12} |\psi_{v_1 h_2}\rangle \\ &\quad + \frac{1}{2} \sin^2 \theta_{12} |\psi_{h_1 v_2}\rangle + \frac{1}{2} \cos^2 \theta_{12} |\psi_{h_1 h_2}\rangle \end{aligned} \quad (12.9)$$

Hence the probabilities, P_{xy} , of Alice finding photon 1 in polarization x , and Bob finding photon 2 in polarization y are:

$$P_{v_1 v_2} = |\langle \psi_{v_1 v_2} | \Psi \rangle|^2 = \frac{1}{2} \cos^2 \theta_{12} \quad (12.10)$$

$$P_{v_1 h_2} = |\langle \psi_{v_1 h_2} | \Psi \rangle|^2 = \frac{1}{2} \sin^2 \theta_{12} \quad (12.11)$$

$$P_{h_1 v_2} = |\langle \psi_{h_1 v_2} | \Psi \rangle|^2 = \frac{1}{2} \sin^2 \theta_{12} \quad (12.12)$$

$$P_{h_1 h_2} = |\langle \psi_{h_1 h_2} | \Psi \rangle|^2 = \frac{1}{2} \cos^2 \theta_{12} \quad (12.13)$$

where $\cos^2 \theta_{12} = \cos^2(\theta_2 - \theta_1)$. Notice that the probabilities for the possible outcomes add up to 1.

A particularly interesting situation arises when Alice and Bob are so far apart that no signal, even one traveling at the speed of light, can possibly reach Bob from Alice and vice versa in the time taken for Alice and Bob to complete their measurements of the polarization orientations of their respective photons. On commonsense grounds (as Einstein, Podolsky and Rosen would see it) the fact that Alice and Bob are spacelike separated means that outcome of Alice’s measurement should not affect the outcome of Bob’s measurement. Based on this assumption, *which amounts to assuming reality is local*, it is possible to derive an inequality that says how the

pairs of measurement outcomes Alice and Bob see should be related to one another when Alice and Bob set their polarizers at various pairs of orientations.

To obtain the inequality let us introduce a third polarizer having its polarization axes oriented along v_3 and h_3 rotated through angle θ_3 with respect to the same common reference frame as the first two polarizers. Using a classical viewpoint, in which reality is assumed to be local, in which case Alice and Bob's measurements ought not to affect one another whenever they are spacelike separated) standard probability arguments would predict:

$$P_{v_1 h_2} = P_{v_1 h_2 v_3} + P_{v_1 h_2 h_3} \quad (12.14)$$

where the right hand side has taken into account the two possible outcomes for the third polarization measurement. Similarly, for other combinations of measurement outcomes we have:

$$P_{v_2 h_3} = P_{v_1 v_2 h_3} + P_{h_1 v_2 h_3} \quad (12.15)$$

and

$$P_{v_1 h_3} = P_{v_1 v_2 h_3} + P_{v_1 h_2 h_3} \quad (12.16)$$

From these relations it follows that:

$$P_{v_1 h_2} \geq P_{v_1 h_2 h_3} \quad (12.17)$$

and

$$P_{v_2 h_3} \geq P_{v_1 v_2 h_3} \quad (12.18)$$

from which it follows

$$P_{v_1 h_2} + P_{v_2 h_3} \geq P_{v_1 h_2 h_3} + P_{v_1 v_2 h_3} \quad (12.19)$$

or more simply

$$P_{v_1 h_2} + P_{v_2 h_3} \geq P_{v_1 h_3} \quad (12.20)$$

which is Bell's inequality. Said more plainly in words:

Bell's Inequality The fraction of times that Alice observes “vertical” and Bob observes “horizontal” when Alice's polarizer is at θ_1 and Bob's polarizer is at θ_2 plus the fraction of times that Alice observes “vertical” and Bob observes “horizontal” when Alice's polarizer is at θ_2 and Bob's polarizer is at θ_3 *must be greater than or equal to* the fraction of times that Alice observes “vertical” and Bob observes “horizontal” when Alice's polarizer is at θ_1 and Bob's polarizer is at θ_3 .

Thus, Bell's inequality is a statement about the correlations between probabilities (and hence frequencies of outcomes) of various polarization results when we perform such an experiment that depends upon the orientations of the three polarization detectors. The inequality is derived on the assumption that if Alice and Bob are sufficiently well separated so that no signal, not even one traveling at the speed of light, could propagate between Alice and Bob within the time-frame of the experiment, then nothing that Alice does can affect Bob and vice versa.

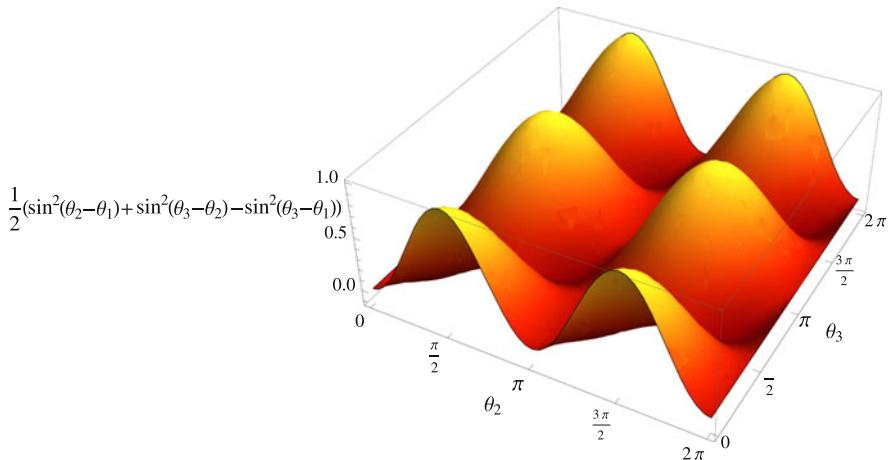


Fig. 12.2 Graphical illustration of Bell’s Inequality

Upon expanding out the definitions of these probabilities explicitly, Bell’s inequality (12.20) becomes:

$$\frac{1}{2} \sin^2(\theta_2 - \theta_1) + \frac{1}{2} \sin^2(\theta_3 - \theta_2) \geq \frac{1}{2} \sin^2(\theta_3 - \theta_1) \quad (12.21)$$

If reality is “local”, Bell’s inequality should always hold regardless of the angles at which we set the polarization detectors. In this case the left hand side ought always to be greater than or equal to the right hand side. However, if we fix (say) $\theta_1 = 0^\circ$ and plot the difference between the left and right hand sides of Bell’s inequality, we obtain the surface shown in Fig. 12.2. If Bell’s inequality holds, this surface ought to touch or be above the (θ_2, θ_3) -plane at height 0, but never below it. However, by introducing a plane that cuts the surface at height 0, and then rotating the surface so we can view it from below, we see that indeed there are portions of the surface that are below zero. This means that quantum mechanical reasoning implies that there are values at which the polarizer orientations can be set that will cause a violation of Bell’s inequality! So which theory is right—classical reasoning based on pure logic and the (reasonable-sounding) assumption of locality, or quantum mechanics?

12.3 Experimental Tests of Bell’s Inequality

Although John Bell derived his inequality in 1964 it was not until 1972 that anyone attempted to check it experimentally [188]. Part of the delay was due to the inability to build perfect polarization detectors and to coordinate sufficiently closely-timed measurements that no speed of light information could make it from one photon to the other within the duration of pair of measurements. In addition, there was very little interest in “reality” research at the time.

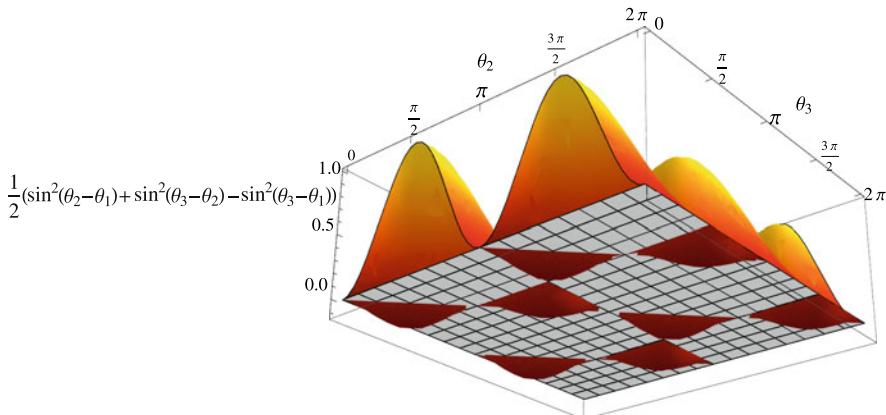


Fig. 12.3 Graphical illustration of violations of Bell's Inequality. In this figure the *vertical axis* is the function

John Clauser, a young researcher at Columbia University was different. Clauser took the reality question seriously. To him, and a growing number of physicists since, it really does matter what is going on behind the mathematical veneer of quantum mechanics. Calculational adequacy alone doesn't cut it. Most physicists became physicists precisely because they wanted to understand how the Universe worked. Comprehension, rather than calculation, was their overriding motivation. However, a physicist's training discourages philosophical musings in favor of prowess in calculation. This is partly cultural as the eminent Austrian physicist Anton Zeilinger has observed:

“[...] there was and still is a tradition in Europe of philosophical thinking among physicists. I saw that in 1977 when I went to America for the first time. Already after a couple of weeks I started to miss philosophical discussion. Here we're more ready to ask really fundamental questions. In Europe it's important to question things. In America it's important to be able to build something. I don't mean that at all negatively.”

— Anton Zeilinger³

The results of Clauser's experiment [188], and even more convincing versions performed later by Alain Aspect, Philippe Grangier, Gérard Roger, and Jean Dalibard [22–24] confirmed the result shown in Fig. 12.3. When one does the experiment one finds that there are indeed certain settings of the angles of the polarizers at which Bell's Inequality is *violated*. Thus the inequality is *wrong*. This means that there must be a mistake in the reasoning under which the inequality was derived. However, the only assumption that was used was the assumption of locality, i.e., events in spacelike separated regions ought not to be able to influence one another. Hence, the assumption of locality must be wrong.

Thus the Clauser and Aspect experiments provide strong *experimental* evidence that reality is non-local. In fact, rather than non-local influences being rare and es-

³Source: [561].

oteric events, quite the contrary, every time particles interact with one another their quantum states tend to entangle. Subsequently, when one member of the pair is “measured” the other member behaves as if it too had been measured, and acquires a definite quantum state also. Thus, non-local influences are not the exception they are the rule. We don’t notice them in our macroscopic world because we never have occasion in the everyday world to deal with spacelike separated events. But if we could scale the quantum world up to larger proportions these exotic quantum states should be quite evident.

Remarkably such a scaling up has been performed since the original Clauser and Aspect experiments and the phenomenon of non-locality has been shown to persist over much greater distances [493] and the potential so-called “locality” and “detector” loopholes in the original experiment have been closed [21, 346, 426, 526]. Thus, it does indeed appear that Nature *is* non-local and the parts of an entangled system can display correlations that are much stronger than can be accounted for by assuming they always had some definite values from the outset, i.e., were classically correlated.

12.3.1 Speed of Non-local Influences

Strictly speaking the experimental tests proving violations of Bell’s inequality only prove that no influence traveling the speed of light (or less) could be responsible for enforcing the observed non-local correlations. However, a philosophical possibility (if not a physical possibility), is the possibility that something (let us call it a “non-local influence”) could be traveling faster than the speed of light between the spacelike separated polarization measurements, and these explain how one part of a system can affect the other. How can we test that? Can we place a lower bound on the speed of propagation of such hypothetical influences (assuming they exist)?

Any hypothetical non-local influence has its speed defined in some preferred frame of reference, which is different from the local latitude/longitude frame of the rotating Earth. Thus an experiment that appears fixed with respect to the Earth’s surface (i.e., in a latitude/longitude frame) would not be in a fixed orientation with respect to this hypothetical preferred frame. As the Earth rotated during the course of a day the two frames could not possibly be aligned at all times. When the Earth frame was not aligned with the preferred frame, then events that would be simultaneous in the preferred frame would not be simultaneous in the Earth frame. Thus, if a Bell inequality would be violated at simultaneous polarization measurement events in the preferred frame it would not be violated in the Earth frame too as those polarization measurement events would not be simultaneous in the Earth frame, and so the visibility of the interference fringes (in the Earth frame) should disappear.

In 2008, in an experimental tour de force, Swiss physicists Daniel Salart, Augustin Bass, Cyril Branciard, Nicolas Gisin and Hugo Zbinden performed such an extended two photon interface experiment over a 24 hour period between two villages in Switzerland that were oriented along a roughly east-west route, as shown



Fig. 12.4 Experiment to place a lower bound on the speed of quantum information. Pairs of correlated photons were created in Geneva. One member of each pair was sent to the Satigny and the other to Jussy—a pair of villages aligned in an East-West direction—over fiber-optic links of exactly equal length. Tests of Bell's inequality violations were performed continuously over a 24 hours period as the whole experiment rotated with the Earth with respect to a hypothetical reference frame in Space. The observation of persistent Bell inequality violations regardless of the time of day confirms both that “standard” result that the quantum correlations are greater than any classical correlation could be (and so the correlations cannot be accounted for as merely arising from a common cause) but, more importantly, that if there is any nonlocal influence passing between the two receiving stations then, given the timing resolution of the experimental equipment, its speed must be greater than ten thousand times the speed of light. However, quantum mechanics does not predict such nonlocal influences propagate between the receivers (because nonlocal effects are unmediated), and this experiment does nothing to confirm that they exist. Rather the experiment proves that *if* such nonlocal influences travel between the receivers they would have to travel much faster than light, if not instantaneously

in Fig. 12.4. The east-west alignment meant that as the Earth rotated the experiment essentially scanned through all possible orientations for the hypothetical preferred reference frame within a 24 hour period. If there is a preferred frame, then it should be revealed by seeing (in the rotating Earth frame) periodic times when Bell's inequality is violated and times when it is not violated over any continuous 24 hour period.

However, when Salart et al. performed their experiment two-photon interference fringes were observed throughout the full 24 hour period with a visibility at all times far exceeding the threshold set by Bell's inequality. This implies that there is no preferred frame for non-local effects.

Moreover, assuming non-local influences propagated at all (which is neither predicted by quantum mechanics nor implied by the Salart et al. experiment) then the experimental results showed that their speed must be at least ten-thousand times the speed of light! So $10,000 c$ can be regarded as a lower bound for the speed of propagation of these hypothetical non-local influences.

Last but not least, the Salart et al. experiment provided yet another confirmation violations of a Bell inequality—and hence non-local effects—over a distance of approximately 18 km.

Given the apparent reality of long distance entanglement, and non-local effects, can we put these phenomena to use? In the next section we show that the answer is a resounding YES!

12.4 Quantum Teleportation Protocol

The basic idea is that Alice wishes to send Bob a qubit that is in a state unknown to her, but she does not want to transmit it through the medium between herself and Bob. If Alice and Bob had met face to face previously and had each retained one member of an entangled pair of particles, Alice can accomplish her desired state transfer by the process of quantum teleportation.

Thus quantum teleportation depends crucially on Alice and Bob each having possession of one end of an entangled pair of particles. Such shared prior entanglement could take many forms. For example, Alice and Bob might each be in possession of any of the following maximally entangled Bell pairs:

$$\begin{aligned} |\beta_{00}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\beta_{01}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\beta_{10}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\beta_{11}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \end{aligned} \quad (12.22)$$

These states can be summarized in a single equation as:

$$|\beta_{xy}\rangle = \frac{1}{\sqrt{2}}(|0, y\rangle + (-1)^x|1, 1-y\rangle) \quad (12.23)$$

Such a state can be synthesized using a quantum circuit such as that shown in Fig. 12.5. To obtain different Bell states, one need only input different combinations of computational basis states, $|x\rangle|y\rangle$, in order to obtain $|\beta_{xy}\rangle$.

Let us suppose that Alice and Bob each possess one particle from the Bell state pair $|\beta_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$. This state is known as a “singlet” state and has a net spin of zero.⁴ It is an especially interesting Bell state because it retains the same basic form under any unitary transformation.

The qubit Alice wishes to teleport to Bob may be assumed to be in state $|\psi\rangle_1 = a|0\rangle_1 + b|1\rangle_1$ such that $|a|^2 + |b|^2 = 1$, but we assume Alice is ignorant of the values of a and b . Hence, we can say $|\psi\rangle_1$ is “unknown” to Alice. This pre-

⁴The “singlet” name refers to the fact that the quantum number M_S can only take on a single value $M_S = 0$ when the net spin is $S = 0$ as it is for $|\beta_{11}\rangle$. Contrast this with the Bell state $|\beta_{01}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$, which is known as a triplet state because the quantum number M_S can take on three values, namely, $M_S = -1, 0, +1$, when the net spin is $S = 1$ as it is for $|\beta_{01}\rangle$.

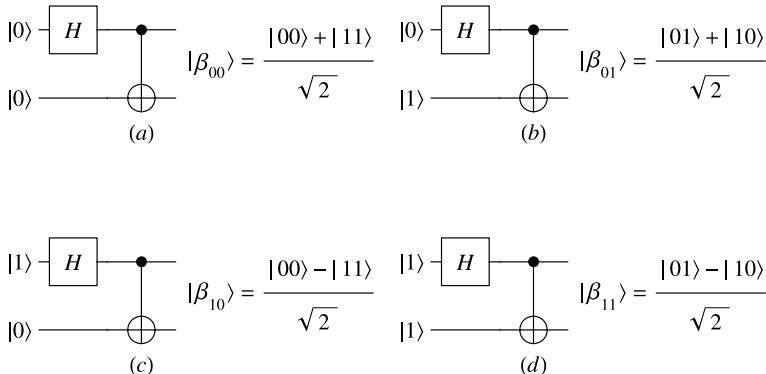


Fig. 12.5 Quantum circuit for synthesizing each of the four Bell states starting from different combinations of computational basis states

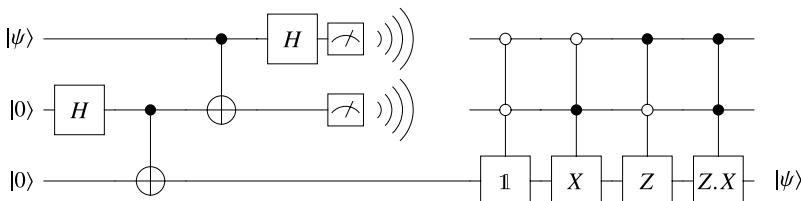


Fig. 12.6 Quantum circuit for teleporting an unknown quantum state from Alice to Bob. The protocol begins with Alice creating an entangled pair of particles in the Bell state \$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\$. She retains one of these qubits and sends the other to Bob. Next Alice performs a Bell basis measurement between the qubit she wishes to teleport and the particle she retained which is entangled with a particle in Bob's possession. After the measurement Alice obtains two classical bit values that she passes to Bob. Upon receipt Bob performs a rotation of the particle he obtained from Alice conditional on the values of the two bits he received from Alice. This conditional rotation transforms Bob's particle into an exact replica of the state Alice wished to teleport. In the process Alice's state has been destroyed locally due to the Bell basis measurement Alice made. Hence, Bob obtains the state that was originally in Alice's possession without that state traveling through the intervening space between Alice and Bob

vents Alice from measuring the state to confirm its identity, and if she attempts to measure the state without choosing the right basis, her attempt will perturb the state dramatically.

So instead, Alice transmits the quantum information defining the unknown state to Bob using the non-local correlations established by the shared Bell state, and two bits of classical communication. The scheme, which is illustrated in Fig. 12.6, works as follows: Initially Alice possesses the state \$|\psi\rangle_1 = a|0\rangle_1 + b|1\rangle_1\$ and Alice and Bob each hold one particle from a singlet state \$|\beta_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle_{23} - |10\rangle_{23})\$. Here we have used subscripts to keep track of which particles we are discussing. Thus there are three particles in all, labeled 1, 2, and 3. Initially, as particle 1 is not

Table 12.1 Alice's measured states and Bob's corresponding corrective actions. N.B. the 1-qubit gates $\mathbb{1}$, X , and Z are all Pauli operators

Alice's state	Alice's measurement	Bob's state	Corrective action	Operators
$ \beta_{00}\rangle_{12}$	00	$a 1\rangle - b 0\rangle$	$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$	$Z.X$
$ \beta_{01}\rangle_{12}$	01	$a 0\rangle - b 1\rangle$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	Z
$ \beta_{10}\rangle_{12}$	10	$a 1\rangle + b 0\rangle$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	X
$ \beta_{11}\rangle_{12}$	11	$a 0\rangle + b 1\rangle$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\mathbb{1}$

entangled with particles 2 and 3, the 3-qubit input state, $|\Psi_{\text{init}}\rangle$, is:

$$\begin{aligned}
 |\Psi_{\text{init}}\rangle &= |\psi\rangle_1 \otimes |\beta_{11}\rangle_{23} \\
 &= (a|0\rangle_1 + b|1\rangle_1) \otimes \frac{1}{\sqrt{2}}(|01\rangle_{23} - |10\rangle_{23}) \\
 &= \frac{a}{\sqrt{2}}|001\rangle_{123} - \frac{a}{\sqrt{2}}|010\rangle_{123} + \frac{b}{\sqrt{2}}|101\rangle_{123} - \frac{b}{\sqrt{2}}|110\rangle_{123}
 \end{aligned} \quad (12.24)$$

Without applying any further physical operation, this state can simply be re-written as:

$$\begin{aligned}
 |\Psi_{\text{init}}\rangle_{123} &= \frac{1}{2}[|\beta_{11}\rangle_{12}(a|0\rangle_3 + b|1\rangle_3) + |\beta_{01}\rangle_{12}(a|0\rangle_3 - b|1\rangle_3) \\
 &\quad + |\beta_{10}\rangle_{12}(a|1\rangle_3 + b|0\rangle_3) + |\beta_{00}\rangle_{12}(a|1\rangle_3 - b|0\rangle_3)]
 \end{aligned} \quad (12.25)$$

Thus if Alice measures particles 1 and 2 in the Bell basis (for which the four states $\{|\beta_{00}\rangle_{12}, |\beta_{01}\rangle_{12}, |\beta_{10}\rangle_{12}, |\beta_{11}\rangle_{12}\}$ are all orthogonal to one another) the state of particle 3 will be projected into a state that bears a simple relationship to the (unknown) quantum state being teleported, i.e., $|\psi\rangle = a|0\rangle + b|1\rangle$.

Specifically, if Alice finds particles 1 and 2 to be in the Bell state $|\beta_{11}\rangle_{12}$, particle 3 will then be in state $a|0\rangle_3 + b|1\rangle_3$. Likewise, if Alice finds particles 1 and 2 to be in state $|\beta_{01}\rangle_{12}$, particle 3 will then be in state $a|0\rangle_3 - b|1\rangle_3$ etc. If Alice communicates the results of her Bell basis measurements to Bob, Bob will then be able to determine what operation to apply to his qubit in order to place it in the (unknown) state $|\psi\rangle$. Table 12.1 lists the operations Bob must perform on his qubit depending on the joint state Alice determines particles 1 and 2 to be in.

The aforementioned steps are summarized in the quantum teleportation protocol:

Quantum Teleportation Protocol

1. Alice wishes to teleport to Bob a single qubit in a pure quantum state, $|\psi\rangle_1 = a|0\rangle_1 + b|1\rangle_1$, which is unknown to her.
2. To do so, Alice creates an entangled pair of particles shared between herself and Bob by feeding the state $|00\rangle_{23}$ into the quantum circuit shown in Fig. 12.5. The net state in Alice's possession is then $|\psi\rangle_1|\beta_{00}\rangle_{23}$

3. Next Alice performs a “Bell basis measurement” on qubits 1 and 2. This is equivalent to applying a CNOT and Hadamard gate to qubits 1 and 2 and then measuring their values in the computational basis to obtain, in output, two classical bits.
4. Alice then transmits these classical bits to Bob using any classical channel of he choosing.
5. Upon receipt, Bob uses the two classical bit values to determine which one of four possible actions he is to perform on the qubit he already received from Alice. The four possible actions are $00 \rightarrow$ no action, $01 \rightarrow$ apply an X rotation, $10 \rightarrow$ apply a Z rotation, or $11 \rightarrow$ apply an $Z \cdot X$ rotation.

12.4.1 Teleportation Does Not Imply Superluminal Communication

It is important to realize that quantum teleportation does not imply superluminal communications. This is perhaps best understood by redrawing the teleportation decoding circuit as in Fig. 12.7 to expose its reliance on *classical* bit values.

The teleportation protocol requires two bits of classical information to be sent from Alice to Bob and these bits cannot be transmitted faster than the speed of light. Moreover, non-local effects between entangled pairs of particles cannot be used for super-luminal communications either, because although the non-local influence is conveyed instantaneously (or at least at speeds in excess of $10,000 c$ —in accordance with Sect. 12.3.1) such links *cannot* be used for communicating an information bearing message. Instead they can only communicate random bits.

Thus, quantum teleportation is a sound physical procedure and does not violate any known law of physics.

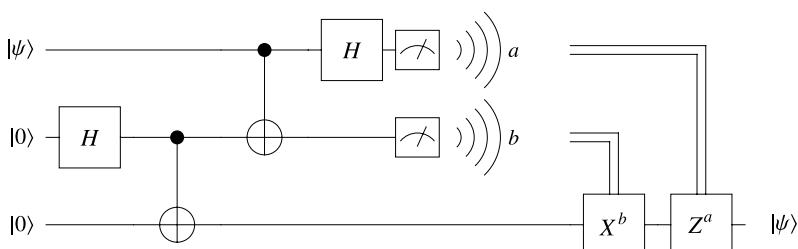


Fig. 12.7 Quantum circuit for teleporting an unknown quantum state from Alice to Bob. This circuit is functionally equivalent to that shown in Fig. 12.6 but emphasizes the fact that the teleportation decoding procedure relies on classical bit values, a and b , to control a pair of quantum gates, which collectively implement the operation $Z^a \cdot X^b$ (X first then Z)

12.5 Working Prototypes

In the late 1990's several working prototypes of quantum teleportation devices were demonstrated. One was built by Dirk Bouwmeester, J.-W. Pan, K. Mattle, Anton Zeilinger, M. Eibl, and H. Weinfurter, in Innsbruck [69] and another was built by Francesco De Martini and collaborators D. Boschi, S. Branca, L. Hardy and S. Popescu in Rome [66], and a third by Jeff Kimble's team at Caltech [194]. There is a little rivalry between the researchers as to which machine constitutes the first *genuine* demonstration of quantum teleportation. But all three schemes are similar in using bench optics components such as beam splitters, parametric down converters, mirrors and photon detectors.

A sketch of Bouwmeester et al.'s set-up is shown in Fig. 12.8. On the left side of Fig. 12.8, Alice sends her "message" photon M , which is prepared in a 45° polarization state, towards a beam splitter, with a specific state, 45 degrees polarization. That is Alice intends to send the quantum state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ to Bob. Simultaneously, two entangled photons, A (shown as "Photon to Alice") and B (shown as "Photon to Bob"), are created and travel in opposite directions: photon A goes to Alice's beam splitter and photon B to Bob's beam splitter. The timings are arranged so that one of the entangled photons arrives at Alice's beam splitter at just the same instant as Alice's message photon M . Some of the time the two photons emerge from Alice's beam splitter in different directions but Alice is unable to distinguish which photon is which. As a result of this indistinguishability, Alice's message photon becomes entangled with photon A . Now neither M nor A has a definite polarization state but they must be opposite since they went to different detectors when they emerged from Alice's beam splitter. However, photon B also had the opposite polarization state to photon A . Therefore, photon B must acquire the same polarization state

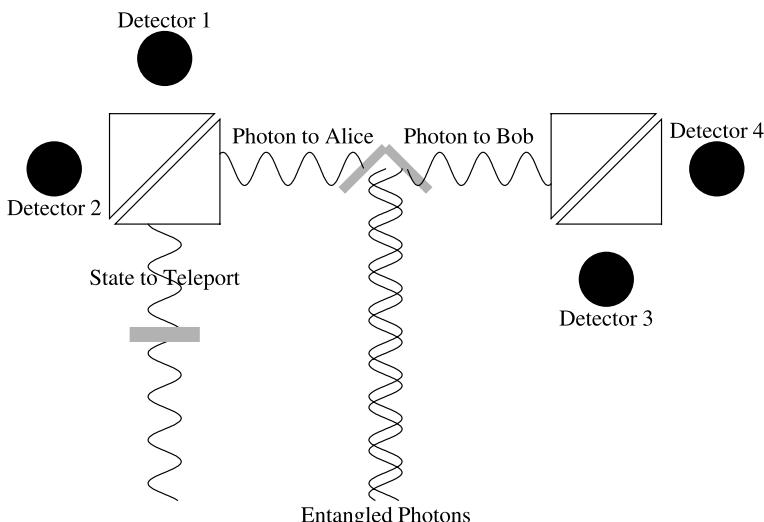


Fig. 12.8 The Innsbruck quantum teleportation experiment

as photon M (the message photon). Hence teleportation is complete and Bob sees photon B has a polarization of 45° .

It was quite a surprise that there was a physical basis for teleportation and an even bigger surprise that the process evolved from a concept to a working prototype in just four years. Who knows what potential this technology has over the coming decades.

12.6 Teleporting Larger Objects

From a technological perspective quantum teleportation is much simpler than even the most rudimentary quantum computation. In fact, in 1997 two groups reported optical schemes in which they successfully teleported an unknown quantum state across a laboratory bench [67, 68]. Scaling quantum teleportation up to the level of an entire human being however, is quite unrealistic at this point. Samuel Braunstein has estimated how much information you would need to transmit in order to perform such a feat. Starting from the observation that the visible human project, sponsored by the American National Institute of Health, requires about 10 Gigabytes of bits (about 10 CD-ROMs) to hold the information needed to describe the full three-dimensional structure of a human to a 1 mm^3 resolution, Braunstein estimates that an entire human could be described, down to the atomic level, using roughly 10^{32} bits. With current communication channel capacities, Braunstein estimates that it would take about a hundred million centuries to transmit this information down a single channel!

However, there have been some interesting advances in quantum teleportation recently, that push it in interesting new directions. Of special note is an experiment by Qiang Zhang, Alexander Goebel, Claudia Wagenknecht, Yu-Ao Chen, Bo Zhao, Tao Yang, Alois Mair, Jörg Schmiedmayer, and Jain-Wei Pan, showing that it is possible to teleport a multi-particle entangled state [563]. The experiment basically doubled the complexity of the regular quantum teleportation circuit, requiring a 6-photon interferometer to transfer the joint polarization state of a pair of photons in such a manner that their entanglement was preserved under the teleportation operation. This shows that it is possible to teleport the quantum state of objects that are more complex than single qubits. This is a step in the direction of teleporting the state of a complete molecule.

Another key advance is an experiment by Jacob Sherson, Hanna Krauter, Rasmus Olsson, Brian Julsgaard, Klemens Hammerer, Ignacio Cirac, and Eugene Polzik, showing teleportation of a quantum state between light and matter, i.e., objects of dissimilar type [454]. The significance of this is that there is currently much interest in using photons to convey quantum information over long distances, and in using the long-lived collective spin states of ensembles of alkali atoms to store quantum information over relatively long times. Quantum teleportation could be useful in transferring quantum information from flying qubits into stationary qubits by controlled light-matter interactions. If such interfaces can be perfected they could

enable true quantum repeaters that would greatly extend the range of quantum communications in fiber optic cables [83].

Finally, it is worth mentioning, insofar as demonstrations of quantum teleportation rely upon quantum interferometry, that there have been many exciting developments on demonstrating quantum interference using objects as complex as fullerenes, i.e., molecules consisting of a cage of 60 carbon atoms [18, 224]. These experiments are quite remarkable given the relative complexity of the molecules and consist of multi-level quantum systems.

12.7 Summary

This chapter has examined the physical basis for true teleportation—the transmission of a quantum state from *A* to *B* without it having to pass through the intervening medium. In the process the quantum state is necessarily destroyed at the source location and is re-incarnated at the receiving station. The scheme requires shared prior entanglement between the source and receiver, and a classical communications channel over which to pass the two bit result obtained by making a complete Bell basis measurement. For the latter reason, quantum teleportation cannot be achieved super-luminally as the transmission of the classical message through the medium is limited to traveling at the speed of light.

Notice that quantum teleportation teleports the quantum *state* of an object, not the object itself. This is slightly different from the usual science fiction view of teleporting an object. Consequently, we cannot use this scheme to teleport an electron in its entirety from one place to another. Rather, we can teleport the *spin* state of one electron at a particular location to another electron at a different location (or indeed a different kind of particle entirely). The net effect, however, is similar: A particle in a specific state at the source location has its state destroyed and reincarnated on another particle at the destination without the original particle traversing the intervening distance.

We emphasize that the non-local effects that underpin quantum teleportation cannot be used to transmit a content bearing message super-luminally either. At best they would be limited to transmitting random bits. Quantum mechanics neither requires nor predicts these non-local influences propagate through the medium between *A* and *B*. However, a Swiss team showed recently established experimentally that *if they did so propagate* (which is not proven and is frankly unlikely) then they would have to travel in excess of ten thousand times the speed of light. The quantum mechanical prediction that non-local effects should exist between the parts of a spacelike-separated entangled quantum system regardless of the distance between them, the nature of the intervening medium, and without the need for the mediation of any influence of any kind, was established experimentally by Freedman and Clauser [188] and Aspect, Grangier and Roger in [22, 24], and substantiated with much improved experiments later [346, 426, 493, 526]. These results show the quantum mechanical predictions are correct and contrary to the expectations of Einstein, Podolsky and Rosen: reality is, as far as we can tell, non-local.

Thus, quantum teleportation is a physically sound protocol, has been demonstrated many times experimentally, and researchers are inching forward to more teleporting more complex entities. The protocol has been demonstrated in photonic systems, atomic systems and even between the two. However, a teleportation machine of the complexity envisioned by science fiction writers is utterly impossible given current know-how. But that does not mean teleportation is not useful. Indeed it could prove to be key to making practical quantum repeaters, and has been proposed as a primitive operation for quantum computation. More on that in Chap. 15.

12.8 Exercises

12.1 Consider two observables represented by Hermitian matrices, A and B . Prove that their commutator $[A, B] = A \cdot B - B \cdot A$ is anti-Hermitian, and that their anti-commutator, $\{A, B\} = A \cdot B + B \cdot A$, is Hermitian. A matrix, M , is Hermitian iff $M = M^\dagger$ and anti-Hermitian iff $M = -M^\dagger$.

12.2 Consider the two Bell states $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$. Transform each of these states to a new basis. What is the form of each of these states in the U basis? Do you notice any difference?

12.3 Recall the definition of the Bell basis states, $|\beta_{00}\rangle$, $|\beta_{01}\rangle$, $|\beta_{10}\rangle$, and $|\beta_{11}\rangle$ defined by:

$$|\beta_{xy}\rangle = \frac{1}{\sqrt{2}}(|0, y\rangle + (-1)^x|1, 1-y\rangle)$$

and consider an *arbitrary* single qubit pure state defined by:

$$|\psi\rangle = \alpha|0\rangle + \sqrt{1 - |\alpha|^2}|1\rangle$$

such that $|\alpha| \leq 1$.

- (a) Write down a state, $|\psi^\perp\rangle$, which is orthonormal to $|\psi\rangle$, i.e., a state for which $\langle\psi^\perp|\psi\rangle = 0$.
- (b) If α is *purely real* with $-1 \leq \alpha \leq 1$, but otherwise arbitrary, prove that $\frac{1}{\sqrt{2}}(|\psi\psi\rangle + |\psi^\perp\psi^\perp\rangle) = |\beta_{00}\rangle$.
- (c) If α is *complex* with $|\alpha| \leq 1$, but otherwise arbitrary, prove that $\frac{1}{\sqrt{2}}(|\psi^\perp\psi\rangle - |\psi\psi^\perp\rangle) = |\beta_{11}\rangle$.

12.4 In this chapter, we developed the quantum teleportation protocol using a source of entangled pairs of particles with each pair in the state $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$. But this form of entanglement is not essential. Modify the quantum teleportation scheme to use a source of entanglement that produces pairs of particles each in the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

12.5 The states $|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$ and $|W\rangle = \frac{1}{\sqrt{2}}(|001\rangle + |010\rangle + |100\rangle)$ are fundamentally *inequivalent* types of entangled states.

- (a) Can you devise a quantum teleportation scheme that uses $|GHZ\rangle$ as the source of entanglement?
- (b) Can you devise a quantum teleportation scheme that uses $|W\rangle$ as the source of entanglement?

12.6 In many quantum information processing tasks it is useful to “measure a state in the Bell basis”, i.e., a basis consisting of 2-qubit entangled states, $\{|{\beta}_{00}\rangle, |{\beta}_{01}\rangle, |{\beta}_{10}\rangle, |{\beta}_{11}\rangle\}$. However, we often find the state represented, initially, in the computational basis. Thus, it is worthwhile knowing how to switch from the computational basis to the Bell basis. Practice this by rewriting the following states in the Bell-basis:

- (a) An entangled state: $\frac{1}{\sqrt{2}}(|00\rangle + i|11\rangle)$
- (b) An unentangled state: $\frac{1}{\sqrt{2}}(|01\rangle + |11\rangle)$
- (c) A state whose entanglement is unknown $a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$

12.7 In many quantum information processing tasks it is useful to perform operations on a state represented in the Bell-basis, i.e., a basis consisting of 2-qubit entangled states, $\{|{\beta}_{00}\rangle, |{\beta}_{01}\rangle, |{\beta}_{10}\rangle, |{\beta}_{11}\rangle\}$. However, we often find the operator is given, initially, in the computational basis. Thus, it is worthwhile knowing how to map gate specified in the computational basis into the equivalent gate in the Bell basis. Practice this by rewriting the following states in the Bell-basis:

- (a) CNOT
- (b) SWAP
- (c) iSWAP
- (d) Berkeley B

12.8 Suppose Alice and Bob are a pair of Space-faring astronauts who desire to stay in touch with one another over long spaceflights in opposite directions deep into the cosmos. Realizing that speed of light signal delays will pose a challenge they take a crash course in quantum information theory in the hopes of devising a way to use entanglement to overcome speed-of-light signal delays, and thereby keep in contact. Based on their limited understanding of quantum information theory, Alice and Bob believe that the following communication protocol will allow them to communicate superluminally over arbitrarily great distances! They suppose they each start out with an inexhaustible supply of marching EPR particles each in the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Before leaving Alice, and Bob agreed on an order and timing pattern in which to measure their respective EPR particles. Alice promises to measure an EPR particle in her possession (let’s call it her “message ebit”) on the stroke of each minute. If the answer she obtains is the bit she wishes to send to Bob, Alice subsequently measures the next 29 ebits (her “check” bits) in the agreed

upon order on every *even* numbered second (2, 4, 6, . . . , 58) making 29 additional measurements in all.

Likewise, Bob promises to measure the EPR particle matching Alice’s “message ebit” at precisely one second after the start of each minute. In addition, Bob promises to apply a Walsh-Hadamard gate, H , to each of the next 29 ebits in the agreed upon order and then to measure their bit values on every *odd* numbered second (3, 5, 7, . . . , 59) making 29 additional measurements in all.

Alice and Bob believe that this scheme will allow them to communicate one message bit per minute over arbitrarily great distances by reasoning as follows: Each of their EPR pairs begins in the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. When Alice makes her measurement and obtains a 0 or a 1, Bob’s matching EPR particle acquires the *same* bit value *instantaneously* and so reveals the bit Alice obtained. However, Bob does not know whether this was the bit Alice intended to send or just a random bit. To communicate a real message Alice must do something to allow Bob to tell whether the “message ebit” he measured at one second after the start of the minute is the bit Alice *intended* to send. This is the reason for the subsequent 29 measurements. Before Bob makes his measurements, and if Alice has not measured, each of Bob’s particles are equally likely to be measured as “0” or “1”, and hence will be in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. But after applying the Walsh-Hadamard transform, Bob’s particle is rotated into the state $|0\rangle$. Hence, when Bob makes his measurements on the remaining 29 particles he will always obtain a “0” if Alice has not measured her corresponding EPR particles. However, Bob will only obtain a “0” 50% of the time if Alice has measured her 29 “check” particles. Hence, by observing the number of “0”’s he obtains, Bob can determine (with probability $1 - \frac{1}{2^{29}}$) whether or not Alice measured her 29 “check” particles. Hence, Bob learns whether the “message ebit” he measured (at one second after the start of the minute) is or is not the bit Alice intended to send. As this scheme requires no classical communication Alice and Bob can be arbitrarily far apart and be able to communicate classical information at the rate of one bit per minute!

Alas, the aforementioned scheme contains a crucial flaw. Your job is to find the flaw and explain why it prohibits superluminal communication of classical information. The following facts about Walsh = Hadamard gates may be helpful:

$$\begin{aligned} H &= \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \\ H|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ H|1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \tag{12.26}$$

Chapter 13

Quantum Cryptography

“In Nature’s infinite book of secrecy A little can I read”

– William Shakespeare

Over the past two decades society has become more and more reliant on network communications. Today, most important electronic communications are encrypted using public-key cryptosystems, such as the RSA cryptosystem (whose security rests on the intractability of factoring composite integers) [419] and elliptic curve cryptosystems (whose security rests on the intractability of computing discrete logarithms) [425]. However, as we saw in Chap. 6, such cryptosystems can both be broken classically in exponential time using networks of computers [527], or quantumly in polynomial time using a single quantum computer capable of running Shor’s algorithm [455, 458]. Hence messages transmitted using the existing public-key infrastructure are potentially vulnerable to attack in one way or another.

At present a quantum attack might not seem like a serious risk because the difficulty of building a quantum computer of sufficient complexity to run Shor’s algorithm is so challenging that no-one is likely to create one any time soon. However, basing the security of ones’ confidential communications on the *presumed* technical prowess of an adversary is notoriously risky, especially when there are ideas floating around, such as topological quantum computing, that may lower the barrier on building practical quantum computers from imperfect components [27].

Worse still, if today’s encrypted traffic were intercepted, copied, and stored by an adversary they could potentially break such encrypted communications at a future date when classical algorithms or technology has progressed far enough to deliver the required CPU cycles within a reasonable time. If the encrypted communications only have transient value, then perhaps this is unimportant. But certain communications, e.g., medical records or diplomatic communiqués, might need to be kept confidential for several decades. It is questionable whether one can choose a key today sufficient to guarantee a message encrypted with it will remain secure for decades to come, invulnerable to all advances in code-breaking over the intervening time-period.

Fortunately, just as quantum mechanics can take away security via Shor's algorithm so too can it restore it via a technology known as "quantum cryptography".

Quantum cryptography provides a new foundation for an *unconditionally* secure global communications infrastructure, i.e., one whose security does not rely on any assumed limitations concerning the mathematical sophistication, algorithmic ingenuity, or computational resources available to an adversary. Instead, its security rests upon quantum mechanical phenomena, such as the inevitability of changing a quantum state if it is measured in the "wrong" basis, and the impossibility of copying (or cloning) an unknown quantum state.

In this chapter we will look at why we might need a more secure communications infrastructure, describe how quantum cryptography satisfies this need, and look at other quantum cryptographic protocols besides key exchange.

13.1 Need for Stronger Cryptography

Electronic communications in the form of telephone calls, text messages, emails, faxes, banking transactions, and database transactions etc, have become so ubiquitous that is hard to imagine living without them. However, as the communications technology has matured so too has the ability of unintended recipients to tap into those channels and listen in for information of value to them. In the past such intercepts were limited to criminal or blackmail activity, or narrowly focussed lawful interception of specific channels under court order and proper oversight. But as automated data mining technology has advanced, and as anxiety over issues such as global terrorism have grown, widespread trawling of communications has emerged. This raises serious questions for free societies that need to balance the legitimate needs of intelligence and law enforcement agencies with the privacy rights of their citizens. At a higher level, governments are also concerned that such practices could undermine not only their security but also their commercial interests. Are such concerns really justified or is the reality quite different?

13.1.1 Satellite Communications Can Be Tapped

There is ample evidence that sensitive communications are being intercepted and analyzed. For example, in 2001 the European Parliament issued a report "On the Existence of a Global System for the Interception of Private and Commercial Communications (ECHELON Interception System)" [436]. In this report, the European Parliament alleges that there is a global surveillance system called "ECHELON" operated cooperatively between Australia, Canada, New Zealand, the United Kingdom and the United States (known collectively as the "UKUSA" alliance), which grew out of intelligence collaborations that started during the Second World War. The geographic distribution of the partners across the globe makes it especially easy to intercept satellite transmissions as there is at least one partner country within

the footprint of every transmitting satellite, and these satellite transmissions can be monitored without disclosing such monitoring is occurring. However, it appears likely the ECHELON has a far greater reach than this, extending at least to cell phone intercepts.

The main thrust of the European concern appears to be in ensuring compliance with E.U. law, protecting the privacy rights of European citizens, and ensuring that the surveillance system is not being used to conduct industrial espionage to glean intelligence that might be passed to companies in ECHELON member countries to give them an unfair economic advantage when negotiating contracts and deals. The European Parliament report concluded (p. 82, [436]) that:

"To sum up, it can therefore be said that the current legal position is that in principle an ECHELON type intelligence system is not in breach of Union law because it does not concern the aspects of Union law that would be required for there to be incompatibility. However, this applies only where the system is actually used exclusively for the purposes of state security in the broad sense. On the other hand, were it to be used for other purposes and for industrial espionage directed against foreign firms, this would constitute an infringement of EC law. Were a Member State to be involved in such action, it would be in breach of Community law."

To emphasize its concern over industrial espionage, the European Parliament report cited several examples of alleged intercepts and their subsequent leaking to either U.K., U.S., or European companies. In one case, the report claims that the details of a European invention for a new wind turbine were intercepted using ECHELON and passed to a U.S. competitor who filed a U.S. patent before the European inventor could do so, causing the European company to abandon its plans to enter the U.S. market ([436] p. 104). In another case, the European Parliament report claims ECHELON intercepts uncovered evidence for bribery in contract negotiations between a Saudi Arabian customer and a European aircraft company [155, 436] p.103. The report alleges that a U.S. aircraft company also bidding on the contract was informed of the evidence for bribery and ultimately won the contract.

A similar pattern of ECHELON uncovering evidence for bribery and passing it to a U.S. company also competing on the contract was alleged to have been repeated in a third case involving a \$1.4B contract for monitoring the Brazilian rainforest. A European company and a U.S. company were both bidding on the contract. The European Parliament report alleges that ECHELON uncovered evidence for bribery and passed this to the U.S. company, who ultimately won the contract [436] p. 106.

The potential use of ECHELON as a way to level the playing field in the face of corruption is alluded to, but explicitly denied, in remarks made at a press conference given by James Woolsey, the former director of the CIA:

"James Woolsey, the former director of the CIA, said at a press conference [546] he gave at the request of U.S. State Department, that the USA did conduct espionage operations in continental Europe. However, 95% of 'economic intelligence' was obtained by evaluating publicly accessible information sources, and only 5% came from stolen secrets. Espionage was used to secure economic intelligence from other countries where compliance with sanctions and dual-use goods were concerned, and in order to combat bribery in connection with the award of contracts. Such information is not, however, passed to U.S. companies."

Such allegations of industrial espionage are not confined to one direction, however. In the same EU report cases are cited alleging improper access to a U.S. car

company's documents by a European rival [436] p. 105, and the discovery of microphones in the First Class cabin of a European airline to eavesdrop on the conversations of business travelers [436] p. 103.

In cases where illegal intercepts are used to expose corruption in contract negotiations, and (as the EU report alleges but others deny) the information is passed along to commercial competitors [156], it is difficult for either side to assume the moral high ground. However, intercepts are not confined to economic espionage. In 2003 bugging devices were found on the telephone system of the Justus Lipsius building of the European Union (EU). The tapping incident coincided with heated internal debate regarding the strategy the EU should pursue with respect to Iraq [174].

In the face of such incidents, the European sensitivity to the potential competitive disadvantage posed by widespread eavesdropping and intelligence gathering, as well as strategic technology development and scientific pursuit, led the European Union to create a quantum cryptography program called "SECOQC" in 2004 [245]. This program invested 11.4 million euros in developing quantum cryptographic technologies capable of circumventing espionage attempts by systems such as ECHELON [534], by developing protocols for routing, storing, and managing keys within a large mesh network. A prototype version of the system was demonstrated successfully in Vienna in 2008 [246].

13.1.2 Fiber-Optic Communications Can Be Tapped

Most people are already aware that their key strokes and communications are vulnerable to interception when they connect their laptops to wireless networks in common areas such as airports and coffee shops. Such wireless connections have to be set up carefully to ensure such communications are kept private. However, currently, about 99% of the world's long-distance telephone and data traffic is conveyed over fiber-optic cables [347], and we tend to ascribe far greater security to these *wired* fiber-optic connections. Is such complacency justified? Perhaps not.

The explosion in demand for network traffic that accompanied the growth of the internet during the 1990's drove the communications industry to deploy high capacity fiber-optic cables worldwide. The reason is simple: one fiber-optic cable can handle about 128 times as much traffic as a single satellite transponder. Whereas pre-Internet, most international telecommunications traffic was running over satellite links (and could therefore be intercepted without detection by an ECHELON-like system), nowadays most telecommunications traffic is running over fiber-optic networks, which must be tapped directly in order to intercept traffic on them, and potentially reveal the intrusion attempt in the process.

However, in April 2003, Computerworld reported that:

"Fiber-optic cables use light to transmit information and can be easily intercepted, interpreted and manipulated with standard off-the-shelf equipment that can be obtained legally throughout the world. More important, the vast majority of private and public fiber networks don't incorporate methods for detecting optical taps, offering an intruder a relatively safe way to conduct corporate espionage. Commercial intrusion-detection systems and other

IT security systems operate at the data layer and offer no way to identify the existence of physical taps.”

Computerworld – April 8th 2003 [516], p. 2

Such concerns were borne out in 2003 when it was discovered that an eavesdropping device had been installed illegally on one of Verizon's fiber optic networks [175]. It is believed that the device was intended to learn about a forthcoming quarterly statement from a mutual fund company prior to it being announced publicly [443] p. 8.

“Tapping a fiber-optic cable without being detected, and making sense of the information you collect, isn't trivial but has certainly been done by intelligence agencies for the past seven or eight years,” said John Pescatore, an analyst at Stamford, Connecticut-based Gartner Inc. and a former National Security Agency analyst. “These days, it is within the range of a well-funded attacker, probably even a really curious college physics major with access to a fiber-optics lab and lots of time on his hands.”

Computerworld – April 8th 2003 [516], p. 1

It is surprisingly easy to tap an optical fiber. In part this is because fiber optic networks are *designed* to tolerate a certain amount of noise and anomalies in order to ensure the network is robust and reliable. The downside of this is that it makes it easier for an intruder to tap the fiber optic without detection. Commercially available systems for tapping fibers generate less than 3 dB insertion loss and cost less than \$1000. “Professional” grade espionage systems are reported to have insertion losses of less than 0.5 dB [492]. The simplest method is to gain access to the fiber and attach a “clip on coupler” that introduces a bend in the fiber sufficient to allow some, but not all, of the light to leak out. One can then monitor the leaking light using a photodetector and packet sniffer to extract the packets of interest and stitch them together to extract the sought after information.

“In theory, it's easy to find out what's being transmitted along a fiber. “All you have to do is put a little bit of a bend in the fiber and look at the light that comes off it.””

Jim Hayes, President of the Fiber Optic Association – April, 2003 [98]

Other schemes, such as splicing the fiber, are more likely to reveal the attempted eavesdropping because the very act of splicing the fiber will cause a momentary, but noticeable, interruption in service. Nevertheless, there are reports of splicers having been built into some networks, thereby permitting eavesdropping very easily and quite undetectably since there is no apparent loss in signal strength from the inception of the fiber link.

Remarkably, it even appears possible to use specialized submarines to tap optical fibers that are deep underwater. There is a suggestion that:

“[...] the U.S. has been reconfiguring the submarine USS Jimmy Carter for [fiber-optic tapping]”

Attributed to Jeffrey T. Richelson, author of *The U.S. Intelligence Community* (Westview Press, Boulder, Colo., 1999). – April, 2003 [98]

“... former intelligence officials confirmed that NSA technicians used a special submarine to tap into a fiber-optic cable on the seafloor in the mid-1990s—around the same time that fiber amplifiers began displacing electro-optic amplifiers. The sub supposedly had a special compartment into which the cable could be hauled, enabling technicians to install the tap.”

Attributed to Wall Street Journal Online – May 2001 [443]

In this case one has to use sophisticated methods gain access to the optical fiber core while avoiding short-circuiting the accompanying electrical cables that carry the power needed to run the underwater repeater stations. This is doable but not exactly easy:

[...] in a typical cable, the fiber in question is one of a dozen hair-thin strands of glass, which are embedded inside a laser welded, hermetically sealed, 3-mm diameter stainless steel tube. This tube is in turn covered by a few centimeters of reinforcing steel wire and cables carrying 10 kV of dc power, all at a depth of a couple of thousand meters. [...] “It’s not impossible but it certainly pushes the definition of practical.”

Attributed to Jim Hayes, President of the Fiber Optic Association – April, 2003 [98]

13.1.3 Growing Regulatory Pressures for Heightened Security

Just as the tapping of satellite-based communications and fiber-optic communications are on the rise, so too is the practice of transporting highly sensitive data. Government financial institutions may need to exchange confidential messages to coordinate fiscal policy on a global scale; multi-national companies must share technical and economic data across national boundaries; lawyers need to brief far-away colleagues or clients on details of cases; doctors may need to transmit patient medical records to specialists. The potential damage of improper disclosure of such data is immense. This threat has alarmed legislators and propelled them to enact laws requiring companies to take responsibility for ensuring the electronic transport of technical, medical, or client data is properly protected. Such laws will no doubt levy increasingly large penalties and fines on violators who can be shown to have acted negligently. To reduce this exposure companies will need to move to higher standards of security in their electronic communications.

The concern also extends to satellite communications. The 2007 “National Information Assurance Policy for Space Systems used to Support National Security Missions” [442] states that:

National Security Agency (NSA) approved cryptographies and cryptographic techniques shall be used to protect all communications links in applicable USG-owned or controlled space systems from exploitation, corruption, or denial consistent with mission requirements and the projected threat over the life cycles of these space systems. Information system security architectures for applicable space systems shall be coordinated with NSA at their inception, and periodically during their evolution.

13.1.4 Archived Encrypted Messages Retroactively Vulnerable

Clearly, in light of the foregoing reports, the *physical* layer of security of land-based and underwater fiber-optic systems can no longer be assumed. We must therefore add a layer *algorithmic* security to sensitive communications by encrypting confidential data passing through such networks.

Encryption is a mathematical procedure that maps a plaintext message into a scrambled form that is supposed to be unintelligible any unintended recipient, but which can be decrypted easily back to the original plaintext by its intended recipient. A pair of complementary encryption-decryption procedures forms a “cryptosystem”. All such cryptosystems have to balance the need for fast encryption and decryption algorithms (to make the use of the encryption appealing to users) against need for high security (to make it worthwhile). These conflicting needs have driven the communications industry towards using so-called “public-key” cryptosystems.

Public key cryptosystems make use of a matching pair of cryptographic keys called a “public-key” and a “private-key”. The public key is generating by the person wanting to receive a secure message and it is broadcast widely for all to see. Complementary to this public-key, there is a corresponding private-key that is kept secret by the person wishing to receive secure messages. Anyone can encrypt a message using the widely known public key, but only the owner of the matching private key can decrypt it efficiently back to its plaintext.

But just how secure are such public key systems, and can this level of security be relied upon indefinitely? Two popular public key cryptosystems are the RSA cryptosystem [419] and the Elliptic Curve cryptosystem [294, 353]. The encrypting and decrypting stages of both these cryptosystems are very fast and yet they provide an acceptable level of security for the vast majority of routine transmissions. The ECC system was recently approved for use up to “Top Secret” level by the National Security Agency [372].

In order to break either RSA or ECC one has to solve certain mathematical problems that are believed to be intractable using the best known algorithms running on classical computers. Specifically, the security of the RSA public key cryptosystem relies on the presumed difficulty of factoring large composite integers [419, 455]. The best known classical algorithm for factoring an integer n is the Number Field Sieve (NFS), which has a complexity of $\mathcal{O}(e^{c(\log n)^{1/3}(\log \log n)^{2/3}})$, with $1.523 \leq c \leq 1.923$ [396]. This is a super-polynomial (sub-exponential) complexity, and rapidly becomes intractable with increasing size of the integer being factored. To remind you, Table 6.1 shows the historical scaling in the amount of net computer time needed to factor ever larger composite integers. Using generalized NFS on conventional general purpose computers, the largest composite integer to have been factored successfully to date is RSA-200 (a 200-digit/663-bit number) equal to:

$$\begin{aligned}
 & \text{RSA-200} \\
 & = 27997833911221327870829467638722601621070446786955 \\
 & \quad \dots 42853756000992932612840010760934567105295536085606 \\
 & \quad \dots 18223519109513657886371059544820065767750985805576 \\
 & \quad \dots 13579098734950144178863178946295187237869221823983 \tag{13.1}
 \end{aligned}$$

This factoring feat was reported on 9th May 2005 by F. Bahr, M. Boehm, J. Franke, T. Kleinjung and required multiple computers equivalent to approximately 55 years

of CPU effort for a single 2.2 GHz Opteron processor! Using the Generalized Number Field Sieve algorithm, the factors were found to be:

$$\begin{aligned} p = & 35324619344027701212726049781984643686711974001976 \\ & \dots 25023649303468776121253679423200058547956528088349 \end{aligned} \quad (13.2)$$

and

$$\begin{aligned} q = & 7925869954478333033347085841480059687737975857364 \\ & \dots 219960734330341455767872818152135381409304740185467 \end{aligned} \quad (13.3)$$

as can be verified easily by direct multiplication.

Likewise, the security of the elliptic curve public key cryptosystem relies on the presumed difficulty of computing elliptic curve discrete logarithms [489]. The best known classical algorithm for computing the elliptic curve discrete logarithms is the Pollard rho-method [395], which has a complexity of $\mathcal{O}(\sqrt{\pi n}/2) \equiv \mathcal{O}(e^{\frac{1}{2} \log n + \frac{1}{2} \log \frac{\pi}{4}})$ serially and $\mathcal{O}(\sqrt{\pi n}/(2r))$ when parallelized on r processors [489]. Thus, for the same key length the ECC cryptosystem provides a higher level of security than the RSA cryptosystem.

So long as the integer factorization problem and the elliptic curve discrete logarithm problem remain intractable, the RSA and ECC cryptosystems will be secure. However, it is possible that, in the future, a better factoring algorithm or a better discrete logarithm algorithm will be found. Equally, it is possible that special-purpose hardware will be developed that can factor integers or compute discrete logarithms much faster than expected. If the value of the confidential data exchanged is transient, then the level of security of existing public key cryptosystems may be acceptable for routine commercial, governmental, and military communications because these cryptosystems do not appear that vulnerable to classical algorithms and conventional computers in polynomial time and space. However, if the data exchanged needs to be kept secret for decades to come, then over such long time periods it is conceivable that technology might progress in currently unforeseen ways and new code breaking schemes might become possible. If so, intercepted encrypted communications that cannot be broken today could become vulnerable to attack due to new code-breaking algorithms or hardware. Given the lengths to which an eavesdropper must go to intercept the signal in the first place, it would be foolhardy to assume they will not place similar efforts in acquiring the computational resources and decryption methods needed to break the codes.

To illustrate how hard it can be to foresee how code-breaking technology might evolve, consider the Israeli TWINKLE device [448]. TWINKLE is an ingenious, special purpose, factoring machine. It is not a general purpose computer, but a hand-held optoelectronic device that is extraordinarily good at factoring integers. TWINKLE is the brainchild of the Weizmann Institute of Science in Israel. It can factor large integers using the NFS about 500 to 1000 times faster than was possible on the conventional computer hardware available at the time, namely, 100 MHz computers [448]. TWINKLE consists of an opaque blackened cylinder 6 inches in diameter

by 10 inches tall. At the base of the cylinder is an array of light emitting diodes (LEDs). The TWINKLE device assigns primes to LEDs (using space) and loops over the modular squares (using time). This makes the LEDs “twinkle” at different frequencies. At the top of the cylinder is a photodetector that measures the net illumination of the LED array. The photodetector sends an alarm to a PC to which it is connected whenever the total illumination exceeds a pre-set threshold. This defines a window on the possible factors of the integer being factored that is small enough to allow the Number Field Sieve to be used on the possible cases. TWINKLE could analyze 100,000,000 integers and determine in just 10 milliseconds which ones factor over the first 200,000 prime numbers. Moreover, the device could be replicated for about \$5000 a copy!

In 2003 TWINKLE was superceded by “TWIRL”, another hypothetical special purpose factoring engine out of the Weizmann Institute [449, 491]. TWIRL is purported to be able to factor 1024-bit numbers in about a year of effort, and one TWIRL device would cost a few tens of millions of dollars, an insignificant sum in certain quarters. As 1024-bit RSA keys are still used for securing sensitive communications, there is a concern that such keys may be insufficient to confer any real security.

Thus, factoring integers using massive computational resources, coupled with ingenious exploitation of analog physical methods, can allow codes of increasing key size to be broken in ways that were unanticipated when the code was created. This illustrates the potential vulnerability of public-key cryptosystems in the face of non-obvious special purpose hardware designed exclusively for code-breaking purposes. We need a much stronger cryptosystem whose security does not rely on technological, algorithmic, or computational resource assumptions.

13.2 An Unbreakable Cryptosystem: The One Time Pad

Classical cryptosystems that are much stronger than the public-key variety already exist. The “One Time Pad” (OTP), for example, invented by G.S. Vernam in 1917 in response to a task AT&T had set him to make a cryptosystem the Germans could not break, uses matching *private* keys, i.e., identical sequences of secret random numbers, to create an *unconditionally* secure classical cryptosystem. Provided the keys are truly random, known only to the legitimate parties wishing to communicate, and never re-used, then any messages encrypted using the OTP will be unintelligible to an adversary *forever* no matter how computationally powerful, mathematically gifted, or algorithmically advanced they might be! In fact, the one time pad cryptosystem is so secure it is rumored to be used for communicating diplomatic information between Washington and Moscow (see [528] p. 126).

How is such *unconditional* security possible? How can a cryptographic scheme be immune from all future advances in mathematics, computer science, and cryptology regardless of what they might be? That’s an astonishing warranty! The fundamental reason is that, if the OTP is used properly, there is no information from which the public key can be computed. Moreover, there is no value in even *guessing*

the key because the ciphertext can be mapped into any plaintext of the same length given the appropriate key. So guessing the key is pointless.

Despite its high level of security the “One Time Pad” (OTP) is surprisingly simple. The two parties wishing to exchange secret messages are traditionally called “Alice” and “Bob”. Before secure communication can take place, Alice and Bob must agree on the “alphabet” of symbols from which their messages will be composed. Typically, such an alphabet contains lower-case letters, upper-case letters, punctuation marks, numbers, and parentheses. But any symbols are allowed. The exact composition of the alphabet is unimportant. All that matters is that Alice and Bob agree to use exactly the same set of symbols. For example, the following 76 symbol alphabet is sufficient for simple communications:

$$\begin{aligned} & a \ b \ c \ d \ e \ f \ g \ h \ i \ j \ k \ l \ m \ n \ o \ p \ q \ r \ s \ t \ u \ v \ w \ x \ y \ z \\ \$Alphabet = & A \ B \ C \ D \ E \ F \ G \ H \ I \ J \ K \ L \ M \ N \ O \ P \ Q \ R \ S \ T \ U \ V \ W \ X \ Y \ Z \\ & 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 0 \ ! \ ? \ . \ , \ ; \ : \ (\) \ { \ } \ [\] \ \underline{\wedge} \ ' \end{aligned} \quad (13.4)$$

To be able to perform cryptographic operations on symbols Alice and Bob must also agree upon a common convention by which they associate a unique integer with each symbol. These integers should be padded with leading zeroes so that every integer uses the same number of digits, i.e., has the same “width”. For example, using the 76 symbol alphabet above Alice and Bob might make the associations:

$$\begin{aligned} & 00 = a \ 01 = b \ 02 = c \ 03 = d \ \dots \ 23 = x \ 24 = y \ 25 = z \\ \$Substitutions = & 26 = A \ 27 = B \ 28 = C \ 29 = D \ \dots \ 49 = X \ 50 = Y \ 51 = Z \\ & 52 = 1 \ 53 = 2 \ 54 = 3 \ 55 = 4 \ \dots \ 73 =] \ 74 = \underline{\wedge} \ 75 = ' \end{aligned} \quad (13.5)$$

Any plaintext message, written exclusively from the symbols in \$Alphabet, can then be mapped into an equivalent sequence of integers by substituting the appropriate integer from the \$ Substitutions for the successive symbols in the plaintext. Such a substitution code confers little security in itself. For example, it is easy to break a substitution code by matching the frequency distribution of the code integers to the frequency distribution of symbols in the language in which the plaintext is written. In the context of the OTP, the main purpose of the substitution code is to set the stage for a numerical procedure to be applied to those integers to conceal their true identity, and hence the true symbols, from a potential eavesdropper.

This numerical procedure combines the message integers with a set of private cryptographic “keys”. These are “private” in the sense they are known only to Alice and Bob, and they are “keys” in the sense that they can be used to lock (i.e., encrypt) and unlock (i.e., decrypt) a message. They are random integers picked uniformly in the range 0 to $\ell - 1$, where ℓ is the number of distinct symbols in the alphabet. Typically, these keys are printed in a booklet or “pad”, which is where the word “pad” in the name “one-time pad” originates. Thus, an admittedly unrealistically

small key pad containing 5 pages with 25 keys per page might be as follows:

$$\begin{aligned}
 & 37\ 69\ 40\ 19\ 17\ 65\ 34\ 26\ 62\ 32\ 29\ 57\ 31\ 27\ 56\ 53\ 36\ 15\ 52\ 72\ 07\ 48\ 48\ 19\ 41 \\
 & 05\ 75\ 70\ 18\ 56\ 15\ 15\ 09\ 44\ 41\ 00\ 72\ 26\ 31\ 20\ 37\ 36\ 23\ 41\ 19\ 38\ 63\ 01\ 68\ 18 \\
 \$Pad = & 30\ 57\ 26\ 33\ 36\ 75\ 52\ 16\ 01\ 70\ 48\ 14\ 42\ 23\ 15\ 20\ 28\ 45\ 34\ 51\ 55\ 37\ 06\ 08\ 66 \\
 & 32\ 73\ 68\ 22\ 00\ 70\ 57\ 00\ 09\ 24\ 42\ 26\ 32\ 45\ 46\ 47\ 14\ 35\ 10\ 59\ 35\ 24\ 62\ 66\ 13 \\
 & 54\ 36\ 71\ 01\ 28\ 23\ 26\ 39\ 04\ 67\ 23\ 33\ 07\ 09\ 38\ 37\ 10\ 32\ 05\ 64\ 73\ 63\ 32\ 20\ 68
 \end{aligned} \tag{13.6}$$

Alice and Bob are now ready to start exchanging secure messages. Here are the steps that must be taken for Alice to send a message to Bob that will be forever unintelligible to a potential eavesdropper no matter how mathematically sophisticated, computationally powerful, or algorithmically advanced they might be.

One-Time Pad Cryptosystem

1. Alice converts her plaintext message into a sequence of integers, $M = \text{MessageToIntegers}(\text{plaintext})$. Thus the original plaintext, consisting of N symbols, becomes the N message integers $M = \{m_1, m_2, \dots, m_N\}$.
2. Next Alice chooses a page, P , of keys from her copy of the key pad that matches the one in Bob's possession. This provides a supply of random integers $\{k_1, k_2, \dots\}$. Alice only needs to use the first N such keys to encrypt M .
3. To perform the encryption, Alice computes a sequence of N encrypted integers $E = \{e_1, e_2, \dots, e_N\}$ by applying the rule $e_i = m_i + k_i (\text{mod } \ell)$; that is, the i -th encrypted integer is obtained by adding the i -th message integer to the i -th key on page P of the key pad, dividing the result by ℓ and keeping the remainder.
4. Alice sends Bob (E, P) the encrypted message, E , and page number, P , of the keys that she used.
5. Upon receipt, Bob looks up the keys on page P from his copy of the key pad. He finds the keys that Alice used, namely, $\{k_1, k_2, \dots\}$. Using the encrypted message $E = \{e_1, e_2, \dots, e_N\}$, Bob reconstructs the message integers, $M = \{m_1, m_2, \dots, m_N\}$, by computing $m_i = e_i - k_i + \ell (\text{mod } \ell)$.
6. Finally, Bob converts M back into the original message using $\text{plaintext} = \text{IntegersToMessage}(M)$ operation, which is the inverse of the MessageToIntegers operation.

Alice and Bob agree on a common alphabet and a common encoding of integers for the symbols in it. They also are assumed to possess identical pads of true random numbers, which have never been used before in any prior encrypted communications. Alice encrypts her “plaintext” using her private key, sends the ciphertext to Bob, who then decrypts it using his matching private key.

$$\text{Alice encrypts: } \text{Encrypt}[\text{plaintext}, k_{\text{private}}] = \text{ciphertext} \tag{13.7}$$

$$\text{Bob decrypts: } \text{Decrypt}[\text{ciphertext}, k_{\text{private}}] = \text{plaintext} \tag{13.8}$$

Example of a OTP To see how the OTP works, consider the following example. Suppose Alice wants to send Bob a secret message composed of characters from \$

Alphabet, such as “My PIN number is 1234!”, using the key pad, \$Pad, mentioned above. We assume Alice and Bob have already agreed to exchange messages that were only composed of the symbols from \$ Alphabet and each has a copy of \$Pad.

Given Alice’s choice of which page to use from the key pad, the key pad itself, and the common alphabet, she can create a one-time pad encryption of a message. For example, if Alice uses page 3 of her key pad, the message “My PIN number is 1234!” becomes $E = \{68, 5, 24, 74, 70, 38, 50, 29, 21, 6, 49, 18, 59, 21, 23, 38, 26, 21, 11, 29, 34, 23\}$. Bob can decrypt Alice’s coded message using page 3 of his matching key pad to obtain $M = \text{My PIN number is 1234!}$

13.2.1 Security of OTP: Loopholes if Used Improperly

If used correctly, the One Time Pad is guaranteed to be secure regardless of the computational power of an adversary. Fundamentally, this is because there is no information from which an adversary can compute the key (as the key is truly random), and guessing the key is impossible because there exists keys that are capable of mapping any ciphertext into any plaintext of the same length! So an adversary cannot distinguish between the single “correct” key and the vastly greater number of “incorrect-but-grammatically-plausible” keys. Because of this, the One Time Pad is formally regarded as an *unconditionally secure* cryptosystem. However, in practice, when human or technological failings cause the actual use of the OTP to deviate from the ideal, there are several vulnerabilities.

First one needs keys that are *truly random*. It is not good enough to use the sequence of numbers arising from (say) a pseudo-random number generator of the sort found on most modern computers. Instead, one should use a naturally random process to obtain the random numbers, such as a *quantum* random number generator, or the intervals between radioactive decay events. If one uses a less than perfect random number generator there is an opportunity for an eavesdropper to recognize subtle structure amongst the (pseudo-)random numbers which can be exploited, e.g., by anticipating the continuation of the key sequence.

Secondly, one should never re-use the keys. The reason for this is that in a OTP there are always some random numbers (i.e., potential keys) that will convert any ciphertext of N characters into *any* plaintext of N characters. So merely *guessing* what the key material could be is of no help whatsoever. But if you use the same key material to encrypt two different plaintext messages into two different ciphertexts, then there are very few options for what the key material could be that would result in *both* ciphertexts being converted back to comprehensible language. One of the most notorious “failures” of the OTP cryptosystem, which led to the discovery of the Rosenberg spy ring, was due to the spies running out of and, consequently, re-using their key material. This gave the authorities the ability to determine the keys and unscramble incriminating message [441].

Thirdly, one needs to keep the random numbers in the key pads secure. If an adversary obtained the key pads it would make all OTP-encrypted communications based on those key pads readable.

So even though the OTP is, theoretically, an unconditionally secure cryptosystem, in actuality this depends on the OTP protocol being followed precisely.

13.2.2 Practicality of OTP: Problem of Key Distribution

A bigger obstacle in creating a global secure communications infrastructure based on the OTP lies not in the potential insecurities that would arise if the OTP were used improperly, but rather the impracticality of performing the required *key distribution*.

Before Alice and Bob can communicate securely using a OTP, they must possess matching private keys, i.e., key pads containing identical sets of true random numbers. Unfortunately, the OTP protocol consumes one key bit for each message bit that is encrypted, and therefore consumes key material at such a voracious rate that any finite key pad will be exhausted quickly. Moreover, there is no convenient classical method for replenishing the key material at the rate necessary to support modern communications. This makes it difficult, therefore, to create a viable OTP based cryptographic infrastructure seamlessly on a wide scale using conventional (i.e., purely classical) methods. Sharing more bit-intensive data such as images, speech recordings, and video files will only exacerbate the problem.

Today, using classical physics, the best Alice and Bob can do is to meet periodically face to face, create new matching key pads, and then return home keeping their new key material secret. But since they need an independent random number for each character in each message, they will have to exchange giga-bytes or tera-bytes of key material each time they meet to ensure they won't run out of key material before their next planned meeting.

An alternative is for Alice to create a set of true random numbers, copy them onto a set of CDs or hard drives, and employ a trusted intermediary to convey them to Bob. Here "trusted" is a bit of a euphemism: in reality, Alice would use a courier under armed guard, carrying a tamper resistant, explosive-laden briefcase loaded up with CD's or hard drives of true random numbers. This gives Alice high confidence, but still not absolute certainty, that her random numbers arrived in Bob's hands without being compromised. Moreover, certain applications, such as re-keying a satellite, pose even greater challenges. Hence, the necessity for such face to face encounters undermines the practical utility of the OTP greatly.

Moreover, the integrity of the entire cryptosystem rests on maintaining the secrecy of the key pads. Should one of the key pads ever fall into the wrong hands, the messages could be decrypted easily by an adversary.

These factors make one-time pads of limited utility. A more practical scheme needs a way to distribute the keys, in a secure fashion, without the sender and recipient having to meet face to face. Fortunately, quantum information provides such an alternative means of achieving perfectly secure key distribution under the nose of a potential adversary using a technique known as "quantum key distribution".

13.3 Quantum Key Distribution

Quantum key distribution (QKD) provides a means for two parties, traditionally called Alice and Bob, to establish matching, assuredly private, cryptographic keys across a potentially insecure communications channel. Once such keys have been established they can be used within a classical cryptosystem such as the unconditionally secure One Time Pad (OTP) or the still very strong, but not unconditionally secure, AES [16]. By using AES one is sacrificing absolute security in return for being able to support a much greater data rate on the encrypted channel.

13.3.1 Concept of QKD

There are many different ways in which a QKD scheme can be accomplished. Although they may differ in the quantum physical resource used to encode the key material, at their core they all work basically the same way. First, Alice generates a stream of truly random bits from which Alice and Bob are to distill a matching private cryptographic key. Neither Alice nor Bob have any particular key in mind at the outset. In fact, having the key emerge out of the QKD protocol affords a certain additional security, since the key identity is not written down anywhere in advance. Once the stream of random bits has been generated, they are encoded in the quantum states of a corresponding stream of photons. The encoding is chosen in such a manner as to guarantee that any attempt to measure the quantum-encoded key material in transit, without proper knowledge of the encoding used, will increase the bit error rate (BER) on the channel sufficient to expose the fact that eavesdropping had occurred. Nothing like this is possible in conventional cryptography because classical information can be read and re-transmitted in a manner imperceptible to the legitimate parties. However, in the quantum realm, the laws of quantum physics make it *impossible* to read the keys without scrambling enough of them to cause a detectable increase in the bit error rate (BER) on the channel, and hence alert the legitimate parties to the presence of an eavesdropper. If such a test reveals no evidence for eavesdropping, then the channel may be assumed to have been secure during the key distribution, and hence the random bits remaining after protocol has ended may be used as cryptographic keys. However, if eavesdropping was detected, the keys exchanged must be discarded and a fresh key distribution exchange attempted.

13.3.2 Security Foundations of QKD

Unlike classical public key cryptosystems whose security relies upon the difficulty of factoring integers or computing discrete logarithms, the security of

QKD rests upon quantum physical laws that cannot be circumvented no matter how mathematically gifted, algorithmically sophisticated, or computationally powerful an adversary might be. These laws include Heisenberg's Uncertainty Principle (see Sect. 12.1), which quantifies the minimum degree of disturbance an act of measurement on one observable must impart to a complementary observable, and the No-Cloning Theorem (see Sect. 11.6.2) which prevents an eavesdropper from making a perfect copy of an unknown quantum state. These quantum physics laws are a fundamental aspect of Nature, verified experimentally to an exceedingly high level of precision, and are impossible to circumvent. If they were, then all sorts of bizarre implications would ensue, such as the ability to communicate messages faster than the speed of light.

13.3.3 OTP Made Practical by QKD

Thus, QKD makes the OTP cryptosystem finally *practical* because the parties will no longer have to meet face-to-face periodically to refresh their key material. As the OTP is an *unconditionally secure* cryptosystem, it is much stronger than the public-key cryptosystems we use today for all our discreet communications, such as online banking transactions. Hence, QKD could potentially revolutionize confidential communications if it could be deployed on a global scale without compromising its underlying theoretical model. In fact, a global OTP-based cryptographic infrastructure, rekeyed with QKD, would render eavesdropping on encrypted communications quite useless. This fact has not gone unnoticed by parties disgruntled at the discoveries

13.3.4 Varieties of QKD

There are many different ways to make a QKD system. These differ in the quantum physical effects being exploited, the key-establishment protocols being used, and the physical laws being relied upon for security. These include:

1. Bennett and Brassard's "BB84" protocol based on two non-commuting observables [47].
2. Bennett's "B92" protocol based on two non-orthogonal states [46].
3. Bruss' "Six-state" protocol (which extends BB84) [86].
4. Ekert's "Entanglement-based" protocol [167].
5. Spedalieri's "Orbital Angular Momentum"-based protocol [476].
6. Protocols based on coherent states, e.g., [214, 215, 252, 298, 325, 339, 408, 459, 524].

Of these, the BB84 protocol, invented by Charles Bennett and Gilles Brassard in 1984, has a special place in the history of quantum cryptography since it was the first QKD scheme to be invented.

13.4 Physics Behind Quantum Key Distribution

13.4.1 Photon Polarization

In classical physics, light is regarded as consisting of electromagnetic waves that can travel through space. Traditionally, such waves are pictured, as shown in Fig. 13.1, as consisting of oscillating electric and magnetic fields that lie in planes perpendicular to each other and to the direction of wave propagation. Thus, in a three-dimensional coordinate system with mutually perpendicular x -, y -, and z -axes, if a photon is propagating in the positive z -direction, its electric and magnetic fields will oscillate in the (x, z) -plane and the (y, z) -plane, respectively. The “polarization” of such a wave is determined by how its electric field is varying in space and time. Several types of polarization are possible. If we imagine riding along with the light wave along the z -axis, at any instant one can ask what trajectory the electric field is tracing out in the (x, y) -plane (as this plane glides forward with the wave). If the electric field follows a well defined trajectory in the (x, y) -plane the light is said to be “polarized”, and the shape of this trajectory determines the type of polarization. In particular, if the electric field oscillates back and forth along a line, the light is *linearly* polarized. Conversely, if the electric field executes a circular motion, the light is *circularly* polarized. And if the electric field follows an elliptical trajectory it is *elliptically* polarized.

These superficially different types of polarizations turn out to be related to one another in the sense that, if we superimpose two linearly polarized waves, the result can be a linearly polarized, circularly polarized, or elliptically polarized wave. If the two linearly polarized waves being superposed are of equal amplitude and in-phase, the result will be a linearly polarized wave oscillating in an intermediate plane. If the two linearly polarized waves are of equal amplitude, perpendicular to one another, but $\pi/2$ out of phase, the result will be a circularly polarized wave. As such a wave propagates its polarization vector will rotate in a plane perpendicular to the direction of propagation. Similarly, if the two linearly polarized waves are of

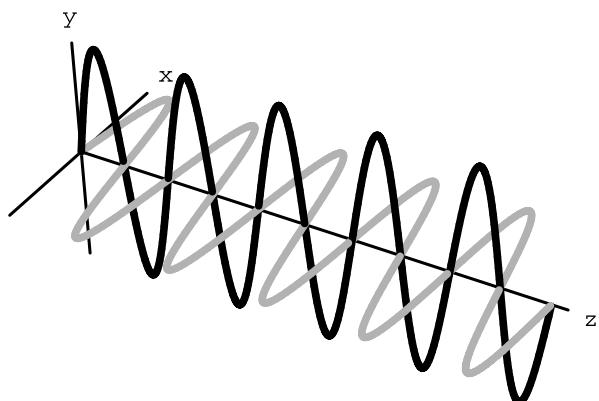


Fig. 13.1 A linearly polarized photon consists of an oscillating electric field and oscillating magnetic field that are perpendicular to each other and to the direction of propagation

different amplitudes, perpendicular to one another, and $\pi/2$ out of phase, the result will be an elliptically polarized wave.

Given the ability to obtain linear, circular, or elliptically polarized waves from superpositions of linearly polarized ones, it is not surprising that the converse is also true. Suitable combinations of a left circular polarization and a phase-shifted right circular polarization can give us a net linear, circular, or elliptically polarized light wave. This perspective helps us make the connection to quantum mechanics.

13.4.1.1 Quantum View of Polarized Photons

At the level of individual photons, we can no longer think of light as an electromagnetic wave. Instead we can only speak in terms of the quantum mechanical state of a photon. What we have been calling circular polarization at the classical level is closely related to the quantum mechanical “spin” at the quantum level.

According to quantum mechanics, particles can be broadly classified as bosons (having integer values of spin) or fermions (having half integer values of spin). Photons are bosons and have an intrinsic spin of 1, but this can be pointing in the direction of propagation or opposite to it, and so their spin quantum numbers can only assume one of two values: $+1\hbar$ or $-1\hbar$. These correspond, crudely, to circularly polarized states, $|\circlearrowleft\rangle$ or $|\circlearrowright\rangle$, having either a left-handed circular polarization or a right-handed circular polarization. These spin eigenstates rotate: as the photon propagates its polarization vector rotates in a plane that is perpendicular to the direction of motion. Photons therefore also possess angular momentum, just like rotating masses. A photon has no spin eigenstate for spin 0 (analogous to linear polarization), but this can be created by superposing the allowed spin eigenstates, corresponding to spin $-1\hbar$ and spin $+1\hbar$, with a suitable phase shift between them. Again, this is analogous to creating linearly polarized light from circularly polarized light. As the spin of a linearly polarized photon is spin 0 it carries no net angular momentum.

Thus, the most general state of a photon is a superposition of its two allowed states of spin polarization, i.e., $|\psi\rangle = a_L|\circlearrowleft\rangle + a_R|\circlearrowright\rangle$. In particular, the standard polarizations can be obtained as follows:

- Left circular polarization, $|\circlearrowleft\rangle$
- Right circular polarization, $|\circlearrowright\rangle$
- Horizontal polarization, $|\leftrightarrow\rangle = \frac{i}{\sqrt{2}}(|\circlearrowleft\rangle - |\circlearrowright\rangle)$
- Vertical polarization, $|\updownarrow\rangle = \frac{1}{\sqrt{2}}(|\circlearrowleft\rangle + |\circlearrowright\rangle)$
- Diagonal $+45^\circ$ polarization, $|\nearrow\rangle = \frac{1+i}{2}|\circlearrowleft\rangle + \frac{1-i}{2}|\circlearrowright\rangle$
- Diagonal -45° polarization, $|\nwarrow\rangle = \frac{1-i}{2}|\circlearrowleft\rangle + \frac{1+i}{2}|\circlearrowright\rangle$

13.4.2 Single Photon Sources

As our goal is to generate secure raw key material then, when we generate a random bit, not only do we want it to be truly random, but we also require it to be *one* bit—

not two or three or zero bits etc. If a purportedly “single” photon source actually emits multiple photons, this opens a potential security hole in the QKD protocol to follow, via a so-called “photon-splitting attack”. To minimize this risk, the early pioneers of QKD used strongly attenuated laser beams that contained, on average, 0.1 photons per pulse. In such systems, the probability of there being n photons is governed by Poisson statistics [201]:

$$P(n, \mu) = \frac{\mu^n}{n!} e^{-\mu} \quad (13.9)$$

Hence, the probability that a non-empty weak pulse contains more than one photon is:

$$P(n > 1 | n > 0, \mu) = \frac{1 - P(0, \mu) - P(1, \mu)}{1 - P(0, \mu)} = \frac{1 - e^{-\mu}(1 + \mu)}{1 - e^{-\mu}} \approx \frac{\mu}{2} \quad (13.10)$$

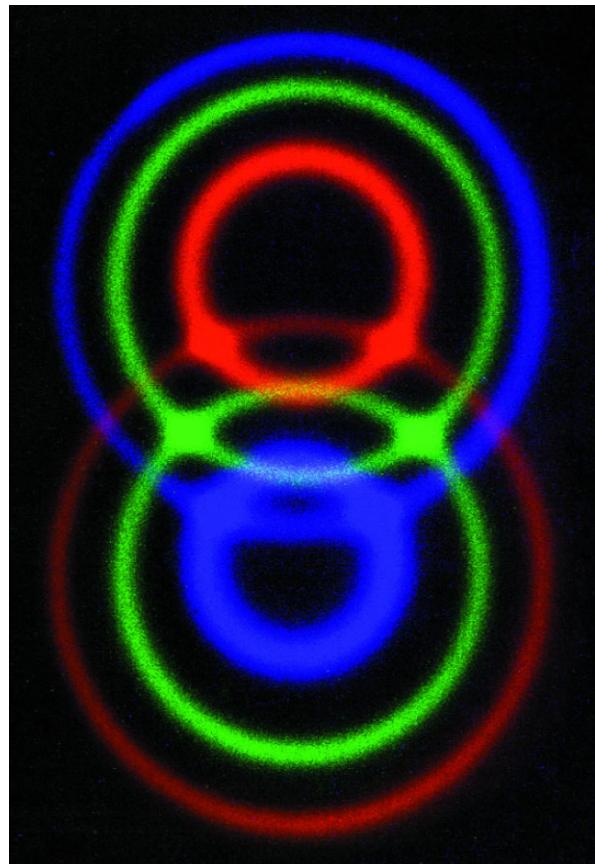
If μ is sufficiently small, i.e., if the beam intensity is sufficiently low, double or triple photon events are very rare, limiting the potential information an eavesdropper might glean. Unfortunately, by reducing the intensity so much, most pulses contain zero photons and so the key generation rate suffers. Although this can be offset by driver the rate at which the laser is pulsed, the detectors currently available cannot keep up. The good news is that with the appropriate setting for μ and after sufficient error reconciliation and privacy amplification the weak-laser scheme can still be made provably secure, albeit at a disappointingly low key distribution rate.

A better solution is to have a true single photon source, i.e. a device that spits out exactly one photon when commanded to do so [392]. Over recent years much progress has been made on such devices. They are generally based on either “single photon guns” that emit a single photon when stimulated and are typically based on intrinsic single-emitters such as molecules, quantum dots or diamond color centers [173, 185, 381, 430], or “heralded” sources that use the detection of one photon from an entangled pair of photons produced in parametric downconversion to signal the presence of the other photon in a different path [179, 203, 352, 393]. If one gangs several such devices together, than in a certain time interval one can be almost sure of creating at least one heralded single photon, which can be stored in a fiber loop and released at a pre-determined emission time. Such a source is not exactly on “on-demand” but is at least available at predictable times, which is almost as good.

13.4.3 Entangled Photon Sources

A key step in Ekert’s protocol is the generation of pairs of maximally entangled particles. This can be accomplished by sending coherent ultra-violet light into a special type of crystal called “beta barium borate” (or BBO). In such a crystal, about one photon in 10 billion undergoes a remarkable transformation: it interacts with the material so as to convert a single UV photon (at 351 nm) into a pair of longer wavelength photons (at 702 nm) with one of the photons horizontally polarized and the

Fig. 13.2 Entangled photons can be created by a process known as “parametric down conversion”. Ultraviolet light (351 nm) is shone into a crystal of “beta barium borate” (BBO). About 1 in 10 billion of the photons is down-converted to two photons of wavelength 702 nm that emerge in opposing places on *two green cones*. The *green photons* on one cone are vertically polarized and those on the other are horizontally polarized. Where the *green cones* overlap it is impossible to know from which cone any particular photon came, and the pairs for photons from the two intersection regions are therefore polarization entangled. The *blue and red photons* emitted are not entangled. (Copyright the Institute for Quantum Optics and Quantum Information, Austrian Academy of Sciences, and printed with their permission)



other vertically polarized. Moreover, these photons leave the BBO crystal in two overlapping and intersecting (green) cones. In the region where the two green cones overlap it is impossible to tell from which cone a particular photon came. Hence, in the region where the two green cones intersect there is entanglement in polarization states of the form $\frac{1}{\sqrt{2}}(|\downarrow\rangle|\leftrightarrow\rangle + |\leftrightarrow\rangle|\downarrow\rangle)$. This becomes the raw source of entanglement that can be used in a polarization-implementation of Ekert’s entanglement-based QKD protocol. Figure 13.2 shows a cross section through the cones of light leaving a BBO crystal. This has become an iconic image for the quantum information sciences.

Some better more modern entangled photon sources are [10, 13, 176].

13.4.4 Creating Truly Random Bits

The BB84 protocol begins with Alice choosing a truly random sequence of bits. These bits are the raw material from which a cryptographic key will be distilled.



Fig. 13.3 idQuantique’s quantum random number generator [405] is available as a PCI board, a USB device, or as a component ready for mounting on a printed circuit board. It has been tested and certified by the Swiss Federal Office of Metrology. Images courtesy of idQuantique

One way to generate truly random bits is to repeatedly shoot single photons into a perfect 50:50 beam splitter and see from which port each photon exits. By adopting a convention such as “upper port” = 0 and “straight-through” = 1, we can obtain a random sequence of bits. In reality, there might be complications to this simple minded scenario: the beam-splitter might not be *exactly* 50:50 but harbor a small bias, e.g., 49.9:50.1, in favor of one port over the other. Similarly, the photo-detectors used on each port might not have *exactly* the same detection efficiency due to slight variations in materials. However, there are ways to beat such systematic biases. For example, suppose you had a biased coin that came up heads with probability 49.9% and tails with probability 50.1%. You can still make a perfectly balanced 50:50 random number generator out of such a coin. Instead of naively taking “heads” = 0 and “tails” = 1, you simply use two coin tosses per bit, such as, “heads on the first toss and tails on the second toss” = 0, and “tails on the first toss and heads on the second toss” = 1, and throw out any trials that were both heads or both tails. Thus by sacrificing half the coin tosses, you can generate a truly random bit even though the coin is biased. Quantum random number generators (QRNG) are now available commercially [405], and these should be used in preference to a classical pseudo-random number generator to generate the set of random bits from which the cryptographic key is to be derived. As shown in Fig. 13.3 such QRNGs are available pre-packaged in form factors that enable seamless integration into other products.

13.4.5 Encoding Keys in Polarized Photons

We can use the polarization state of a photon to encode a bit. For example, if we were to use linearly polarized light we might agree on the convention that a vertically polarized photon, $|\downarrow\rangle$, corresponds to 0, and a horizontally polarized photon, $|\leftrightarrow\rangle$, corresponds to 1. Or we might have agreed that a $+45^\circ$ polarized photon, $|\diagup\rangle$, is a 0 and a -45° polarized photon, $|\diagdown\rangle$, is a 1. Hence, given a sequence of bits, e.g., 010100010101101..., a choice of encoding basis, e.g., $\{| \downarrow \rangle, | \leftrightarrow \rangle\}$ or $\{| \diagup \rangle, | \diagdown \rangle\}$, to encode each bit we need to be able to create a photon whose electric field is oscillating in a desired plane.

One way to do this is simply to pass the photon through a polarizer whose polarization axis is set at the desired angle. If the photon makes it through its polarization will have been “set” in the plane corresponding to the alignment of the polarizer.

13.4.5.1 Polarizers

The development of modern polarizers began with Edwin Land, the founder of Polaroid Corporation, in 1928, when he was an undergraduate at Harvard College. His interest in polarization was piqued when he read about the strange properties of crystals, called herapathite in honor of their discoverer Dr. William Herapath, that were formed when iodine was dropped into the urine of a dog that had been fed quinine. Whereas Herapath had struggled to make a single large herapathite crystal, Land realized he could assemble something almost as good by mechanically placing several small herapathite crystals in the proper alignment embedded in an extruded polymer, and thus the first large scale polarizers were born [264].

According to physics, one of two things can happen to a single photon passing through a polarizer: either it will emerge with its electric field oscillating in a plane aligned with the axis of the polarizer, or else it will be absorbed and its energy re-emitted later in the form of heat.

If the axis of the polarizer makes an angle of θ with the plane of the electric field of a photon fed into it, there is a probability of $\cos^2(\theta)$ that the photon will emerge with its polarization set at the desired angle (i.e., aligned with the polarization axis of polarizer) and a probability of $1 - \cos^2(\theta)$ that it will not emerge at all.

Therefore, a cascade of two polarizers with their polarization axes set at 90° with respect to one another will not pass any light. This is consistent with quantum theory because either the photon is absorbed at the first polarizer, in which case it is lost, or it makes it through the first polarizer but in doing so will have had its polarization state set to be orthogonal to that of the second polarizer (since the first and second polarizers are orthogonal). Either way the photon will not pass both polarizers. You can verify this experimentally by looking at some reflected light (which is often polarized) through two pairs of polarizing sunglasses that are tilted with respect to one another. As one of the polarizers is rotated while keeping the other one fixed, the scene will go brighter and darker, extinguishing the transmitted light when the two polarizers are crossed at 90° . The downside of using polarizers to set the polarization plane of the photon is that we are not guaranteed that the photon will make it through. A better method would simply imprint the desired polarization on the photon deterministically. This is possible using a Pockels cell.

13.4.5.2 Pockels Cells

The Pockels cell was invented in 1893 by German physicist Friedrich Pockels and it basically acts as a birefringent switch. The birefringence is induced by an externally applied electric field. By using a Pockels cell, it is possible to create a photon with

its electric field oscillating in any desired plane. We can therefore (arbitrarily) call polarized photons whose electric fields oscillate in a plane at either 0° or 90° to some reference line “rectilinear” and those whose electric fields oscillate in a plane at 45° or 135° “diagonal”. Furthermore, we can stipulate that photons polarized at angles of 0° and 45° are to represent the binary value 0 and those polarized at angles of 90° and 135° represent the binary value 1. Once this correspondence has been made, a sequence of bits can be used to control the bias in a Pockels cell and hence determine the polarization orientations from the stream of photons emerging from the cell. This allows a sequence of bits to be converted into a sequence of polarized photons. These may then be fed into some communication channel, such as an optical fiber or perhaps even transmitted through free space.

13.4.6 Measuring Photon Polarization with a Birefringent Crystal

In order to recover the bits encoded in the polarization orientation of a stream of photons, it is necessary for the recipient to measure the polarizations. Fortunately, Nature has provided us with a material, “calcite”, beautifully suited for just this purpose.

A calcite, or calcium carbonate (CaCO_3), crystal has the property of *birefringence*. This means that the electrons in the crystal are not bound with equal strength in each direction. Consequently, a photon passing through the crystal will feel a different electromagnetic force depending on the orientation of its electric field relative to the polarization axis in the crystal. For example, suppose the calcite’s polarization axis is aligned so that vertically polarized photons pass straight through it. A photon with a horizontal polarization will also pass through the crystal but it will emerge from the crystal shifted from its original trajectory as shown in Fig. 13.4.

If the calcite crystal is oriented such that $|\downarrow\rangle$ polarized photons pass straight through and $|\leftrightarrow\rangle$ polarized photons emerge lower down, then what happens to a



Fig. 13.4 How a birefringent crystal can be used to separate photons based on their polarization. In the figure shown, a naturally cleaved and unpolished calcite crystal is placed on top of the text “polarization dependent refraction” and illuminated with sunlight containing a mixtures of polarizations. As a result reflected ordinary rays are refracted through the crystal via one path and reflected extraordinary rays are refracted by a slightly offset path. This causes the blurring of the letters that are visible through the crystal. We can therefore use where the light rays emerge from such a crystal to measure their polarization state

photon whose polarization is something intermediate such as $|\diagdown\rangle$? In this case, the calcite crystal acts as if it is measuring the polarization of the photon in the $\{|\uparrow\rangle, |\downarrow\rangle, |\leftrightarrow\rangle\}$ -basis. That is, we need to interpret the incident photon in state $|\diagdown\rangle$ as really being in a superposition state $|\diagdown\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle)$. Hence, there will be some probability the photon will pass straight through the crystal and be left in state $|\uparrow\rangle$, and there is some probability the photon will pass through the crystal, emerge lower down, and be left in $|\downarrow\rangle$. A priori we cannot tell which will be the case. However, when using a calcite crystal to classify the polarization it is always going to exit one way or another and never be blocked, but the result we obtain is not necessarily the polarization of the incident photon, but rather that of the exiting photon. Thus we can use the location from which a photon emerges from a calcite crystal as a way of signaling (i.e., measuring) that photon's polarization in a particular basis. Depending on what the incident photon polarization was, in that basis, will depend on whether the measurement outcome is deterministic or non-deterministic.

13.4.7 Measuring Photon Polarization with a Polarizing Filter

Whereas a calcite crystal always returns a decision regarding the polarization state of each incoming photon, the polarizing filter is only guaranteed to return a result when the polarization orientation of the filter happens to be aligned with the polarization of the photon. At other angles there is some diminishing probability that the photon will make it through the filter, and in fact there is zero chance that the photon will pass through the filter when the photons' polarization is orthogonal to the polarization axis of the filter.

You can verify the behavior of polarizers for yourself quite easily by looking at a light source through two polarizing filters (such as two pairs of polarizing sunglasses). Even if initially the source is unpolarized, it will be polarized after passing through the first filter. Then, as it encounters the second polarizer the chances of it making it through will depend upon the alignment between the (now polarized photons) and the polarization axis of the second polarizer. By looking at the light source through both polarizers, while rotating the second polarizer relative to the first, you can make the source appear brighter or dimmer, and you can distinguish the output light completely when the two polarizers are “crossed”, i.e., when the polarization axis of the first polarizer is orthogonal to that of the second. This behavior is illustrated in Fig. 13.5.

13.5 Bennett and Brassard's BB84 QKD Scheme

The genesis of the BB84 QKD protocol [47] lies in a paper entitled “Conjugate Coding” written by Stephen Wiesner in the 1970’s but which went unpublished until 1983 [531]. Wiesner introduced the concept of a banknote that was impossible to

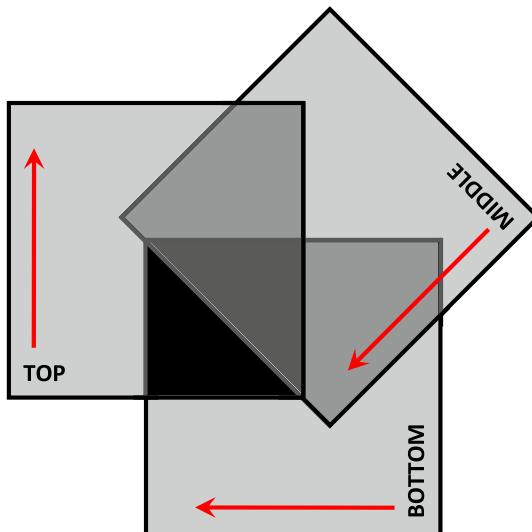


Fig. 13.5 Polarizing filters are only guaranteed to transmit a photon when the polarization of the photon and polarization axis of filter are aligned. At other relative angles there is only some probability that the photon will be transmitted. If the polarization of the photon is orthogonal to that of the filter the photon's passage through the filter is guaranteed to be blocked, and no light will emerge. Hence, the polarizing filter either reports a definite polarization result or no result whatsoever. Contrast this with a calcite crystal that always reports some result regardless of the polarization of the input photon

counterfeit. Each banknote was associated with a classical serial number (such as a classical bit string) together with a quantum serial number (that encoded a bit string in a sequence of four quantum states— $\{|\uparrow\rangle, |\leftrightarrow\rangle\}$ and $\{|\nwarrow\rangle, |\nearrow\rangle\}$)—corresponding to the basis states of two orthogonal bases that were rotated 45° with respect to one another). The bank maintained a record of each classical serial number and its corresponding quantum serial number. When such a banknote was presented to the issuing bank, it could verify the legitimacy of the banknote by looking up the basis-encoding that was supposed to accompany the classical serial number, and then measure the quantum serial number in that basis. If the result agreed with the what the quantum serial number ought to be, the banknote was legitimate and the money could be spent. It is likely that the 1970's reviewers dismissed the scheme as too fanciful since the idea of storing quantum serial numbers for long durations on a banknote would have appeared rather far fetched at that time. Today it would still be a challenge beyond our reach, but is no longer beyond the realm of possibility. It does make one wonder whether there might not be a better approach to scientific reviewing in the internet age. If an idea is too far ahead of its time, it could easily be rejected.

13.5.1 The BB84 QKD Protocol

Like Wiesner's quantum money, the BB84 protocol employs polarization states of single photons selected from one of two orthogonal polarization bases: $\{| \uparrow \rangle, | \leftrightarrow \rangle\}$ and $\{| \nwarrow \rangle, | \nearrow \rangle\}$. These bases correspond, respectively, to photons that are either vertically ($| \uparrow \rangle$) or horizontally ($| \leftrightarrow \rangle$) polarized, and photons that are either diagonally ($| \nwarrow \rangle$), or anti-diagonally ($| \nearrow \rangle$) polarized. These bases are related in that $| \nwarrow \rangle = \frac{1}{\sqrt{2}}(| \uparrow \rangle + | \leftrightarrow \rangle)$ (say) and $| \nearrow \rangle = \frac{1}{\sqrt{2}}(| \uparrow \rangle - | \leftrightarrow \rangle)$, i.e., the polarization orientations of the two bases are tilted at 45° with respect to one another in *real* space. Note that, on the Bloch sphere, the geometry is different: $| \uparrow \rangle$ and $| \leftrightarrow \rangle$ correspond to quantum states at the North and South poles of the Bloch sphere respectively. Likewise, $| \nwarrow \rangle$ and $| \nearrow \rangle$ correspond to East-West antipodal points that lie in the equatorial plane. The relative locations of these states on the Bloch sphere are shown in Fig. 13.6. If you are confused by this, recall that antipodal states on the Bloch sphere correspond to orthogonal quantum states, review the material in Sect. 1.3.3.

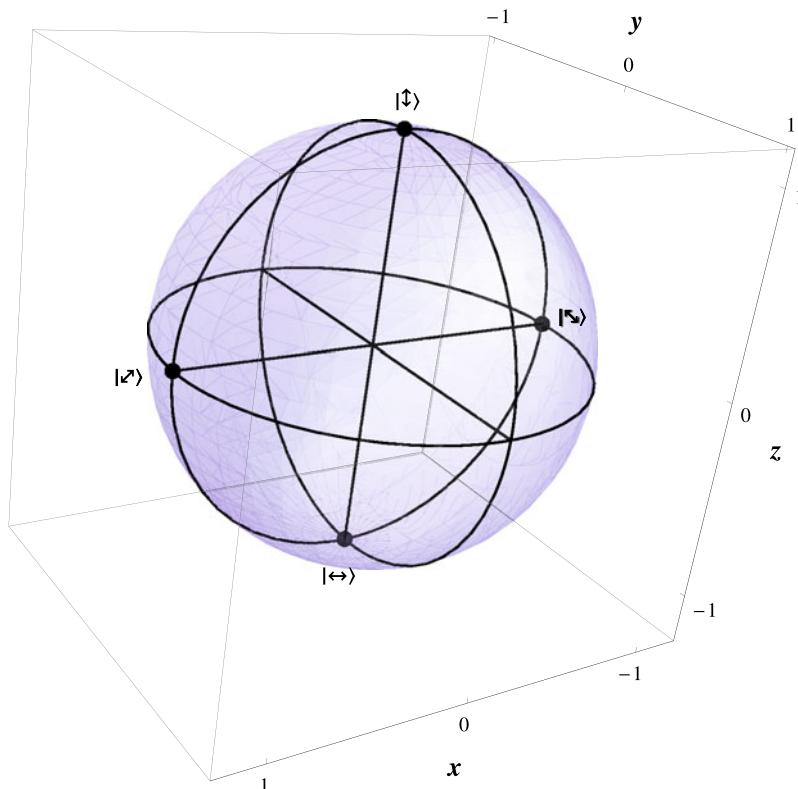


Fig. 13.6 The four states used in the BB84 in terms of their locations on Bloch sphere. Note that all four states lie in the same plane and that states having orthogonal polarizations lie at antipodal points on the Bloch sphere

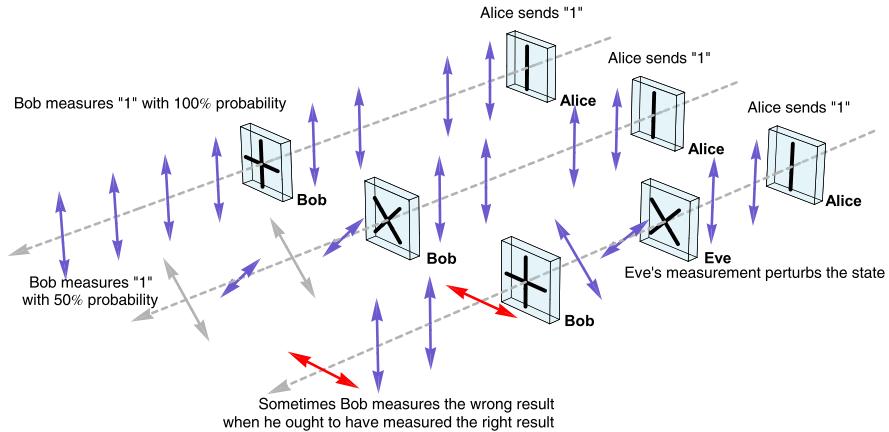


Fig. 13.7 BB84 protocol TEST ONLY. This figure is copyright protected and cannot be used. A new figure conveying similar information needs to be drawn here. Illustration of the BB84 QKD protocol

The BB84 quantum key distribution protocol is summarized in Fig. 13.7 and works as follows:

BB84 Quantum Key Distribution Protocol

1. *Alice Generates Key Material:* Alice uses a *true* random number generator to create a long string of random bits. These are the raw bits from which Alice and Bob must distill a matching private key. Their job is to determine a subset of these bits that they, and only they, will know in common. This privileged subset of bits becomes their private cryptographic key.
2. *Alice Encodes Key Material in Polarized Photons:* Alice sends each of her random bits to Bob, one after another, encoded in the polarization state of single photons. Alice and Bob agree on the following encoding strategy: If Alice wants to send Bob a 0 she transmits either $|\uparrow\rangle$ or $|\downarrow\rangle$ with equal probability, and if she wants to send a 1 she transmits either a $|\leftrightarrow\rangle$ or $|\swarrow\searrow\rangle$ with equal probability.
3. *Bob Measures the Polarization of Each Incoming Photon Using a Birefringent Crystal:* Upon receiving each photon, Bob measures its polarization state. To do so, conceptually, Bob picks a polarization basis ($\{|\uparrow\rangle, |\leftrightarrow\rangle\}$ or $\{|\downarrow\rangle, |\swarrow\searrow\rangle\}$) in which to make his measurement, orients a birefringent crystal (such as calcite) accordingly, and observes from which port the photon exits the crystal. If Bob chooses the polarization basis for his measurement that happens to be the same as the one Alice used to encode the bit, then Bob will determine the correct polarization state of the photon, and hence (knowing Alice's encoding strategy) be able to tell whether she sent a 0 or a 1. However, if Bob chooses the wrong polarization basis for his measurement (i.e., the basis other than the one Alice had used to encode the bit), then Bob will only have a 50:50 chance of determining the correct bit value that Alice. Moreover, at this point Bob has no idea which of

- the bit values he measured are correct and which ones are incorrect with respect to the bit values Alice sent.
4. *Bob Discloses the Basis He Used For Each Polarization Measurement:* To determine which bits are correct, after Bob has completed all his measurements, he tells Alice the *basis* in which he measured each incoming photon, but not the value he observed for the measurement. Alice then tells Bob those occasions on which they used the *same* basis for Alice's encoding and Bob's measurement. This information allows Bob to determine the subset of bits he measured that are supposed to match the bits Alice sent. Hence, Alice and Bob can now discard all those cases in which they used different bases. Thus, in the absence of any noise, imperfections, and eavesdropping, Alice and Bob should come to possess matching sequences of randomly generated bits. These can serve as a fresh supply of cryptographic keys.
 5. *Alice and Bob Test for the Presence of an Eavesdropper by Revealing a Sub-set of the Bit Values they Share:* Unfortunately, Alice and Bob don't yet know whether an eavesdropper, Eve say, might have been eavesdropping on their communications channel. To test for the presence of an eavesdropper, Alice and Bob agree to sacrifice a portion of their matching bits and disclose (publicly) their bit values. If an eavesdropper, Eve, was present she would have to pick a basis in which to measure each bit en route from Alice to Bob. Like Bob, she would only guess this correctly, on average, half the time. When Eve guesses the basis incorrectly (an event that occurs with probability $\frac{1}{2}$), she will necessarily perturb the state of the photon incoming from Alice and will therefore pass on a perturbed state when she re-transmits it to Bob. On those occasions, Bob would receive an incoming photon that was not in the state it ought to have been, but instead be in a superposition of the two states of his measurement basis. Consequently, Bob will measure the "wrong" polarization with probability $\frac{1}{2}$. Thus, for each of the bits Alice and Bob intend to use as key material, there is a probability of $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$ that Eve will be detected, and a probability of $\frac{3}{4}$ she will remain undetected. Hence, if Alice and Bob sacrifice N of the bits from their (supposedly matching) key material, and disclose these N bit values to each other publicly, the probability that Eve will evade detection in all these trials is $(\frac{3}{4})^N$, and so she will be detected with probability $1 - (\frac{3}{4})^N$. By examining only a small fraction of their (supposedly matching) key material for errors, Alice and Bob can thereby determine whether or not an eavesdropper was present.

All the stages of the BB84 protocol are implementable using standard quantum optics laboratory equipment. The most interesting parts of an implementation of BB84 concern the generation or the raw key material, its encoding in polarized photons, and the measurement of the polarization state of each incoming photon to Bob.

13.5.2 Example: BB84 QKD in the Absence of Eavesdropping

Alice and Bob need to choose the probability with which they want to be able to detect eavesdropping and the number of bits they want to use in their key. These parameters determine how many photons they must exchange in order to get a key of the size they require. Suppose they would like to have a 75% chance of detecting any eavesdropping and would like to create a secure key based on 4 bits. These numbers are unrealistically low for a real cryptographic key but allow us to illustrate the principle behind quantum key distribution.

Figure 13.8 is a diagram illustrating the sequence of steps Alice makes to encode bits as polarized photons.

Alice chooses a set of random bits (first row in Fig. 13.8). Then, for each bit, she chooses to encode it in either the rectilinear polarization (+) or in the diagonal (\times) polarization of a photon (second row). This choice of polarization bases must be made randomly. Alice then sends the photons she created to Bob over an open communications channel (third row of Fig. 13.8).

Next consider the actions Bob takes upon receiving the photons; these actions are shown in Fig. 13.9. Upon receipt of the photons (first row of Fig. 13.9), Bob chooses an orientation for his calcite crystal (second row) with which he measures the direction of polarization of the incoming photons. Hence Bob reconstructs a set of bits (third row).

Now Alice and Bob enter into a public (insecure) communication in which Alice divulges to Bob the types of polarizers that she used to encode a subset of the bits. Likewise Bob divulges to Alice the types of polarizers he used to decode the *same* subset of bits, as shown in Fig. 13.10. For those cases in which the orientation of Alice's polarizer (i.e., polarization creator) was the same as that of Bob's calcite crystal (polarization measuring device), Alice tells Bob what bit values he ought to have measured. Assuming that the encoding, decoding, and transmission steps are error free, and provided there is no eavesdropping, Bob's test bits ought to agree with Alice's test bits 100%.

The more bits that are tested, the more likely it is that a potential eavesdropper is detected. In fact, for each bit tested by Alice and Bob, the probability of that

1	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	1	1	
\times	$+$	\times	\times	\times	\times	$+$	\times	\times	$+$	$+$	$+$	$+$	\times	\times	$+$	$+$	\times	$+$	$+$	\times	\times	$+$	\times	\times	\times			
\backslash	-	\backslash	\backslash	\backslash	/	/	-	/	\backslash	1	1	1	1	\backslash	/	1	1	/	-	1	/	/	1	1	/	\backslash	1	\backslash

Fig. 13.8 Alice encodes a sequence of random bits in the polarization orientations of a corresponding sequence of photons

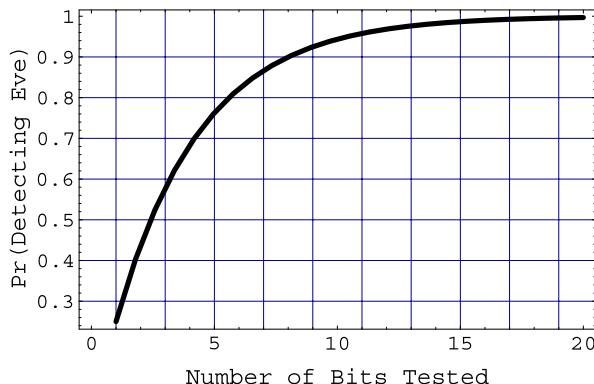
\backslash	-	\backslash	\backslash	\backslash	/	/	-	/	\backslash	1	1	1	1	\backslash	/	1	1	/	-	1	/	/	1	1	/	\backslash	1	\backslash
$+$	$+$	\times	$+$	\times	\times	$+$	\times	$+$	\times	$+$	$+$	$+$	\times	\times	$+$	\times	$+$	\times	$+$	\times	$+$	$+$	\times	$+$	\times	$+$		
0	1	1	1	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	1	1	1	0	1	1	1	0	1

Fig. 13.9 Bob decodes polarized photons as bits

	1	1	0		1	0		0	0	1			1	0	
+	x		x		x	+		x	+	+			x	+	
+	x		x		x	+		x	+	+			x	+	
1	1	0		1	0		0	0	1			1	0		

Fig. 13.10 Alice and Bob compare a subset of their bits to test for the presence of eavesdropping

Fig. 13.11 Probability of detecting eavesdropping as a function of the number of bits tested by Alice and Bob. For a bit to be testable, Alice and Bob must have used the same polarizer orientation to encode and decode that bit, respectively



x	x	x	x	+x	+	++x	+	x	+x	x	+x	++x	x	x	
+	+	x	+x	+	+	x	x	+	x	x	+x	++x	x	+	
0	1	1	0	0	1	0	0	0	1	1	1	0	1	1	1
☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺
1			0			0		0		0		0		1	

Fig. 13.12 Key exchange step

test revealing the presence of an eavesdropper (given that an eavesdropper is indeed present) is $\frac{1}{4}$, i.e., one chance in four. Thus, if N bits are tested, the probability of detecting an eavesdropper (given that one is present) is $1 - (\frac{3}{4})^N$. Figure 13.11 shows how this probability of detecting eavesdropping grows with the number of bits tested. As you can see, the probability of detecting eavesdropping approaches 1 asymptotically as the number of bits tested tends to infinity. Thus we can make the probability of detecting eavesdropping as close to certainty as we please simply by testing more bits. After Alice and Bob have decided that the channel is secure, Alice then tells Bob what polarization orientations she used for each of her remaining bits *but not what those bits were* (first row of Fig. 13.12). Next, Bob compares his calcite orientations with those of Alice's polarizer orientations (second row) and also records his own answers (third row). Bob then, categorizes each bit in terms of whether he used the same orientation as Alice (fourth row). Then he projects out just those cases where the same orientations were used (fifth row).

This sequence of actions allows Bob to deduce a set of bits known only to Alice and himself. To see this, compare the top line of Fig. 13.8 with the bottom line of

Fig. 13.12. You will find that Alice and Bob agree on the bits for those cases in which they used the same orientations for their polarizer and calcite crystal, respectively.

Having deduced a common sequence of bits, Alice and Bob can use this sequence as the basis for a key in a provably secure classical cryptosystem such as a one time pad.

13.5.3 Example: BB84 QKD in the Presence of Eavesdropping

Now consider what would have happened instead if there *had* been an eavesdropper, “Eve”, present. Now although we know that eavesdropping is taking place, Alice and Bob do not, so the first step proceeds as before with Alice encoding her bits in polarized photons, as in Fig. 13.13. This time, however, there is an eavesdropper, Eve, who is intercepting Alice’s photons and making her own measurements of their polarizations in an effort to see what bits Alice is sending to Bob. Eve goes through the operations that Bob would have performed: She intercepts the photons (first row), picks calcite orientations (second row), and decodes the polarized photons as bits (Fig. 13.14). In an effort to cover her tracks Eve then re-transmits the photons she measured to Bob. Eve is free to do a complete recoding of her measured bits into photons polarized in whatever orientations she chooses. But the simplest situation has Eve using the same sequence of orientations that she used during her decoding step.

At this moment he is unaware of Eve’s presence, so he proceeds to decode the photons he thinks are coming from Alice, but which are actually coming from Eve (see Fig. 13.15). Bob intercepts the photons (first row), picks calcite orientations (second row), and decodes the photons as a sequence of bits.

Now Alice divulges to Bob her polarizer orientations and the actual bit values sent, for a subset of the bits. Of these, Bob throws out any for which his calcite crystal had a different orientation. On those cases where they agree on orientation of Alice’s polarizer and Bob’s calcite crystal they should also agree on the bit sent and received. In Fig. 13.16 there is an error in the third bit tested that reveals the

1	1	1	1	1	1	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1		
x	+	x	x	x	x	x	+	x	x	+	+	+	+	+	x	x	+	+	x	+	+	x	x	+	x	+	x
\	-	\	\	\	/	-	/	\	/						\	/			/	-		/	/		/	\	\

Fig. 13.13 Alice encodes her bits as polarized photons

\	-	\	\	\	/	/	-	/	\						\	/			/	-		/	/		/	\	\	
+	+	x	+	x	+	+	x	x	+	+	+	+	+	x	+	+	x	+	x	x	x	+	x	x	+	x	+	x
0	1	1	1	1	0	0	1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	1	0	0	1	0	1	0

Fig. 13.14 Eve intercepts the photons Alice sent to Bob and tries to decode them. Eve then sends the photons she decoded on to Bob using whatever polarizer orientations Eve had picked

	-	\	-	\			\	/			\	/	-	\	/	-	\	/		\		\		
+	+	x	+	x	x	+	x	+	x	+	x	x	+	x	x	+	x	+	x	+	x	+	x	
0	1	1	1	1	0	1	1	0	0	1	0	0	0	0	1	1	1	1	1	0	0	0	1	0

Fig. 13.15 Bob decodes the photons unaware of Eve's presence

x	+	x		x	+	x			+	+	x		+	+	x		+	x	x	+	x	x	x	
+	+	+		+	x	+			x	x	+		x	+	x		x	+	x	x	+	x	+	
0	1	1		0	1	1			1	0	0		0	0	1		1	1	1	0	0	1	1	0
Ⓐ	Ⓑ	Ⓐ		Ⓐ	Ⓑ	Ⓐ			Ⓐ	Ⓑ	Ⓐ		Ⓐ	Ⓑ	Ⓐ		Ⓐ	Ⓑ	Ⓐ	Ⓑ	Ⓐ	Ⓑ	Ⓐ	
1																0	1						1	1

Fig. 13.16 Alice and Bob detect the presence of Eve

presence of Eve, the eavesdropper. Consequently, Alice and Bob decide to discard the keys they established to date.

13.5.4 Spedalieri's Orbital Angular Momentum Scheme for BB84

Photons carry both spin and orbital angular momentum. In 2004 Federico Spedalieri, then at NASA Jet Propulsion Laboratory, Caltech, recognized that the photon orbital angular momentum could be exploited to encode a bit. The photon orbital angular momentum [12] is related to light beams having an axially symmetric intensity structure about the direction of propagation of the photon but a complicated phase structure consisting of ℓ twisting intertwined phase fronts. Whereas the photon spin is related to its circular polarization, the photon orbital angular momentum is related to how much torque a photon can impart to a particle onto which it impinges. Moreover, whereas the spin can be described using a two-dimensional sub-space, and hence is a natural “qubit”, the photon orbital angular momentum can range over potentially infinity many states, and could therefore be used to implement higher base quantum logic, potentially enabling more efficient quantum computing and communications.

Photon orbital angular momentum is quite real: it has been harnessed to make so-called optical tweezers, which can hold, translate, and *rotate* very small particles using OAM states of light. A beautiful demonstration of the application of OAM states of light to rotate objects is to make a set of eight two-micron glass spheres perform the Scottish dance “Split-the-Willow”. A video of this feat set to music is found at [518].

In the original OAM paper Allen et al. discovered that in Laguerre-Gaussian light beam the orbital angular momentum was *quantized*, taking on possible values $\ell\hbar$ per photon, where ℓ is an integer [12]. This was later confirmed experimentally by Zeilinger et al. [334]. The potential for using OAM states of photons as a qudit

was noted at almost the same time [360]. In 2004 Spedalieri realized that, as there are infinitely many different OAM states, in principle, the orbital angular momentum degrees of freedom could be harnessed to implement the BB84 QKD protocol within a d -dimensional Hilbert space [476]. The information is encoded in the spatial modes of propagating photons, with different modes have different values of orbital angular momentum. The use of a d -dimensional Hilbert space can, in principle, boost the key generation rate by increasing the number of bits per photon that can be sent. In particular, whereas in a polarization encoding, one photon can carry one bit, in an OAM encoding one photon can carry $\log d$ bits of information. Although the gain is only logarithmic in the dimension d , Spedalieri suggests it may be possible to double or triple the key generation rate using current technology. A similar trick may be of use in other quantum information processing tasks. Moreover, as OAM states are invariant under rotations about the propagation direction, Spedalieri's scheme is independent of the alignment between sender and receiver. So the scheme does not require them to be in aligned reference frames. In fact, the protocol still works when these reference frames rotate with respect to one other. See [476] for complete details.

Unfortunately, in practice, OAM states are not well suited (so far) to use in long distance quantum communication. When OAM states propagate through fibers or free-space atmospheric paths, their delicate wavefronts can become distorted. Indeed, it is reported that when photons with no OAM propagated through a fiber-optic cable with a weight on it (i.e., “a stressed fiber”) the light acquires an OAM of $1\hbar$ per photon [348]. Similarly, even weak aberrations in atmospheric paths can damage OAM states propagating through the medium [378]. The magnitude of the effect depends upon the beam width relative to the coherence scale of the aberrations. Nevertheless, in laboratory conditions, OAM states are finding many uses in validating hypothetical quantum protocols. For example, Paul Kwiat used them to demonstrate quantum superdense coding [37], and others have proposed a quantum network that uses orbital angular momentum for routing and spin angular momentum for data encoding [482].

13.5.5 Generalization of BB84: Bruss' 6-State Protocol

The states used in the BB84 protocol, i.e., $|\Downarrow\rangle$, $|\leftrightarrow\rangle$, $|\nwarrow\rangle$ and $|\nearrow\rangle$, all lie in the same plane in the Bloch sphere. The dimension perpendicular to this plane is not used. The six-state protocol, invented by Dagmar Bruss in 1998, redresses this oversight [86]. It allows Alice to encode each random bit she wants to send to Bob in one of three non-orthogonal bases— $\{|\Downarrow\rangle, |\leftrightarrow\rangle\}$, $\{|\nwarrow\rangle, |\nearrow\rangle\}$, and $\{|\circlearrowleft\rangle, |\circlearrowright\rangle\}$ where $|\circlearrowleft\rangle = \frac{1}{\sqrt{2}}(|\Downarrow\rangle + i|\leftrightarrow\rangle)$ and $|\circlearrowright\rangle = \frac{1}{\sqrt{2}}(|\Downarrow\rangle - i|\leftrightarrow\rangle)$. Now a 0 bit can be encoded as the states $|\Downarrow\rangle$, $|\nwarrow\rangle$ or $|\circlearrowleft\rangle$ and a 1 bit can be encoded as the states $|\leftrightarrow\rangle$, $|\nearrow\rangle$, or $|\circlearrowright\rangle$. Hence the name “six state” protocol. Now the probability that Alice and Bob choose the same basis is only $\frac{1}{3}$ instead of $\frac{1}{2}$ for BB84, which increases the proportion of “dud” photons. But the six state protocol also reduces the optimal information gain for an

eavesdropper for a given quantum bit error rate (QBER). Indeed, if the eavesdropper measures every photon the QBER will be 33% versus 25% for BB84.

13.6 Bennett's 2-State Protocol (B92)

In 1992 Charles Bennett showed that one did not *need* to use four states (as in BB84) to support a QKD protocol, but that *two* non-orthogonal states were sufficient. This led to a QKD scheme known as the “B92 protocol” which is similar to BB84 except that two states are used instead of four, and the polarization measurement is done using polarizing filters rather than calcite crystals.

13.6.1 The B92 QKD Protocol

The B92 quantum key distribution protocol works as follows:

B92 Quantum Key Distribution Protocol

1. *Alice Generates Key Material:* Alice uses a *true* random number generator to create a long string of random bits. These are the raw bits from which Alice and Bob must distill a matching private key. Alice and Bob’s job is to determine a subset of these bits that they, and only they, will know in common. This privileged subset of bits becomes their private cryptographic key.
2. *Alice Encodes Key Material in Polarized Photons:* Alice sends each of her random bits to Bob, one after another, encoded in the polarization state of single photons. Alice and Bob agree on the following encoding strategy: If Alice wants to send Bob a 0 she transmits $|\Downarrow\rangle$, and if she wants to send a 1 she transmits $|\nearrow\rangle$ (thus only two non-orthogonal states are used).
3. *Bob Measures the Polarization of Each Incoming Photon Using a Polarizer:* Bob measures the polarization state of each incoming photon using a *polarizer*. A polarizer is an optical film that will pass a photon if its polarization is aligned with the polarization axis of the polarizer, but will absorb the photon (block it) if the photon’s polarization is orthogonal to the polarization axis of the polarizer. If the photon polarization is at some intermediate angle, the photon will be passed or blocked with some angle-dependent probability. Bob tests for whether Alice sent a 0 by orienting his polarizer so that it would pass the state $|\nwarrow\rangle$ and block the state $|\nearrow\rangle$. Conversely, Bob tests for whether Alice sent a 1 by orienting his polarizer so that it would pass the state $|\leftrightarrow\rangle$ and block the state $|\Downarrow\rangle$. Thus, whenever Bob’s polarizer is oriented to test for a 0, and Alice sent a 0, Bob detects this with probability $\frac{1}{2}$. But if Bob tests for 0 and Alice sent a 1, Alice’s photon will be absorbed and Bob will detect nothing from his polarizer. Likewise, if Bob tests for a 1 and Alice sent a 1, Bob detects this with probability $\frac{1}{2}$. But if Bob tests for 1 and Alice sent a 0, Alice’s photon will be absorbed and Bob will detect nothing from his polarizer.

4. Thus for each bit Alice sends, for Bob to detect the photon, he must pick the right basis (an event that occurs with probability $\frac{1}{2}$) and the photon must collapse into a state that allows transmission through the properly oriented polarizer (an event that occurs with probability $\frac{1}{2}$). Thus, overall, Bob will see only about $\frac{1}{4}$ of the photons Alice sent, but whenever he sees one he knows his polarizer was oriented the correct way to see Alice bit. Hence, by recording those events at which Bob detected anything, and interpreting the polarizer orientations back into bit values, Bon can learn a subset of the bits Alice sent, but Alice does not know which ones yet.
5. *Bob Discloses When he Detected a Photon:* To complete the key distribution, Bob tells Alice at which events he saw a photon but he does not disclose the orientation of his polarizer. However, Alice knows the polarization of the photons she sent and so can figure out the bit values Bob received. Hence, Alice and Bob come to know a common subset of random bits.
6. *Alice and Bob Test for the Presence of an Eavesdropper by Revealing a Subset of the Bit Values they Share:* But was anyone eavesdropping on the channel? If no-one was eavesdropping the channel was secure and the common bits can serve as a key. To test whether anyone was eavesdropping, Alice and Bob sacrifice a portion of the commonly known bits and compare their values. If there were no errors imperfections, Alice and Bob ought to agree on all these bit values. But if an eavesdropper were present then the error rate will increase. Hence by monitoring the error rate on the sacrificed bits, Alice and Bob can tell whether or not an eavesdropper was present. If not, they can discard the disclosed keys and use the remaining keys in a cryptographic protocol.

13.6.2 Threat of “Discard-on-Fail” Unambiguous State Discrimination

The B92 QKD protocol simplified the key distribution process and was initially applauded by experimentalists as an easier path to demonstrating QKD. Unfortunately, practical implementations of B92 tend to be open subtle security holes that require additional monitoring to plug. This is because, while it is true that one cannot distinguish between two non-orthogonal states unambiguously without inevitable disruption, one *can* distinguish them unambiguously if one discards failed attempts [109, 248, 254, 386]. An eavesdropper could therefore attempt unambiguous state discrimination and merely block further transmission of failed attempts. Of course this would increase the rate of losses on the channel. But it does mean users of the B92 protocol will need to monitor the channel losses very carefully, and be sure the eavesdropping equipment was not already in place at the inception of the channel, i.e., when its “native” loss characteristics of the channel were being assessed.

13.7 Ekert's Entanglement-Based Protocol

“[...] the generalized Bell's theorem can have a practical application in cryptography, namely, it can test the safety of the key distribution.”

– Artur Ekert [167]

In 1991 Artur Ekert, then at Oxford University, proposed a QKD scheme based on entanglement [167]. Before this, no-one had used entanglement as the means for key-generation and security verification. Although originally described in terms of spin- $\frac{1}{2}$ particles, Ekert's E91 protocol can be implemented more easily with polarized photons.

13.7.1 The E91 Protocol

Ekert's E91 quantum key distribution protocol works as follows:

E91 Quantum Key Distribution Protocol

1. An entanglement source sitting between Alice and Bob produces a stream of maximally entangled spin- $\frac{1}{2}$ particles each in the singlet state. One member of each pair is sent to Alice and the other to Bob along an axis we shall call the z -axis.
2. Alice and Bob each choose a measurement axis, randomly and independently, along which to measure the spin component of their respective particles. These measurement axes are not arbitrary. Rather they all lie in planes perpendicular to the z -axis, but differing in azimuthal angle as measured from the vertical x -axis. Alice is required to pick her measurement axis for each trial randomly and uniformly from the set \mathbf{a}_i ($i = 1, 2, 3$), having azimuthal angles of $\theta_1^{(a)} = 0$, $\theta_2^{(a)} = \frac{\pi}{4}$, $\theta_3^{(a)} = \frac{\pi}{2}$. Similarly, Bob is required to pick his measurement axis for each trial randomly and uniformly from the set \mathbf{b}_i ($i = 1, 2, 3$), having azimuthal angles of $\theta_1^{(b)} = \frac{\pi}{4}$, $\theta_2^{(b)} = \frac{\pi}{2}$, $\theta_3^{(b)} = \frac{3\pi}{4}$. If either Alice or Bob fail to detect a particle, they discard that trial, so the results are always in perfect lock-step with each other. Each spin-measurement returns a result (in units of $\frac{\hbar}{2}$) that is either +1 (spin-up) or -1 (spin-down), which Alice and Bob record.
3. Next Alice and Bob announce publicly what spin-axis orientations they used for their respective measurements, but they do not disclose what values they obtained for those measurements. This provides enough information to extract a key. To see this, realize that in making these spin-orientation measurements, $\frac{2}{9}$ of the time Alice and Bob pick axes that are aligned, and $\frac{7}{9}$ of the time Alice and Bob pick axes that are not aligned. Regardless, the correlation between these measurements can always be written as:

$$E(\mathbf{a}_i, \mathbf{b}_j) = P_{++}(\mathbf{a}_i, \mathbf{b}_j) + P_{--}(\mathbf{a}_i, \mathbf{b}_j) - P_{+-}(\mathbf{a}_i, \mathbf{b}_j) - P_{-+}(\mathbf{a}_i, \mathbf{b}_j) \quad (13.11)$$

where $P_{+-}(\mathbf{a}_i, \mathbf{b}_j)$ (and similar terms) is the probability that Alice measures +1 and Bob measures -1 when the spin-measurement directions are \mathbf{a}_i for Alice and \mathbf{b}_j for Bob. According to quantum mechanics, in all cases:

$$E(\mathbf{a}_i, \mathbf{b}_j) = -\mathbf{a}_i \cdot \mathbf{b}_j \quad (13.12)$$

Thus, when Alice and Bob use analyzers that are *aligned similarly* (i.e., when Alice uses \mathbf{a}_2 and Bob uses \mathbf{b}_1 , or when Alice uses \mathbf{a}_3 and Bob uses \mathbf{b}_2) their spin-measurement results will be perfectly anti-correlated, i.e., $E(\mathbf{a}_2, \mathbf{b}_1) = E(\mathbf{a}_3, \mathbf{b}_2) = -1$. Therefore, once Alice and Bob disclose what spin-axis orientations they used for their respective measurements, they can quickly determine those cases in which they used the same orientations. In these cases, if Bob (say) then flips his results (which are anti-correlated with those of Alice), Alice and Bob will obtain a mutually shared private key that could be used in a subsequent private key cryptosystem. So far so good.

4. To complete the protocol, Alice and Bob need to be sure no-one was eavesdropping on the channel. To test for the presence of an eavesdropper, Alice and Bob use those cases in which they had picked *different* settings for their spin-polarization analyzers. In these cases they reveal both the spin-axis orientations used *and* the results of those measurements. Using just a subset of these “useless” results, Alice and Bob can estimate a composite quantity that combines the correlation values at different settings of the measurement directions. As any attempt at eavesdropping must necessarily break the entanglement between the particle pairs in Alice and Bob’s possession, Alice and Bob can detect the presence of an eavesdropper by testing the degree of entanglement between their “useless” results. Specifically, Alice and Bob compute:

$$S = E(\mathbf{a}_1, \mathbf{b}_1) - E(\mathbf{a}_1, \mathbf{b}_3) + E(\mathbf{a}_3, \mathbf{b}_1) + E(\mathbf{a}_3, \mathbf{b}_3) \quad (13.13)$$

If the respective particle pairs are entangled (as they should be if no eavesdropper is present) quantum mechanics predicts $S = -2\sqrt{2}$. However, if eavesdropping occurred, the entanglement between Alice and Bob’s particles will have been broken and S will deviate from the value $-2\sqrt{2}$. Thus, Alice and Bob can use pairs of entangled particles, measurements in non-orthogonal bases, and classical communication to both generate matching random cryptographic keys and test for the presence of an eavesdropper.

A simplified version of Ekert’s E91 protocol, in which Alice and Bob each use only two non-orthogonal measurement axes instead of three, can implement the BB84 protocol. The E91 protocol is illustrated in Fig. 13.17.

13.8 Error Reconciliation and Privacy Amplification

In a perfect world, any of the idealized QKD protocols described above, i.e., BB84, B92, or Ekert91, would leave Alice and Bob in possession of *identical* steps of random keys. Unfortunately, in the real-world noise sources tend to introduce errors

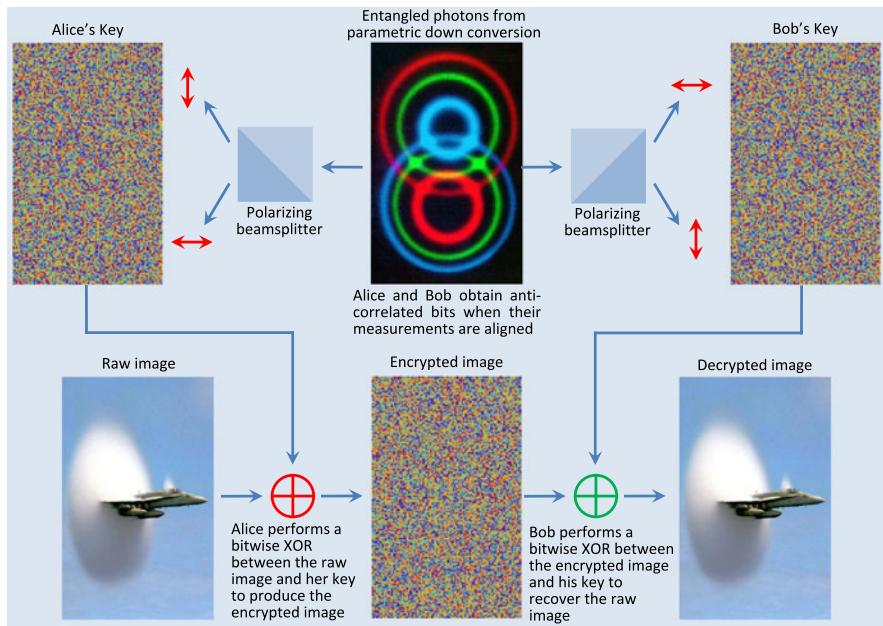


Fig. 13.17 This figure needs to be redrawn showing similar concept. Source Physics World July 2008. The figures shows the basic idea of entanglement-based QKD. A common source creates a stream of pairs of maximally entangled particles, and sends one of each pair to Alice and the other to Bob. Alice and Bob each make independent and random measurements on their particles. On those occasions when they happen to have used the same alignment of their measuring apparatus they will create perfectly anti-correlated bits. By one of them (Bob say) flipping his bits they can therefore create a secret shared random cryptographic key. When this key is XOR-ed with an image, it produces nonsense. But when XOR-ed with the correct key will recover the image. Alice and Bob use the times that did not pick the same orientation are thrown out

that Alice and Bob cannot distinguish from eavesdropping. So after the raw key material has been established using a QKD protocol, Alice and Bob must engage in two additional tasks—error reconciliation and privacy amplification—aimed at eliminating discrepancies between the sets of keys, and then thwarting an eavesdropper who might have gleaned additional information about the key by listening in on the public error reconciliation process.

13.8.1 Error Reconciliation

The tricky part of error reconciliation is that Alice and Bob want to fix discrepancies amongst their respective sets of cryptographic keys without disclosing much information about the remaining “good” keys. In the worst case scenario, Alice and Bob must assume that *all* the discrepancies between their cryptographic keys are due to a malicious eavesdropper, Eve. This is the safest assumption to make although it is

hardly realistic given the inevitable noise sources along quantum communications channels such as fiber-optic cables or atmospheric paths. Nevertheless, it pays to be paranoid in cryptography so let's assume the worst.

Thus, to eliminate discrepancies without disclosing too much Alice and Bob communicate publicly about the *parities* of blocks of their key material, using, e.g., the CASCADE protocol [73, 483] (explained below). For CASCADE to work well one needs a pretty good estimate of the probability a bit at random differs in the two key strings. To obtain this, Alice and Bob sacrifice a portion of their key material by revealing corresponding bits publicly. This provides a crude estimate p^* of the probability a bit is different. Given this estimate of p^* CASCADE proceeds as follows:

Error Reconciliation

1. Use p^* to choose a block size into which to segment the two sets of keys, K_A and K_B .
2. Compute the parity of each block and compares their values publicly.
3. Whenever the parities computed are different there must be an odd number of bit errors between the respective blocks from K_A and K_B . Therefore, divide the offending blocks iteratively and recurse on each of the respective smaller blocks until a single error in a block is isolated. At that point, flip that bit.
4. If an error is found in a block that had, in a previous round, been found to have the correct parity then another error must be contained in that block. Such errors can be eliminated by shuffling the bits and re-running steps 2, 3, and 4.
5. Repeat step 5 until one attains confidence no further errors remain. At this point error reconciliation is complete.

13.8.2 Privacy Amplification

Unfortunately, the error reconciliation process places a lot of information in public view regarding the parities of various blocks of key material. Moreover, Eve may also have obtained some information about the keys based on her original eavesdropping during the QKD protocol. If Eve is sufficiently mathematically gifted she could, from this information, infer better information about the supposedly secret error-reconciled key material. Hence, Alice and Bob need to take some final action to deprive Eve of any significant information about the cryptographic keys Alice and Bob will eventually use. To do so, Alice and Bob perform a round of “privacy amplification”.

Privacy amplification uses Alice and Bob's error-reconciled key to generate a different, shorter key, in such a way that Eve has no significant information about the new (shorter) key [50]. The trick is for Alice and Bob to choose (randomly and publicly) from one of a pre-agreed set of hash functions. The hash functions must accept a bit string whose length matches that of the error-reconciled key, and return a

shorter bit string, with the degree of compression dependent on how much information Eve could have obtained from eavesdropping on the QKD protocol plus monitoring the public discussions between Alice and Bob during the error-reconciliation process.

Privacy Amplification

1. Alice and Bob choose publicly and randomly from a pre-agreed library of hash functions.
2. Alice and Bob use this hash function to map the (error-reconciled) keys to shorter keys (i.e., matching bit strings) that leave Eve with virtually no information about them. These matching bit strings become the final cryptographic keys Alice and Bob will use in future secure communications using a classical cryptosystem.

Although the OTP is the gold-standard cryptosystem, because it is unconditionally secure, it is also a voracious consumer of key material, requiring one key bit per message bit. There are, however, other classical cryptosystems, such as AES, which are pretty secure but not unconditionally secure. So many people have proposed using QKD to increase the frequency of re-keying these less secure (but still good) classical cryptosystems. Strictly speaking, the use of AES sacrifices absolute security compared to the OTP. Nevertheless, it allows the user increase the potential bandwidth for encrypted traffic dramatically as AES consumes much less key material than OTP. Thus, the ability to re-key AES frequently boosts the effective security of AES considerably, making an already good classical cryptosystem even better.

13.9 Implementations of Quantum Cryptography

It is rare for exclamation marks to appear in the titles of scientific papers, yet in 1989 Charles Bennett and Gilles Brassard published a landmark paper entitled “The Dawn of a New Era for Quantum Cryptography: The Experimental Prototype is Working!” [48]. Figure 13.18 shows a schematic of the device. In an after-dinner speech at the 1994 Physics of Computation Conference, Gilles Brassard recalled, whimsically, that the original machine was not that secure after all, as the devices used to place photons in particular polarization states made noticeably different *noises* depending on the type of polarization selected!

13.9.1 Fiber-Optic Implementations of Quantum Cryptography

Fortunately, such technological quirks have not impeded progress in quantum cryptography. Since 1984 there have been dozens of experimental demonstrations of quantum key distribution (QKD) by research teams from around the World including the U.S.A., U.K., Europe, Japan, China, Scandinavia, and Australia. These

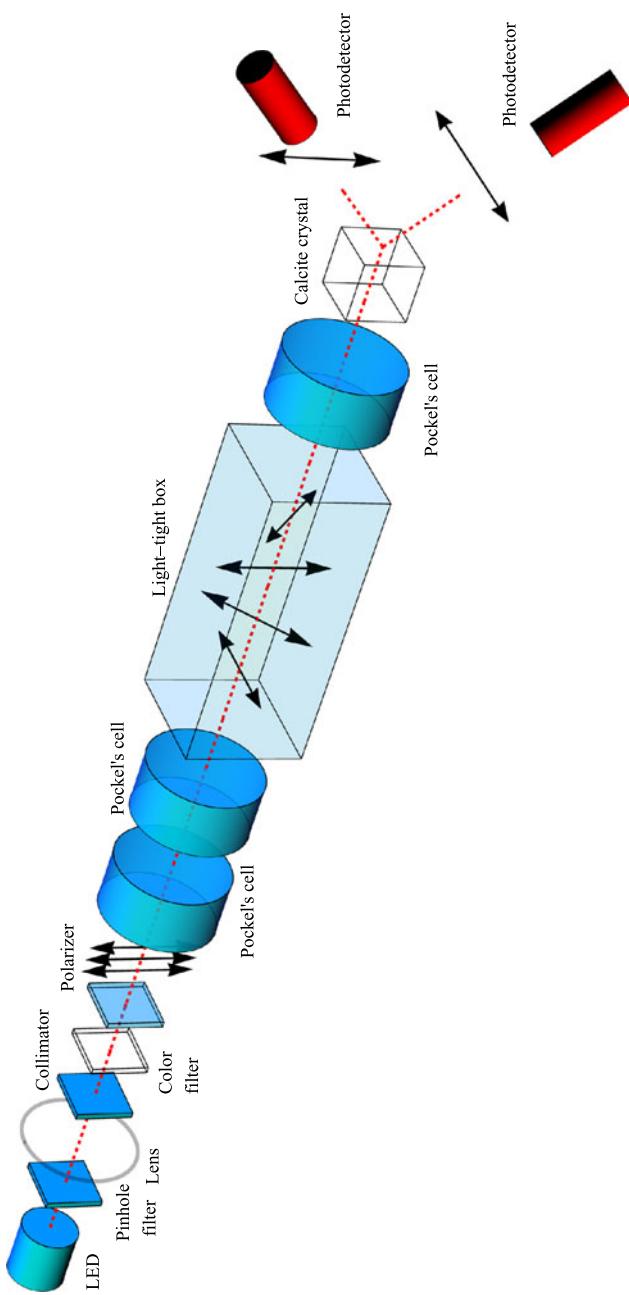


Fig. 13.18 Schematic of the first quantum cryptography prototype. The actual device is about a meter long

demonstrations have extended the range of QKD over both standard telecom networks [171, 246] and atmospheric line-of-sight paths [438], and developed ancillary methods for routing and managing the keys. Indeed the “DARPA Quantum Network”, built by a collaboration between BBN, Boston University, Harvard University, and QinetiQ has been running continuously since 23rd October 2003, and now has ten nodes scattered around the Boston area, including a wireless link supplied by QinetiQ in 2005 [171]. Similarly, the European SECOQC network, demonstrated in 2008, uses 200 km of fiber-optic cable linking six sites around Vienna [246].

To date, the longest range over which an error-corrected, privacy-amplified, cryptographic key has been transmitted via fiber-optics is 148.7 km of dark optical fiber at a mean photon number $\mu = 0.1$, and an astonishing 184.6 km of dark optical fiber at a mean photon number of $\mu = 0.5$. The same team has also exchanged secret keys over 67 km that are guaranteed to be secure against photon-number-splitting attacks [233, 382, 424]. Yamamoto’s group boasts a longer range still but their demonstration did not involve error reconciliation and privacy amplification [236]. Similarly, the highest key generation rate was achieved in a system using the BB84 protocol with so-called “decoy pulses”. This system boasts an impressive key distribution rate of 1.02 Mbit/s over 20 km, and 10.1 kbit/s over 100 km [151].

These achievements make QKD practical on the scale of local area networks typically found in the financial districts of major cities. If financial centers have a quantum key infrastructure in place, banking transactions could be encrypted in a manner that would be utterly impossible to break regardless of the mathematical sophistication, computational power, or algorithmic prowess of any eavesdropper—the ultimate in secure banking! In fact, the first quantum-encrypted banking transaction took place in Vienna on 21st April 2004 when Anton Zeilinger et al. succeeded in transferring a donation of 3,000 euro from City Hall to a branch of the Bank of Austria, in Vienna [435]. This appears to have piqued the interest of the Bank of England and the U.K. Department of Trade and Industry who held meetings in 2005 at which Toshiba Research Europe, idQuantique, MagiQ, and QinetiQ discussed the viability of using QKD within the communications infrastructure of London’s financial district—the “City” [545].

13.9.2 Extending the Range of QKD with Quantum Repeaters

To date the distances over which direct quantum cryptography has been demonstrated in fiber networks has been limited to around 100–150 km [233, 382, 424]. Going beyond this becomes progressively more difficult because the probability of photon absorption and the probability photon depolarization both grow exponentially with the length of the fiber-strand. Thus, to transmit a photon down a fiber successfully will require a number of trials that grows exponentially with the length of the fiber. Moreover, when such a photon finally makes it through, the fidelity of its polarization state will be degraded exponentially with the length of the fiber. Eventually the probability of detecting the photon drops so low that one can no longer

distinguish between a true click and a random dark count of the detector. At this point QKD becomes quite impractical.

The “obvious” solution would be to do what we do classically, i.e., amplify the signal periodically as it passes along the fiber. However, amplification (in the sense of increasing the intensity of the signal by inserting more photons of the desired state) is not allowed quantum mechanically due to the No-Cloning theorem. Moreover, even if it were possible, creating redundant photons to encode a given state would undermine the security of the QKD protocol. It would seem, therefore, that quantum cryptography can only be used on local area networks where inter-node links are short enough to maintain a reasonable key distribution rate.

A brilliant solution to this problem was found in 1998 when Hans Briegel, Wolfgang Dür, Ignacio Cirac, and Peter Zoller (BDCZ) invented a physically-allowed scheme for a “quantum repeater”. This combines the ideas of entanglement swapping with entanglement purification to conceive of a protocol by which long-range, high-purity, entanglement can be established over very long fiber-optic links [83, 160]. The basic idea behind the BDCZ quantum repeater is shown in Fig. 13.19.

First one creates many short-range maximally entangled pairs of photons along segments of the quantum communications channel, and then uses entanglement swapping where two segments abut to extend the range of entanglement. In practice, entanglement swapping will introduce small errors that must be eliminated by applying entanglement purification [51, 52, 139, 268], which sacrifices some entangled pairs in order to increase the purity of the entanglement amongst the ones that remain. Importantly, the scheme requires only a polynomial overhead in time, and a logarithmic overhead in the number of qubits that must be manipulated per node.

Several rudimentary quantum repeaters have now been demonstrated. Some schemes have used cavity QED to achieve a quantum memory to couple a flying qubit (a photon) to a stationary qubit (an atom in the cavity) [106, 108]. However, in 2001 Duan-Lukin-Cirac-Zoller (DLCZ) proposed a new scheme for a quantum repeater based on atomic ensemble quantum memories and linear optical quantum computing elements [153]. This approach was demonstrated experimentally in 2008 [555]. However, lately cavity QED has re-emerged as a contender because, in theory, the rate at which remote entangled pairs can be established is significantly higher than schemes based on atomic ensemble memories [428]. One way or another long-haul quantum communications in fiber appears quite feasible at this point.

13.9.3 Earth-to-Space Quantum Cryptography

Another way to extend the range of quantum cryptography is to create a QKD-link between a ground station A and an orbiting satellite, establish N matching cryptographic keys with the satellite, and then wait for the satellite to continue in its orbit until it is within range of a second ground station, B remote from the first. If

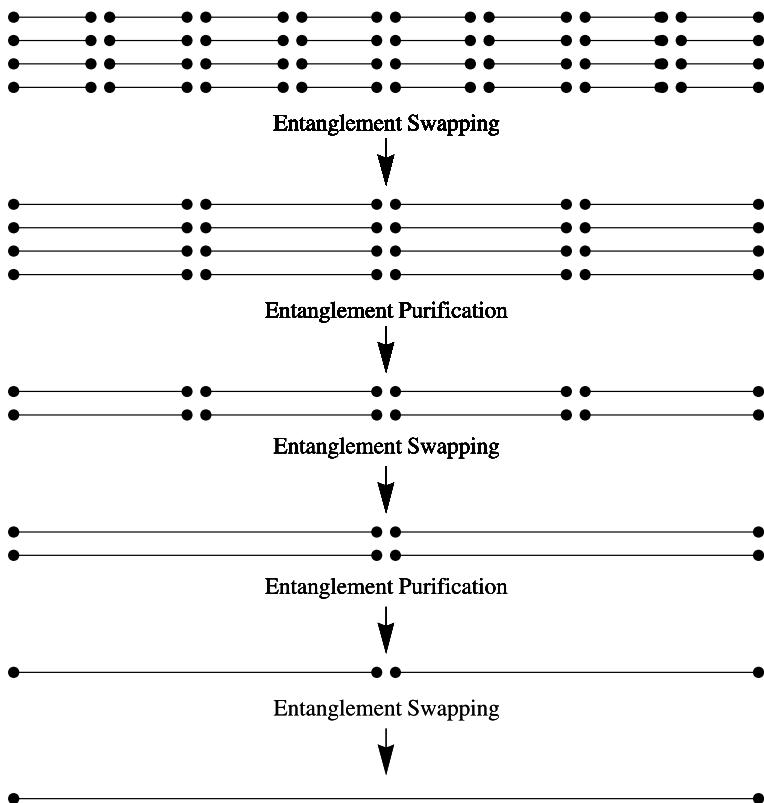


Fig. 13.19 Illustration of the stages of a quantum repeater. Initially, many short-range entangled pairs are created at multiple sites along the communications channel. Then entanglement swapping is used to extend the range of entanglement. However, this process introduces errors which are removed by applying entanglement purification at the cost of sacrificing some of the entangled pairs so that the quality of the remaining ones can increase. Then, another round of entanglement swapping extends the range of entanglement further yet again. This too introduces errors which must be removed by entanglement purification. The process of interleaving entanglement swapping and entanglement purification continues until maximally entangled pure states are established across the entire length of the quantum channel, potentially at a distance far in excess of that which is attainable in a one-shot round of entanglement distribution

the second station B then also establishes N keys with the satellite, the satellite can thereafter transmit the first set of N keys to B securely. Thereafter, A and B will be in possession of identical sets of cryptographic keys.

During the time it takes a satellite to cross the field of view of a groundstation, calculations suggest that it ought to be possible to exchange a few tens of thousands of key bits, from which a few thousand error-free key bits can be distilled. The noise on the atmospheric channel comes partly from atmospheric turbulence, and partly from background stray photons that can be confused with the photons encoding the key material. Hence error reconciliation and privacy amplification are essential. This

is enough key material to re-key strong cryptosystems such as AES often enough to significantly boost their already good security still farther.

The first demonstration of QKD over practically significant distances was performed in 2002 in a technical a tour-de-force experiment by Richard Hughes, Jane Nordholt, Derek Derkacs, and Charles G Peterson of Los Alamos National Laboratory [247]. They demonstrated true QKD over a 10 km, 1-airmass atmospheric range during daylight and at night time conditions and were able to use the experimentally achievable parameters (for air-mass extinction, background optical noise, and achievable optical component quality) to infer that free-space QKD to satellites ought to be feasible.

In 2006 and 2007, in two back to back experiments, a team of European scientists exceeded the Los Alamos National Laboratory distance record by an order of magnitude performing QKD over a 144 km free-space path between the islands of La Palma and Tenerife in the Canary Islands. The experimental set up is depicted in Figs. 13.20 and 13.21. One experiment implemented the Ekert91 QKD protocol, which required establishing polarization-entangled pairs of photons some 144 km apart [504] and the other experiment implemented the BB84 QKD protocol using weak coherent pulses and decoy-states to guarantee the security of the channel [322, 438], establishing a secure key at a rate of 12.8 bits per second. The experiments required the use of a green laser beam tracking system to compensate for the atmospheric variability. There is little doubt after these achievements that Earth-to-satellite QKD is feasible, in principle, as atmospheric density decreases further with altitude making Earth-to-satellite optical paths waver less than those of equal length wholly within the Earth’s atmosphere. A first step in a true Earth-to-satellite QKD exchange occurred in 2008 when a single photon exchange (albeit not true QKD) was demonstrated between the Matera Laser Ranging Observatory in Italy to the Ajisai satellite (which resembles an orbiting disco ball), at some 1485 km above the Earth [519]! Although this experiment does not implement a full QKD protocol, it does show the feasibility of sending and receiving single photons between a satellite and a ground station on the Earth.

There is now considerable interest in exploring the potential of Space for conducting fundamental tests of quantum physics. This was summarized in a 2008 study [505], and a proposal has been made to demonstrate Space-based quantum communications on the International Space Station [383]. Clearly, the state of the art in Earth to Space quantum communications is maturing rapidly and already practical demonstrations are possible.

13.9.4 Hijacking Satellites

However, a more compelling reason to create QKD links with satellites is to maintain proper command and control of orbital assets. If a communications system for a satellite is primed with an authentication key before launch then thereafter, if an Earth-to-satellite QKD mechanism is available, the authentication key can be

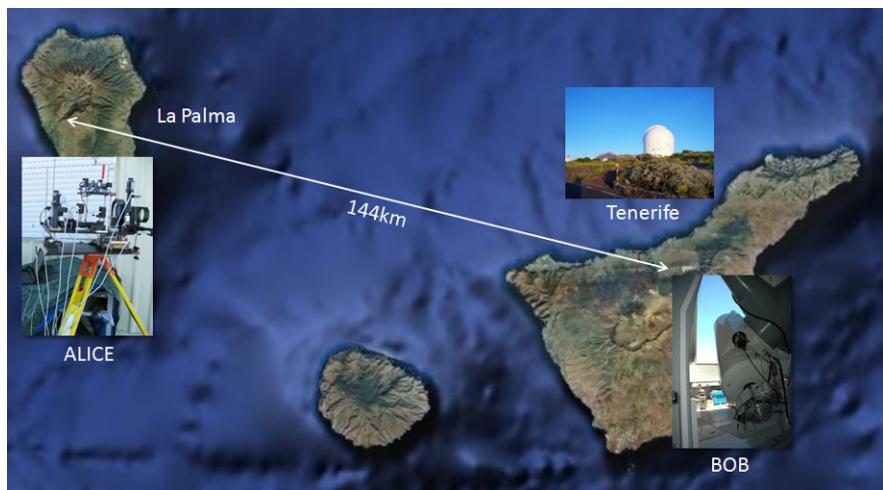


Fig. 13.20 Free space QKD over a 144 km free-space optical path between the islands of La Palma and Tenerife in the Canary Islands. Alice is on La Palma and Bob on Tenerife (Composite photographs copyright the Institute for Quantum Optics and Quantum Information, Austrian Academy of Sciences, and printed with their permission)

changed each time it is used in a manner that has no pattern whatsoever and hence cannot be inferred by anyone monitoring the uplink communications to the satellite. If instead a fixed authentication key is used, or one that is changed according to some deterministic algorithm, then potentially the authentication key can be predicted and an adversary could upload malicious commands to take control of the satellite and make it do something awry.

It might seem preposterous to imagine that satellites could be hijacked in this manner. But an incident that occurred in early hours of April 27th 1986 sent chills down the spines of those entrusted to protect our satellite services [490]. This occurred at a time when satellite television was relatively new and there had been something of a boom in sales of satellite dishes to home owners who wanted to intercept the free television signals raining down from the sky. However, some satellite television companies, such as HBO, had realized that they could monetize their services more effectively by scrambling the signals electronically thereby requiring homeowners to buy specialized decoder equipment and pay a monthly fee. This shift in policy caused something of a slump in satellite dish sales and a loss in revenues to dish retailers.

One such retailer was John MacDougall, then just 25 years old, who saw dish sales plummet and his hopes living the American dream fade. To make ends meet, he took a job as an engineer at Central Florida Teleport in Ocala Florida that uploaded pay-per-view movies for satellite operators. At 12:32AM April 27th, after watching the movie “Pee-wee’s Big Adventure” as it was uploaded, MacDougall began his routine shut down procedure, which involved setting up transmission of a test card of colored bars, and rotating the large 30 foot satellite dish he had been

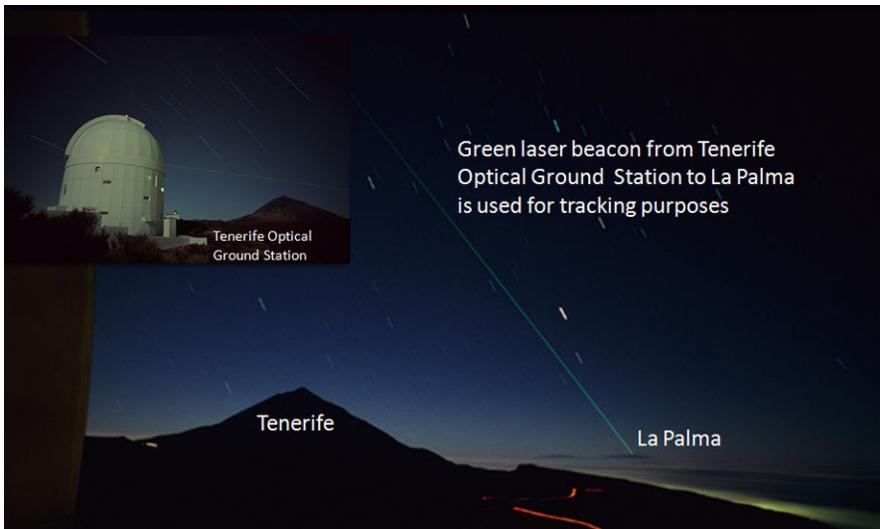


Fig. 13.21 Free space QKD over 144 km requires beam tracking to compensate for atmospheric disturbances. The figure shows a green laser beacon sent from the Tenerife optical ground station to the QKD source in La Palma 144 km away. The laser is used to implement a beam tracking mechanism. The photograph and insert were taken by the team members (Composite photographs copyright the Institute for Quantum Optics and Quantum Information, Austrian Academy of Sciences, and printed with their permission)

using to its parked position in which rainfall could run off it easily. In this position the dish happened to be pointing at the Galaxy 1 satellite used by HBO. At that moment, with the decline of his business still gnawing at him, MacDougall decided to teach HBO a lesson. He had the idea of overriding the HBO signal coming from the Galaxy 1 satellite with a stronger signal he could upload from Central Florida Teleport. So overlaid the colored bars of the test card signal with a message and launched it at Galaxy 1. For four and a half minutes the HBO satellite television signal, which was showing “The Falcon and the Snowman” movie at the time, was overridden with the following message (see Fig. 13.22):

```

GOODEVENING HBO
FROM CAPTAIN MIDNIGHT
$12.95 / MONTH ?
NO WAY !
[ SHOWTIME/MOVIE CHANNEL BEWARE! ]

```

The HBO engineers realized the problem and increased the power of their transmission to wrestle control of the channel back from the mysterious “Captain Midnight”.

However, the content of the message concerned the authorities less than the fact it was possible to take control of the channel. This was not merely a jamming incident. Captain Midnight had replaced a legitimate signal with one of his own choosing. That was serious. Moreover, in the following days, when Captain Midnight’s

Fig. 13.22 Captain Midnight's message to HBO



identity was still unknown, HBO claimed to have received threats that their Galaxy 1 satellite would be re-positioned. They could not tell whether these threats were credible or not, and millions of dollars were on the line.

In the ensuing clamor for his head, Captain Midnight's identity was bound to become known. The FBI was set on his trail and found him using some clever detective work [564]. To be able to override the HBO signal Captain Midnight would need access to a powerful transmitter connected to a satellite dish at least a seven meters in diameter. This reduced it to 580 possible sites. Then by studying the character font used in Captain Midnight's message they determined the make and model of the character generator. This reduced it to 100 possible sites. HBO reported a brief intrusion of colored bars on their channel at the same time one week earlier, which they speculated might be the work of the same person. This meant the culprit was on duty at *two* known times, which narrowed the pool again. A final clue came when an accountant overheard an incriminating conversation regarding Captain Midnight at a pay phone and gave the license plate number of the caller to the authorities, who linked it to MacDougall. Thus, Captain Midnight's identity was revealed. He was fined \$5,000 and put on probation for one year.

Concerns over command and control of our satellite systems have continued. The August 2nd, 2004 issue of New Yorker magazine (pp. 40–53 [370]) contained an article entitled “The Terror Web”, by Lawrence Wright. This reported on an interview with French anti-terrorist judge Jean Louis Bruguiere, who is quoted about the role of Chechnya in the worldwide jihad:

“At present, Al Qaeda and its affiliates operate on a rather low-tech level, but in Chechnya many recruits are being trained to exploit the technical advantages of developed countries. ‘Some of these groups have the capacity for hijacking satellites,’ he told me. Capturing signals beamed from space, terrorists could devastate the communications industry, shut down power grids, and paralyze the ability of developed countries to defend themselves.”

Such concerns illustrate that future battles might involve, paradoxically, the simultaneous use of very high-tech and very low-tech tactics.

Even NASA has had its share on incidents. NASA acknowledged a BBC report that a hacker was able to overload its computers and interrupt communication on a space shuttle mission in 1997. Although NASA disputes the BBC’s interpretation of the significance of the event, they acknowledged that “the transmission of routine medical information was slightly delayed”, confirming that a disruption to space communications had occurred [243]. The same article contains links to several reports on hacking attacks on NASA ground stations [241, 242, 244].

A list of hacking attacks on civilian satellite systems is reported in a 2002 GAO report entitled “Critical Infrastructure Protection: Commercial Satellite Security Should Be More Fully Addressed”, [404]. These include deliberate denial of service attacks, such as when Indonesia intentionally interfered with and denied the services of a commercial satellite belonging to the South Pacific island kingdom of Tonga because of a satellite orbital slot dispute. Incidents also include inadvertent denial of service due to unintentional interference, such as when a GPS transmitter being tested on the ground, unintentionally interfered with the GPS receivers of a commercial aircraft in the area, causing the plane to lose all of its GPS information temporarily.

These examples illustrate the potential vulnerability to our satellite systems to a spectrum of threats. Although Earth-to-satellite QKD cannot overcome all of these threats, e.g., denial-of-service attacks cannot be defeated, it can contribute to ensuring a firmer grip on command and control of orbital assets.

13.9.5 Commercial Quantum Cryptography Systems

The perceived need for greater network security, and the proven ability to achieve higher security using quantum cryptography running over standard fiber-optic networks, has inspired a handful of entrepreneurs to launch start-up companies offering security solutions that include quantum cryptography. These include MagiQ Technologies based in New York City and Sommerville, Massachusetts [333], idQuantique, a spin-off from the University of Geneva, Switzerland [249], a relatively new arrival, SmartQuantum of Brittany, France [462], and another new arrival, Quintessence Laboratories of Australia [406].

MagiQ, whose slogan is “Any sufficiently advanced technology is indistinguishable from magic” (Arthur C. Clarke), sells their “QPN™8505 Security Gateway” [469]. This is based on the BB84 QKD protocol and provides a refresh rate of 100 256-bit AES keys per second, over a metro-area scale (100 km natively to 140 km using the “decoy state” protocol), has an intrusion detection capability, and uses the QKD-distributed keys to encrypt messages using the triple DES and AES cryptosystems.

Similarly, idQuantique sells an impressive variety of quantum devices in addition to their flagship quantum cryptography product “Cerberis” [468]. Cerberis is based on the BB84 and SARG QKD protocols [433], provides a refresh rate of one key per minute for up to 12 encryption appliances, over a 50 km length of dark fiber, and uses those keys within the 256-bit AES cryptosystem.

SmartQuantum offers three QKD-related products—the “SQKey Generator” (for QKD-based key exchange), the “SQDefender” (for an end-to-end quantum cryptography solution that combines QKD with digital cryptography for civilian markets), and the enhanced “SQFibreShield” (for military-grade cryptography). The company is developing a worldwide reseller network and has recently established offices in Sunnyvale, California in the heart of Silicon Valley [402].

Quintessence Labs is the newest arrival and expects to be in production by early 2010. In June 2009 Quintessence Labs announced a partnership with Lockheed Martin to develop QKD solutions for the U.S. and Australian markets [401]. Quintessence is unique in basing their QKD systems on continuous quantum variables rather than single photon sources and detectors. This allows them to imprint a cryptographic key on a bright laser beam (rather than single photon transmissions) at standard telecom wavelengths (1550 nm) using standard commercial-off-the-shelf (COTS) components and standard fiber-optic networks. By using conventional laser beams to carry the key material they can extend the range of QKD in fiber dramatically while simultaneously increasing the key generation rate to the point that direct use of the One Time Pad becomes feasible. This has the potential to slash the costs of QKD and make it even easier to integrate into conventional fiber optics networks, perhaps even allowing wavelength division multiplexing to further enhance the key generation rate. All exciting developments for this fledgling industry!

13.10 Barriers to Widespread Adoption of Quantum Cryptography

The fledgling quantum cryptography industry faces some pretty steep hurdles in gaining market share over the conventional PKI infrastructure. While most of these hurdles can be overcome over time, it is going to make it challenging for quantum cryptography companies to gain traction in the short term. Most likely, the early adopters will be those customers who are most in tune with the unique characteristics of the solution quantum cryptography offers. That is, those customers for whom the *perpetual* security of potentially intercepted communications is a prime issue.

13.10.1 Will People Perceive a Need for Stronger Cryptography?

The first obstacle is one of perceived need. Quantum cryptography systems establish matching cryptographic keys across potentially insecure communications channels, and then use those keys within a strong cryptosystem to ensure the perpetual confidentiality messages encrypted subsequently with the keys. This makes quantum cryptography optimally suited for securing information, such as military and diplomatic communications, that must be kept confidential indefinitely. However, will a large number of customers perceive the need for the level of security offered by

quantum cryptography? Even though eavesdropping is potentially a problem it does not, at present, seem to deter electronic commerce. This means the public must perceive the existing PKI infrastructure as *secure enough* for routine electronic transactions. So long as the financial losses from eavesdropping are manageable, e.g., by banks passing on the losses to their customers in the form of fractionally higher interest rates for everyone rather than the few affected directly, or by identity theft being repairable, a fundamentally insecure system can (and does) prevail. However, quantum cryptography is likely overkill for the majority of message, like email, whose value, even if intercepted, is transient.

13.10.2 Will People Believe the Foundations of QKD Are Solid?

Even if the *need* for absolute security *does* become perceived widely, companies commercializing quantum cryptography still have to convince skeptical information technology managers charged with protecting networks that the laws of quantum mechanics on which the security of their QKD schemes are based cannot, in fact, be circumvented. Such a possibility is unthinkable to most quantum physicists because if there was a way to elude quantum non-determinism or the quantum no-cloning theorem we could perform feats, such as superluminal communication, which Nature appears to abhor at the macroscopic scale. Yet quantum mechanics is so alien to most people that many IT managers will be asked, in effect, to place their trust in a technology they do not understand, and whose security relies upon our current understanding of physics, which historically has undergone radical shifts more than once. This disconnect is potentially an impediment to adoption of QKD technology.

13.10.3 Will People Trust the Warranties of Certification Agencies?

One way to address such security concerns would seem to be to entrust the verification of the security of quantum cryptography systems to third party certification agencies. However, as quantum cryptography is still relatively new, and as ways to attack a quantum cryptography system are potentially more subtle than the ways to attack a conventional cryptosystem, will customer have confidence in the security warranties issued by certification agencies? In particular, even if one believes the underlying theoretical quantum cryptography protocols are secure, any real quantum cryptography hardware is likely to harbor slight imperfections, which have the potential to introduce unanticipated security loopholes. So while a quantum cryptography *protocol* may be secure, any given *realization* of it in hardware may not and it may be rather difficult to anticipate the loopholes such imperfections open up.

Certainly, researchers have long recognized the possibility of such loopholes and have actively sought to find them [329, 335–338, 432, 549]. For the most part the proposed attack scenarios have been rather hypothetical in nature, and have required

an adversary to possess extraordinary technical abilities. Moreover, in most cases the attack itself would exhibit a detectable signature. However, in August 2010 a particularly strong attack was published by the “Quantum Hacking” group at the Norwegian University of Science and Technology in collaboration with teams from the University of Erlangen-Nürnberg and the Max Planck Institute [331]. This attack was unusual in that it can be mounted using off the shelf components, was quite undetectable, and was effective, at the time, in revealing the full key material generated by two commercially available systems, namely, MagiQ Technology’s QPN 5505 system, and ID Quantique’s Clavis2 system. Luckily, prior to publishing the details of the attack the quantum hackers collaborated with idQuantique to develop suitable countermeasures [251]. So this particular vulnerability has been addressed. Nevertheless, the fact that a quantum cryptography system that was commercially available could be hacked successfully points to the need to conduct much greater testing of quantum cryptosystems, and the need to develop a clear set of standards for certifying their security. Fortunately, the manufacturers of quantum cryptosystems welcome such hacking attempts as, ultimately, they will make quantum cryptosystems as strong in practice as they are in principle [250]. Work on hacking quantum cryptography systems continues [330, 341, 529] and this is likely to be essential in bolstering confidence in these systems.

13.10.4 Will Wide Area Quantum Cryptography Networks Be Practical?

Another concern in deploying quantum cryptography on a wide scale is the need, to be truly secure, to use an *authenticated* channel. If the channel is not authenticated, quantum cryptography systems will be vulnerable to “man-in-the-middle” attacks, wherein an eavesdropper sitting between Alice and Bob executes independent QKD protocols with them both while pretending to be the other party. To overcome this, one needs to have some means to authenticate the person you are talking to and this requires the establishment of a shared prior secret, or the exchange of a small amount of truly random key material face-to-face, prior to the onset of the quantum key distribution. This is an acceptably small price to pay for dedicated channels known in advance, but it makes it difficult to see how proper QKD would be practical on a massive scale wherein one wishes to communicate with assured security to someone you have not met with face-to-face beforehand and agreed upon one of more authentication keys.

Moreover, another current practical impediment to widespread adoption of QKD is that the range of QKD in fiber is limited to around 100km until true quantum repeaters arrive. However, this concern can be alleviated somewhat by using continuous quantum variables (as in done by Quintessence Labs), or Earth-to-Space QKD links, and (eventually) quantum repeaters.

13.10.5 Will Key Generation Rate Be High Enough to Support OTP?

Once QKD has achieved the establishment of matching private cryptographic keys across an insecure channel, those keys should, ideally, be used within an unconditionally secure classical cryptosystem such as a one time pad (OTP). However, the OTP is a voracious consumer of key material, requiring one key bit for each message bit. The current quantum key distribution rates are too low compared with the data volume of traffic to make this acceptable. Therefore, in some modern implementations of quantum cryptography, the keys created and distributed using quantum effects are used within a less secure classical cryptosystem such as AES. This weakens the overall security of the system, but also places lower demands on the amount of key material needed. Depending on one's degree of paranoia, one can vary the rate at which the key material is refreshed sufficient to achieve any desired level of practical security, depending on the technological capabilities you assume your adversary to possess. In practice, this is often good enough, even if absolute security is lost.

13.10.6 Will Security Be the Dominant Concern?

Finally we should always remember that secure key distribution is not the whole story in making an end-to-end secure cryptosystem. We still have to keep those keys secret at all times. Unfortunately, humans and trusted insiders are susceptible to blackmail, bribery, and corruption. Thus, if the key material is stored somewhere, and humans have access to it, there is a risk of improper disclosure.

Worse still, perhaps, the greatest *economic* threat to networks is not loss of confidential information but rather “denial-of-service” type attacks, wherein a hacker merely seeks to deprive parties of the ability to communicate. These can often cause more harm than the direct loss of communications data via eavesdropping and interception per se. Quantum cryptography does nothing to remedy denial-of-service attacks.

Despite these obstacles, the outlook for quantum cryptography remains bright. It is a quantum technology that offers a unique capability that cannot be provided by classical means, and appears to fulfill a proper niche in the space of secure communications, especially those whose content has to remain secret indefinitely.

13.11 Summary

The security of current public key cryptosystems rests upon unproven, but widely believed, mathematical assumptions about the difficulty of solving certain problems, such as factoring composite integers (for RSA cryptosystems) or computing discrete

logarithms (for elliptic curve cryptosystems). Shor's algorithm proves that it is possible, in principle, for quantum computers to factor composite integers and compute discrete logarithms (and hence break all known forms of public key cryptosystems) in polynomial time. Consequently, the current public key infrastructure for secure communications will become vulnerable to attack as soon as sufficiently powerful quantum computers are built. At such time the integrity of our secure communications infrastructure will be compromised. For routine non-sensitive communications this may not cause most people much concern. However, today, there are many types of communications passing through networks, such as diplomatic and military messages, financial transactions, medical records, and corporate data, that are of a highly sensitive nature. Moreover, such data may not only be sensitive at the time it is transmitted, but could remain sensitive for decades to come. Therefore, there is a need for a new secure communications infrastructure that will remain invulnerable to attack even if hackers and eavesdroppers have access to quantum computers, and which can ensure perpetual security of encrypted information even if it is intercepted. Fortunately, although quantum mechanics undermines the security of the classical public key infrastructure, so too does it offer a route to building a much stronger secure communications system based on what is known as *quantum cryptography*.

Quantum cryptography allows two parties, traditionally called Alice and Bob, to establish matching private cryptographic keys across an insecure communications channel by a process of quantum key distribution (QKD) followed by (classical) error reconciliation and privacy amplification. In QKD no particular key is intended initially. Rather, the key emerges by Alice and Bob following one of the QKD protocols. Regardless of which variant is used, QKD always involves encoding a sequence of random bits in a corresponding sequence of quantum states that are transmitted along the insecure communications channel. The encoding is such that if any eavesdropper (Eve) attempts to read the quantum states in transit, her actions are guaranteed to perturb the correlation statistics Alice and Bob would expect to see from following the QKD protocol in the absence of eavesdropping. If there is no evidence of any eavesdropping having taken place, the random cryptographic key Alice and Bob establish is deemed to be secure, and can be used in a subsequent unconditionally secure cryptosystem such as a OTP or a strong, but not unconditionally secure, cryptosystem such as AES.

Whereas the security of most classical cryptosystems relies upon the difficulty of solving certain mathematical problems such as factoring integers or computing discrete logarithms, the security of QKD rests upon quantum physical laws that cannot be circumvented no matter how mathematically gifted, algorithmically sophisticated, or computationally powerful an adversary might be. These laws include Heisenberg's Uncertainty Principle (see Sect. 12.1), and the No-Cloning Theorem (see Sect. 11.6.2). They are a fundamental aspect of Nature, confirmed to an incredibly high precision experimentally, and are impossible to circumvent.

An excellent in depth review of quantum cryptography can be found in the article "Quantum Cryptography" written by Nicolas Gisin, Grégoire Ribordy, Wolfgang Tittel, and Hugo Zbinden [201].

13.12 Exercises

13.1 The polarization of a photon can be expressed in several different bases. In particular, photons can be polarized “rectilinearly”, “diagonally”, or “circularly”. If a “rectilinear” polarization measurement device is used, a photon can be observed in either the vertical state, $|\downarrow\rangle$, or the horizontal state, $|\leftrightarrow\rangle$. Likewise, if a “circular” polarization measurement device is used, a photon can be observed in either the left circular state, $|\circlearrowleft\rangle$, or the right circular state, $|\circlearrowright\rangle$. These two bases are related via:

$$|\circlearrowleft\rangle = \frac{1}{\sqrt{2}}(|\downarrow\rangle - i|\leftrightarrow\rangle)$$

$$|\circlearrowright\rangle = \frac{1}{\sqrt{2}}(|\downarrow\rangle + i|\leftrightarrow\rangle)$$

where $i = \sqrt{-1}$.

- (a) If you prepare a photon in the left circular state, $|\circlearrowleft\rangle$, and immediately measure its polarization using a “rectilinear” polarization measuring device, what is the probability of finding the photon to be in state $|\leftrightarrow\rangle$?
- (b) What state will the photon be in immediately after making such a measurement and obtaining the result $|\leftrightarrow\rangle$?
- (c) If you prepare a photon in the state $|\psi\rangle = \frac{1}{2}|\downarrow\rangle - i\frac{\sqrt{3}}{2}|\leftrightarrow\rangle$ and immediately measure its circular polarization, what is the probability of finding the photon in state $|\circlearrowright\rangle$?

13.2 There are many possible ways to construct a QKD protocol. Alice and Bob come up with the following scheme:

1. Alice and Bob agree that if Alice wants to send Bob a 0 she sends him a $+45^\circ$ diagonally polarized photon, $|\nearrow\rangle$, and if she wants to send him a 1 she sends a right circularly polarized photon, $|\circlearrowright\rangle$.
2. Alice picks a random sequence of bits.
3. For each bit in her sequence, Alice transmits a single photon of the appropriate kind to Bob.
4. Bob tests the polarization of each arriving photon in one of two ways: he either tests for -45° polarization, $|\nwarrow\rangle$, to reveal 1's, or for left circular polarization, $|\circlearrowleft\rangle$, to reveal 0's. Note that Bob's measurement basis differs from Alice's encoding basis.

Answer the following questions:

- (a) Conceptually, what QKD protocol are Alice and Bob implementing? Justify your answer.
- (b) With what probability does Bob measure a 0 if he tests for left circularly polarized photons when Alice sends Bob a 0?

- (c) With what probability does Bob measure a 1 if he tests for left circularly polarized photons when Alice sends Bob a 1?
- (d) With what probability does Bob measure a 0 if he tests for -45° polarized photons when Alice sends Bob a 0?
- (e) With what probability does Bob measure a 1 if he tests for -45° polarized photons when Alice sends Bob a 1?
- (f) Assuming no losses or errors during transmission, what fraction of the bits will Alice and Bob come to know jointly?
- (g) Suppose an eavesdropper, Eve, taps the channel between Alice and Bob and tests each bit using the same protocol as followed by Bob. She then records the answer, re-encodes the bit value she measured as a polarized photon according to the same convention as used by Alice, and transmits that fresh polarized photon to Bob. What fraction of the bits that Alice sent to Bob will Alice, Eve, and Bob come to know jointly?

13.3 At their core, the 1-qubit BB84 protocol (which does not use entanglement) and the 2-qubit Ekert91 protocol (which does) are surprisingly similar. To see this,

- (a) Verify that the following identity holds for arbitrary 1-qubit gates U_1 and U_2 :

$$(U_1 \otimes U_2)|\beta_{0A0B}\rangle = \mathbb{1} \otimes (U_2 \cdot U_1^t)|\beta_{0A0B}\rangle \quad (13.14)$$

where $|\beta_{0A0B}\rangle = \frac{1}{\sqrt{2}}(|0_A0_B\rangle + |1_A1_B\rangle)$ and t denotes the transpose of the matrix.

- (b) Explain how this identity can be used to find a connection between the BB84 protocol and the Ekert91 protocol.

13.4 Let the rectilinear, diagonal, and circular polarization states of a photon be defined as follows:

- Left circular polarization, $|\circlearrowleft\rangle$
- Right circular polarization, $|\circlearrowright\rangle$
- Horizontal polarization, $|\leftrightarrow\rangle = \frac{i}{\sqrt{2}}(|\circlearrowleft\rangle - |\circlearrowright\rangle)$
- Vertical polarization, $|\Downarrow\rangle = \frac{1}{\sqrt{2}}(|\circlearrowleft\rangle + |\circlearrowright\rangle)$
- Diagonal $+45^\circ$ polarization, $|\nearrow\rangle = \frac{1+i}{2}|\circlearrowleft\rangle + \frac{1-i}{2}|\circlearrowright\rangle$
- Diagonal -45° polarization, $|\nwarrow\rangle = \frac{1-i}{2}|\circlearrowleft\rangle + \frac{1+i}{2}|\circlearrowright\rangle$

Determine whether each of the following states is entangled or unentangled:

- (a) $\frac{1}{\sqrt{2}}(|\circlearrowleft\rangle|\circlearrowleft\rangle + |\circlearrowright\rangle|\circlearrowright\rangle)$
- (b) $\frac{1}{\sqrt{2}}(|\leftrightarrow\rangle|\circlearrowleft\rangle + |\Downarrow\rangle|\circlearrowright\rangle)$
- (c) $\frac{1}{\sqrt{2}}(|\nearrow\rangle|\circlearrowleft\rangle + |\circlearrowright\rangle|\nwarrow\rangle)$
- (d) $\frac{1}{\sqrt{2}}(|\leftrightarrow\rangle|\nearrow\rangle + |\Downarrow\rangle|\nwarrow\rangle)$

13.5 The Pauli operators are defined as in (2.6). Are the following claimed equalities true or false?

- (a) $X \cdot Y = iZ$
- (b) $Y \cdot Z = iX$
- (c) $Z \cdot X = iY$
- (d) $X \cdot Y \cdot Z = Y \cdot Z \cdot X = Z \cdot X \cdot Y = i\mathbb{1}$
- (e) $X \cdot Z \cdot X = Y \cdot Z \cdot Y = -Z$
- (f) $Y \cdot X \cdot Y = Z \cdot X \cdot Z = -X$
- (g) $Z \cdot Y \cdot Z = X \cdot Y \cdot X = -Y$

13.6 Let $\mathbb{1}$, X , Y , and Z be the Pauli matrices defined in (2.6). A matrix, H is hermitian if and only if it equals its own conjugate transpose, i.e., $H = H^\dagger$. Likewise a matrix, U , is unitary if and only if its inverse equals its conjugate transpose, i.e., $U \cdot U^\dagger = \mathbb{1}$ where $\mathbb{1}$ is the identity matrix. Prove that the Pauli matrices, $\mathbb{1}$, X , Y , and Z , are both unitary and hermitian. Classify the following matrices according to whether they are hermitian and unitary, hermitian but not unitary, non-hermitian but unitary, or neither hermitian nor unitary:

- (a) $\frac{1}{2}(X + Y)$
- (b) $\frac{1}{\sqrt{2}}(Y + Z)$
- (c) $\frac{1}{\sqrt{2}}(X + iZ)$
- (d) $\frac{1}{\sqrt{3}}(X + Y + Z)$
- (e) $\frac{1}{2}(X + Y + Z + \mathbb{1})$
- (f) $\frac{1}{2}(X + Y + Z + i\mathbb{1})$

13.7 Let $\mathbb{1}$, X , Y , and Z be the Pauli matrices defined in (2.6). Show that the matrix $\frac{1}{2}(X + Y + Z + i\mathbb{1})$ can be factored in terms of $\sqrt{\text{NOT}}$, a single $R_z(\alpha)$ -rotation gate, and a single $Ph(\beta)$ phase gate. Determine the required values for the angles α and β .

13.8 Compute the following matrix exponentials:

- (a) $e^{i\alpha X}$
- (b) $e^{\alpha X}$
- (c) $e^{i(\alpha X + \beta Y)}$
- (d) $e^{i(\alpha X + \beta Y + \gamma Z)}$

13.9 Let A , B , and C be hermitian operators defined by:

$$A = \begin{pmatrix} -0.722 & 0.148 + 0.569i \\ 0.148 - 0.569i & -1.674 \end{pmatrix}$$

$$B = \begin{pmatrix} -2.548 & 0.3 - 0.692i \\ 0.3 + 0.692i & -1.787 \end{pmatrix}$$

$$C = \begin{pmatrix} 0.894 & -0.805 - 0.351i \\ -0.805 + 0.351i & 1.585 \end{pmatrix}$$

Verify that A , B and C are Hermitian and compute the following commutators and anti-commutators:

- (a) $[A, B]$
- (b) $[A, B + C]$
- (c) $[A, [B, C]]$
- (d) $\{A, B\} + [A, B]$

Part IV

Towards Practical Quantum Computers

Chapter 14

Quantum Error Correction

“I wish to God these calculations had been executed by steam!”

– Charles Babbage¹

The descriptions of quantum algorithms and quantum information processing protocols given in the foregoing chapters all assume a correct design, precise implementation, and perfect operation of our quantum computing device. But real quantum hardware, and real quantum computations run on it, are unlikely to be manufactured exactly to their specifications, and unlikely to perform flawlessly. Components of real quantum computers can only be manufactured and assembled to within some finite tolerances. Pulses can only be shaped and timed to within certain limits. Voltages, currents, fluxes, and inter-qubit couplings cannot be turned on and off instantaneously, etc. Moreover, the fundamental paradox of quantum computation is that at one instant we desire our qubits to be isolated perfectly from their environment, but at another, we want them to interact strongly with some “external” measuring apparatus. Turning such environmental interactions wholly on and off at will is challenging. Thus, real quantum computers will be beset with errors causing their computations to go awry.

A similar situation holds in classical computing, of course. But in that case we can identify and correct bit-errors using various classical error-correction techniques. We are helped profoundly in this regard by the ability classical physics gives us to look at the instantaneous state of a classical computation, assess its correctness, and then make the necessary adjustments. In the quantum realm we do not have this luxury, because we cannot read the state of a quantum memory register in the midst of a quantum computation without necessarily, and irreversibly, perturbing the future course of the computation. Thus it is not at all obvious, *a priori*, whether the techniques developed for correcting errors in classical computers are useful for correcting errors in quantum computers. In fact, shortly after Shor’s algorithm was first published several highly respected physicists expressed skepticism about the feasibility of an error-correction method for quantum computers [227, 300, 301, 502].

¹Source: Computer History Museum, <http://www.computerhistory.org/babbage/history/>.

14.1 How Errors Arise in Quantum Computing

In the idealized models of quantum computers that we studied in Chaps. 1–3, the qubits representing the computational state of the computer were assumed to be perfectly isolated from their environment. In other words, from the moment the quantum computer is prepared in some initial state to start the computation off, to the moment it is measured to extract an answer, the logical qubits of ideal quantum computers are supposed to evolve unitarily in accordance with Schrödinger’s equation. Unfortunately, such an idealization is unattainable. Any real quantum system couples to its environment over time. In the process, information leaks out of the logical state of the qubits in the quantum memory register. If you did not model the effect of the environment explicitly, it would appear as if the logical qubits were no longer evolving unitarily in accordance with Schrödinger’s equation. Indeed, this coupling between a quantum system and its environment, and the resulting loss of coherence, is what prevents quantum effects from being evident at the macroscopic level [202].

Even with our best efforts keeping a quantum memory register isolated from its environment is difficult. For one thing, a quantum memory register has to be built out of *something* so there must be some supporting infrastructure, or “scaffolding,” in the vicinity of the computationally active qubits. There is, therefore, a chance that the particles within the scaffolding will couple to the computational elements. In addition, there can be a coupling between the memory register and an ambient thermal heat bath. Also, incoming stray particles such as cosmic rays or gas molecules can interact with the memory register. In fact, there are many physical processes that can perturb the state of a quantum memory register. Broadly speaking, these physical processes fall under the headings of dissipation and decoherence.

14.1.1 Dissipation-Induced Bit Flip Errors

Dissipation is a process by which a qubit loses energy to its environment. Thus, for example, if an excited state is used to represent a $|1\rangle$ and a lower energy state is used to represent a $|0\rangle$, a qubit might transition, spontaneously, from the $|1\rangle$ state to the $|0\rangle$ state emitting a photon in the process. In computational terms, a bit in the quantum memory register would have “flipped” spontaneously.

Dissipation causes a bit to flip and this operation is described by the action of the Pauli X matrix. We assume that the qubit starts off in an arbitrary superposed state given by

$$|\psi\rangle = a|0\rangle + b|1\rangle \quad (14.1)$$

such that $|a|^2 + |b|^2 = 1$.

The affect of σ_x on the state of a qubit is:

$$\sigma_x(a|0\rangle + b|1\rangle) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} b \\ a \end{pmatrix} = b|0\rangle + a|1\rangle \quad (14.2)$$

This time the action of the operator has caused the bits to flip. That is, σ_x causes the transformation $a|0\rangle + b|1\rangle \rightarrow b|0\rangle + a|1\rangle$. So let us call such an operation a “bit flip error.”

14.1.2 Decoherence-Induced Phase Shift Errors

Decoherence, on the other hand, is more insidious. Rather than an overt bit flip, stray interactions between the qubits and the environment cause the quantum memory register and the environment to become entangled with one another. As a result, the initially pure state of our ideal quantum memory register becomes progressively more mixed over time. This mixing alters the relative phases of the computational basis eigenstates of the memory register. As a result, the interference effects, needed in any true quantum computation, become distorted and the quantum computation no longer proceeds as it should.

An overly simplified, but intuitive, model for the impact of such decoherence on a quantum memory register is as follows. Suppose that initially, i.e., at a time $t = 0$, a single qubit in a quantum memory register starts out in the pure state $|\psi\rangle = a|0\rangle + b|1\rangle$. As a density matrix such a state may also be written as:

$$\rho(0) = |\psi\rangle\langle\psi| = \begin{pmatrix} a \\ b \end{pmatrix} \cdot (a^* & b^*) = \begin{pmatrix} |a|^2 & ab^* \\ a^*b & |b|^2 \end{pmatrix} \quad (14.3)$$

where the asterisk denotes taking the complex conjugate. After merely “storing” such a qubit in a realistic, i.e., weakly noisy, environment for a time t , its density matrix will become:

$$\rho(t) = \begin{pmatrix} |a|^2 & e^{-t/\tau}ab^* \\ e^{-t/\tau}a^*b & |b|^2 \end{pmatrix} \quad (14.4)$$

where τ , called the “decoherence time,” sets the characteristic time-scale of the decoherence process, i.e., the time it takes for the off-diagonal elements of ρ to decay appreciably. In the long time limit, $\tau \rightarrow \infty$, the density matrix becomes a mixture of the two possible measurement outcomes for this qubit, namely:

$$\rho(\infty) = \begin{pmatrix} |a|^2 & 0 \\ 0 & |b|^2 \end{pmatrix} \quad (14.5)$$

It is as if the environment has “measured” the qubit.

Similarly, applying σ_z to the qubit results in the following transformation:

$$\sigma_z(a|0\rangle + b|1\rangle) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ -b \end{pmatrix} = a|0\rangle - b|1\rangle \quad (14.6)$$

That is, σ_z causes the “correct” state to evolve according to the rule $a|0\rangle + b|1\rangle \rightarrow a|0\rangle - b|1\rangle$, which has changed the *phase* of the qubit. Consequently, we call such an operation a “phase shift error.”

The action of the identity matrix on a state is to leave the state unchanged. So that must represent the “no error” possibility.

That only leaves us to consider what happens when we apply σ_y to the state of the qubit:

$$\sigma_y(a|0\rangle + b|1\rangle) = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} -ib \\ ia \end{pmatrix} = -ib|0\rangle + ia|1\rangle \quad (14.7)$$

This operation corresponds to *both* a phase shift and a bit flip. That is, σ_y causes the transformation $a|0\rangle + b|1\rangle \rightarrow -ib|0\rangle + ia|1\rangle$. Thus, any error in a single qubit can be described by the action of a linear combination of the operators σ_x , σ_y , σ_z , and $\mathbb{1}$ (the identity operator).

14.1.3 Natural Decoherence Times of Physical Systems

Usually, decoherence occurs on a faster timescale than dissipation. The time it takes a memory register to decohere depends, principally, upon what kind of quantum systems it is made from, the size of the register, the temperature of the thermal environment, and the rate of collisions with ambient gas molecules.

A crude estimate of decoherence times in various settings can be obtained from the Heisenberg Uncertainty Principle, in energy and time,

$$\Delta t \approx \frac{\hbar}{\Delta E} = \frac{\hbar}{k_B T} \quad (14.8)$$

where k_B is Boltzmann’s constant (approximately 1.38×10^{-23} Joules K $^{-1}$) and T is the absolute temperature of the environment. In this estimate we have taken the uncertainty in the energy to be of the order of the energy of a typical particle at the ambient temperature. At room temperature, this gives a typical decoherence time of about 10^{-14} seconds. At lower temperatures, systems take longer to decohere. For example, at the temperature of liquid helium, it takes about 100 times as long for a system to decohere as it does at room temperature. Consequently, the simplest way to try to combat decoherence is to operate the computer at a lower temperature. Table 14.1 summarizes some characteristic decoherence times, under various physical scenarios. These estimates were derived using a more sophisticated analysis [257].

Once we have chilled our quantum computer and sealed it in as good a vacuum as we can, what else can we do to slow down decoherence? Well, we could try building the quantum memory register out of different types of quantum systems. Certain quantum systems are much more resilient to decoherence than others. David Di-Vincenzo has collected statistics on the intrinsic decoherence properties of various materials [146]. The data are shown in Table 14.2. They reveal that trapped ions, for example, can potentially sustain a large number of computational steps before the succumb to decoherence. Step counts reported in Table 14.2 suggest that it might

Table 14.1 Approximate decoherence times (in seconds) for various sized systems in different thermal and gaseous environments [257]

System size (cm)	Cosmic background radiation	Room temp. (300 K)	Sunlight	Vacuum	Air
10^{-3}	10^{-7}	10^{-14}	10^{-16}	10^{-18}	10^{-35}
10^{-5}	10^{15}	10^{-3}	10^{-8}	10^{-10}	10^{-23}
10^{-6}	10^{24}	10^5	10^{-2}	10^{-6}	10^{-19}

Table 14.2 The maximum number of computational steps that can be accomplished without losing coherence for various quantum systems

Quantum system	Time per gate operation	Coherence time	Max. no. of coherent steps
Mössbauer nucleus	10^{-19}	10^{-10}	10^9
GaAs electrons	10^{-13}	10^{-10}	10^3
Gold electrons	10^{-14}	10^{-8}	10^6
Trapped indium ions	10^{-14}	10^{-1}	10^{13}
Optical microcavity	10^{-14}	10^{-5}	10^9
Electron spin	10^{-7}	10^{-3}	10^4
Electron quantum dot	10^{-6}	10^{-3}	10^3
Nuclear spin	10^{-3}	10^4	10^7

be possible to build a quantum memory register that can support a significant number of computational steps. Nevertheless, decoherence looks like it will preclude quantum computation beyond a certain number of steps. This poses a severe problem for anyone wanting to build a universal quantum computer. Ideally, we would like a quantum computer that could, in principle, maintain coherent quantum computations indefinitely. Thus, if we can prevent decoherence, we need to think about ways of undoing its affects. Thus there needs to be a way of doing quantum error correction. But to understand how to correct errors, we need to understand how errors will perturb the quantum states we wish to protect. So what we need next is a mathematical model of the effects of errors on quantum computations.

14.1.4 What Makes Quantum Error Correction so Hard?

With classical computers, it is possible to measure the state of the physical system used to encode a bit without disrupting the bit. Thus, if a voltage were used to represent a classical bit, you could, in principle, detect a slight drop from its nominal value and then give the voltage a nudge to restore it to its correct level.

Secondly, once a full bit-error has occurred, the nature of the error is far more limited in the classical domain than the quantum domain. In particular, the principal types of errors that can afflict a classical bit are either a bit *flip*, i.e., $0 \rightarrow 1$ or

$1 \rightarrow 0$, or, especially in the case of communication channels, the *loss* of a bit or the *insertion* of spurious bit. These types of errors are discrete and flagrant. There is no subtle “drift” in a bit value—it is either correct or flipped, and present or absent. Contrast this with the kinds of errors that can afflict qubits.

Qubits, however, do not have to be in states that are wholly $|0\rangle$ or wholly $|1\rangle$, but can be in superpositions of $|0\rangle$ and $|1\rangle$, e.g., $\alpha|0\rangle + \beta|1\rangle$, where the values of the amplitudes span a continuum of values. Thus, qubit states can “drift” off their intended values rather than suffer only gross errors (as do classical bits). This makes the errors that can afflict a qubit potentially far more subtle and insidious than the errors that can afflict a classical bit. Thus, in addition to a qubit bit flip $\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|1\rangle + \beta|0\rangle$, it is also possible to have qubit phase shifts $\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|0\rangle + \beta e^{i\phi}|1\rangle$ in which, even though the amplitudes remain the same magnitudes, errors can creep into the relative phase between the $|0\rangle$ and $|1\rangle$ components causing error states such as $\alpha|0\rangle + \beta e^{i\phi}|1\rangle$. Such corruption of the relative phase between the $|0\rangle$ and $|1\rangle$ components can mess up subsequent interference effects that all quantum algorithms rely upon. This particular, failure mode does not exist in the case of a classical computer.

Thirdly, in classical computing, we can make copies of bits we want to protect, replicate computations done on them, and use majority votes of the results to help eliminate errors. This ability to have redundant information is a great asset in error-correcting classical information. In quantum computing, however, the quantum no-cloning theorem precludes the possibility of copying an unknown quantum state. This makes it much more difficult to see how one could exploit redundancy in quantum computations for error correction purposes.

The aforementioned differences between classical and quantum information from the perspective of its intrinsic ability to be error-corrected are summarized in Table 14.3.

Table 14.3 Intrinsic differences between classical information and quantum information that make quantum error correction more difficult than classical error correction

Feature	Classical	Quantum
Information	Discrete encoding (0 or 1)	Continuous encoding ($\alpha 0\rangle + \beta 1\rangle$)
Bit Errors	$0 \rightleftharpoons 1$	$\alpha 0\rangle + \beta 1\rangle \rightarrow \alpha 1\rangle + \beta 0\rangle$
Phase Errors	Phase errors cannot occur for classical bits	$\alpha 0\rangle + \beta 1\rangle \rightarrow \alpha 0\rangle + \beta e^{i\phi} 1\rangle$
Compound Errors	Compound bit and phase errors cannot occur for classical bits	$\alpha 0\rangle + \beta 1\rangle \rightarrow \alpha 1\rangle + \beta e^{i\phi} 0\rangle$
Redundancy	Can be used	Cannot be used once the quantum computation is underway because the no-cloning theorem precludes copying an unknown quantum state
Monitoring	Can read memory register during computation to ascertain nature of error	Cannot read memory register during computation to ascertain nature of error

Fortunately, we now know that there is a solution to our dilemma. The answer lies in *quantum* error correction. The trick, as John Preskill of Caltech likes to say, is “to use entanglement to fight entanglement”. That is, by creating a specially designed entanglement between a quantum state we want to protect and that of other qubits, we can recognize when our protected state has gone bad and fix it, without damaging the delicate quantum correlations within our protected state. In this chapter we will review some approaches to quantum error correction and explain how entanglement is both the problem, and solution.

14.2 Quantum Error Reduction by Symmetrization

Classical computers can be made more reliable through the use of redundancy. Instead of a single computer being used to perform a given computation, several computers are used to perform the same computation simultaneously. If the computers are all running the same deterministic algorithm, they should all produce identical results at each stage of the computation. However, if an error occurs in one of the computers, its computational state will begin to diverge from that of the others. If you periodically poll all the computers and reset their computational states to the majority opinion, you will typically be able to correct errors that arose in a few of the computers since the last poll was taken. This type of majority voting scheme is currently used in the Space Shuttle to improve the reliability of the on-board decision making.

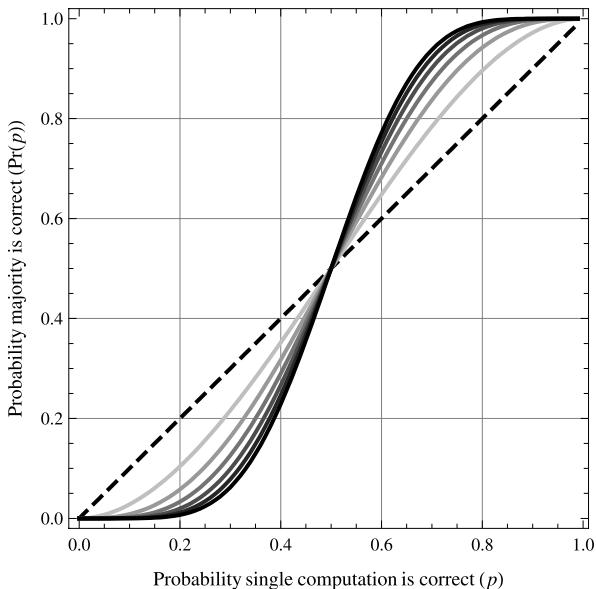
For majority voting to be effective, however, a number of assumptions must hold. First, the individual chances of any one computer obtaining the “correct” result must be greater than 50%. If this were not true then the majority opinion is more likely to be wrong than it is to be right. Secondly, the replicated computations must be independent of one another so that the errors incurred by the different computers are uncorrelated. This can be difficult to achieve in practice if all the computers use the same type of hardware and run the same program. Finally, replicating the computation an odd number of times (i.e., $2N - 1$) guarantees a majority opinion always exists. The more replicated computations you use, the better your chances of fixing potential errors. In fact, if there are $2N - 1$ computers (for $N = 1, 2, 3, \dots$) and the individual probability of each computer obtaining the correct answer is $p > 0.5$ then the probability that the majority opinion is correct is given by:

$$\Pr(\text{Majority Correct}) = \sum_{i=N}^{2N-1} \binom{2N-1}{i} p^i (1-p)^{2N-1-i} \quad (14.9)$$

Although unsophisticated, this scheme is actually used today on the Space Shuttle and Boeing 777.

Figure 14.1 shows how the probability of the majority vote being correct increases as the probability of success of the individual computations increases for various numbers of replicated computations.

Fig. 14.1 Probability that the majority vote is correct based on the probability of a single independent computation being correct. The *dashed line* is the case of a single computation without replication; the *lightest curve* is for the same computation repeated on three computers; and the *darkest curve* is for the same computation repeated on 13 computers. When the individual success probability exceeds 0.5 it pays to repeat computations and adopt the majority decision



Unfortunately, in quantum computation we cannot use such a majority voting scheme. This is because at the intermediate stages of typical quantum computations the quantum memory registers will be in superpositions of possible bit string configurations weighted non-uniformly by different probability amplitudes. If we were to readout the memory register during the course of the quantum computation we could project the state of the register into an eigenstate of the memory register thereby destroying the delicate superposition and in fact de-railing the quantum computation. So if we attempted to use naive majority voting within quantum computation, we would unfortunately destroy the computation.

14.2.1 The Symmetrization Trick

There is, however, a more cunning way to use something akin to majority voting within quantum computation. This is called the method of error reduction via symmetrization [35]. The idea is that although we have no idea whatsoever what the instantaneous state of some quantum computation might be, we do know that if we had R replicas of the same quantum computation, that the *joint* state of all R quantum computations would be the tensor product of the individual quantum computations, i.e.,

$$|\Psi(t)\rangle_{\text{ideal}} = |\psi(t)\rangle \otimes |\psi(t)\rangle \otimes \cdots \otimes |\psi(t)\rangle \quad (14.10)$$

This is because, so long as no observations are made, the quantum evolution of an isolated quantum system is governed by Schrödinger's equation, which is a de-

terministic differential equation. Hence, if no errors afflicted any of the quantum computations then the joint state ought to have a tensor product structure.

In reality, however, each quantum computation might experience some error at random and uncorrelated from the errors afflicting the sister quantum computations. If this happens, the actual joint state of the R quantum computations would be something like:

$$|\Psi(t)\rangle_{\text{actual}} = |\psi_1(t)\rangle \otimes |\psi_2(t)\rangle \otimes \cdots \otimes |\psi_R(t)\rangle \quad (14.11)$$

Quantum error correction by symmetrization works by intermittently projecting the joint state of the R quantum computers into the symmetric subspace \mathcal{SYM} . The correct part of the quantum computation is always guaranteed to lie within \mathcal{SYM} , so by projecting the joint state into \mathcal{SYM} we only knock out parts of the joint state that must be buggy, and thereby boost the proportion of the correct state within \mathcal{SYM} . Unfortunately, there are other symmetric states that can lie within \mathcal{SYM} too which are not part of the true state. Nevertheless, provided we project into \mathcal{SYM} often enough and provided we use enough replicas, R , of our computation these other types of errors can be suppressed to any desired level.

Quantum Error Reduction via Symmetrization

1. Initialize R identical independent quantum computers to be in the same starting state running the same quantum algorithm. If there are no errors then, at any instant, the *joint* state of these R quantum computers would be a state of the form $|\psi\rangle|\psi\rangle\cdots|\psi\rangle$, which is invariant under any permutation of the computers. However, due to independent small errors, the joint state will actually be of the form $|\psi_1\rangle|\psi_2\rangle\cdots|\psi_R\rangle$ where the individual component states (corresponding to the R independent quantum computations) will be slightly different from one another.
2. To suppress the accumulate errors, initialize $O(\log_2 R!) \approx O(R \log_2 R)$ ancillae in state $|0\rangle$.
3. Place the ancillae in an equally weighted superposition of the integers (i.e., bit strings) in the range 0 to $R! - 1$, i.e., perform the transformation:

$$\mathcal{U} |0\rangle \rightarrow \frac{1}{\sqrt{R!}} \sum_{i=0}^{R!-1} |i\rangle \quad (14.12)$$

4. Apply the i -th permutation to the states of the R individual quantum computers conditioned on the value $|i\rangle$ stored in the ancillae. That is, apply the conditional transformation:

$$|i\rangle|\psi_1\rangle|\psi_2\rangle\cdots|\psi_R\rangle \rightarrow |i\rangle|\psi_{\sigma_i}(1)\rangle|\psi_{\sigma_i}(2)\rangle\cdots|\psi_{\sigma_i}(R)\rangle \quad (14.13)$$

thereby creating the entangled state:

$$\sum_i |i\rangle|\psi_{\sigma_i(1)}\rangle|\psi_{\sigma_i(2)}\rangle\cdots|\psi_{\sigma_i(R)}\rangle \quad (14.14)$$

5. Apply the inverse computation \mathcal{U}^{-1} to that applied in step 3 above. As the forward \mathcal{U} operation mapped $|0\rangle$ into equally weighted superposition of the $R!$ possible integers (representing the possible indices of the permutations of R objects), then the inverse operation does exactly the reverse. Thus the state we obtain can be written as:

$$\sum_i |i\rangle |E_i\rangle \quad (14.15)$$

in which the $|E_0\rangle$ component (i.e., the state of the rest of the register when the ancillae is in state $|0\rangle$) represents the desired (i.e., symmetrized) state, and the other components are error states.

6. Measure the ancillae qubits in the computational basis. If they are all found to be in state $|0\rangle$ then $|\Psi\rangle$ has been successfully projected into the symmetric subspace \mathcal{SYM} .

14.2.2 Quantum Circuit for Symmetrization

Projection into the symmetric subspace can be accomplished using a quantum circuit like that shown in Fig. 14.2, which is specialized to the case of three replicated computations. The key insight is realize that you can build up permutations of quantum states *recursively*. Specifically, consider a set of $k+1$ elements e_1, e_2, \dots, e_{k+1} . How can we construct all permutations of this set? Well suppose we already have a permutation, $e_{\sigma(1)}, e_{\sigma(2)}, \dots, e_{\sigma(k)}$ of the first k elements, e_1, e_2, \dots, e_k of the

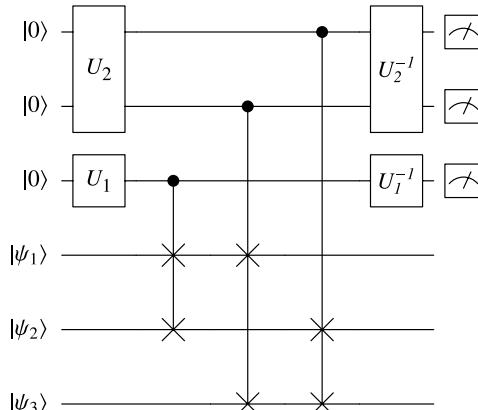


Fig. 14.2 Quantum circuit for error correction via symmetrization. In this example, we symmetrize the state of three replicas of a 1-qubit quantum computation. Provided that, when measured, the ancillae are all found in state $|0\rangle$, the overlap between the joint correct state and the joint symmetrized state, $\langle \Psi_{\text{correct}} | U_{\mathcal{SYM}} | \Psi_{\text{buggy}} \rangle$, will be higher than the overlap between the joint correct state and the joint unsymmetrized state, $\langle \Psi_{\text{correct}} | \Psi_{\text{buggy}} \rangle$

set. We can then join e_{k+1} to the end of this permutation, creating the permutation $e_{\sigma(1)}, e_{\sigma(2)}, \dots, e_{\sigma(k)}, e_{k+1}$. The remaining permutations can be constructed by systematically swapping e_{k+1} with each of the $e_{\sigma(i)}$ in turn. This suggests the structure of a quantum circuit sufficient to generate all possible permutations of $k + 1$ quantum states. These represent $k + 1$ independent realizations of some quantum computation.

In other words, once we have a symmetrized version of the state $|\psi_1\rangle|\psi_2\rangle\dots|\psi_k\rangle$, we can easily symmetrize the state $|\psi_1\rangle|\psi_2\rangle\dots|\psi_k\rangle|\psi_{k+1}\rangle$ by adjoining state $|\psi_{k+1}\rangle$ and applying a sequence of controlled SWAP operations. As you may recall from Chap. 2, controlled-SWAP is synonymous with a FREDKIN gate.

$$U_k = \left[\bigcirc_{j=k-1}^1 \mathbb{1}_{2^{j-1}} \otimes \frac{1}{\sqrt{k-j+1}} \right. \\ \times \begin{pmatrix} \sqrt{-j+k+1} & 0 & 0 & 0 \\ 0 & 1 & \sqrt{k-j} & 0 \\ 0 & -\sqrt{k-j} & 1 & 0 \\ 0 & 0 & 0 & \sqrt{-j+k+1} \end{pmatrix} \otimes \mathbb{1}_{2^{k-(j+1)}} \left. \right] \\ \cdot \frac{1}{\sqrt{k+1}} \begin{pmatrix} 1 & -\sqrt{k} \\ \sqrt{k} & 1 \end{pmatrix} \otimes \mathbb{1}_{2^{k-1}} \quad (14.16)$$

14.2.3 Example: Quantum Error Reduction via Symmetrization

Suppose we have three replicas of the same quantum computation, such that the correct state should be:

$$|\Psi_{\text{correct}}\rangle = |\psi\rangle|\psi\rangle|\psi\rangle \quad (14.17)$$

where

$$|\psi\rangle = \frac{1}{2}|0\rangle - \frac{\sqrt{3}}{2}i|1\rangle \quad (14.18)$$

Imagine that the independent computations have each drifted slightly off the correct states $|\psi\rangle$ so that what we actually have is:

$$|\Psi_{\text{buggy}}\rangle = |\psi_1\rangle|\psi_2\rangle|\psi_3\rangle \quad (14.19)$$

where

$$|\psi_1\rangle = \left\| \frac{1}{2}|0\rangle - \frac{\sqrt{3.5}}{2}i|1\rangle \right\| \\ |\psi_2\rangle = \left\| \frac{1}{2.5}|0\rangle - \frac{\sqrt{3}}{2}i|1\rangle \right\| \\ |\psi_3\rangle = \left\| \frac{1}{1.5}|0\rangle - \frac{\sqrt{2.5}}{2}i|1\rangle \right\| \quad (14.20)$$

where the symbol $\| \|\$ indicates the re-normalized version of the state. We expressed the perturbed states as shown to make it easier to see that they are only slight adrift of their ideal values.

Thus the overall correct state is:

$$\begin{aligned} |\Psi_{\text{correct}}\rangle &= |\psi\rangle|\psi\rangle|\psi\rangle \\ &\approx \left(\begin{array}{c} 0.125|000\rangle - 0.216506i|001\rangle - 0.216506i|010\rangle - 0.375|011\rangle \\ 0.216506i|100\rangle - 0.375|101\rangle - 0.375|110\rangle + 0.649519i|111\rangle \end{array} \right) \end{aligned} \quad (14.21)$$

whereas the actual (buggy) state we have is:

$$\begin{aligned} |\Psi_{\text{buggy}}\rangle &= |\psi_1\rangle|\psi_2\rangle|\psi_3\rangle \\ &\approx \left(\begin{array}{c} 0.127427|000\rangle - 0.15111i|001\rangle - 0.275888i|010\rangle - 0.327163|011\rangle \\ 0.238395i|100\rangle - 0.282701|101\rangle - 0.51614|110\rangle + 0.612066i|111\rangle \end{array} \right) \end{aligned} \quad (14.22)$$

Hence, the overlap between the correct state and the buggy state is:

$$\langle\Psi_{\text{correct}}|\Psi_{\text{buggy}}\rangle \approx 0.979791 \quad (14.23)$$

Now let us see what happens when we attempt to re-symmetrize the buggy state. By how much does the error reduce? To construct the error symmetrization operator we need the following gates:

$$\begin{aligned} U_1 &= \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \\ U_2 &= \begin{pmatrix} \frac{1}{\sqrt{3}} & 0 & -\sqrt{\frac{2}{3}} & 0 \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ 0 & \sqrt{\frac{2}{3}} & 0 & \frac{1}{\sqrt{3}} \end{pmatrix} \end{aligned} \quad (14.24)$$

Then, the full error symmetrization operator is constructed from:

$$\begin{aligned} U_{\mathcal{SYM}_3} &= (U_2^{-1} \otimes U_1^{-1} \otimes \mathbb{1}_8) \cdot \text{FREDKIN}_{1,5,6;6} \cdot \text{FREDKIN}_{2,4,6;6} \\ &\quad \cdot \text{FREDKIN}_{3,4,5;6} \cdot (U_2 \otimes U_1 \otimes \mathbb{1}_8) \end{aligned} \quad (14.25)$$

where the subscript 3 on \mathcal{SYM} indicates the operator is specialized to symmetrizing a triple repetition of the quantum computation, and $\text{FREDKIN}_{i,j,k;\ell}$ means a Fredkin gate inserted in ℓ qubits with control on qubit i , and the SWAP it performs between qubits j and k .

So now let us re-symmetrize the buggy state. That is we compute:

$$U_{\mathcal{SYM}_3}|\Psi_{\text{buggy}}\rangle = 0.129651|000\rangle - 0.232026i|001\rangle - 0.246828i|010\rangle$$

$$\begin{aligned}
 & - 0.42901|011\rangle - 0.198151 i|100\rangle - 0.342299|101\rangle \\
 & - 0.374345|110\rangle + 0.622747 i|111\rangle
 \end{aligned} \tag{14.26}$$

Hence, the overlap between the correct state and the buggy state after re-symmetrization is:

$$\langle \Psi_{\text{correct}} | U_{\mathcal{SYM}_3} | \Psi_{\text{buggy}} \rangle \approx 0.996889 \tag{14.27}$$

which is higher than it was before re-symmetrization. Hence, error correction by symmetrization has succeeded in reducing the error, even without knowing what the error was!

This idea of coupling two systems, so that measuring the state of one system projects the state of the other into a specific subspace, can be used to perform error correction. This technique is most appropriate for correcting several qubits that are slightly wrong rather than correcting a single qubit that is terribly wrong [388].

Quantum error reduction by symmetrization is most suited to correcting small independent errors (such as random phase drifts rather than bit flips) and is more successful the more frequently it is repeated. However, certain error processes, such as spontaneous emission, can result in sudden *large* errors, such as bit flips. These kinds of errors require a different error-correction strategy based on the idea of error-correcting codes.

14.3 Principles of Quantum Error Correcting Codes (QECCs)

Classical error correcting codes are used routinely to immunize classical computations and communications from errors such as accidental bit flips. The key idea is to use classical *codewords*, i.e., carefully chosen bit strings, to encode each logical bit we want to protect, in such a manner that a subsequent error, or perhaps multiple errors, in a codeword can be detected and corrected. Quantum error correcting codes (QECCs) extend this basic idea to the quantum domain but require several modifications to allow the codes to handle quantum, rather than classical, information.

14.3.1 Classical Error Correcting Codes

The simplest classical error correcting code maps the logical bits 0 and 1 into a pair of carefully chosen bitstrings, i.e., *codewords*, chosen so as to be maximally distinguishable from one another. Once so encoded, if a bit-flip occurs within a codeword, causing it to become corrupted, the error can be readily identified and then corrected by replacing the corrupted codeword with the “closest” legal codeword to it. Typically, the distance metric used to assess “closeness” is the Hamming distance between bit strings. This is defined so that, if \mathbf{x} and \mathbf{y} are two bit strings, their Hamming distance is the number of places in which \mathbf{x} and \mathbf{y} differ.

Of course, far more sophisticated classical error-correcting codes can be devised by elaborating on this basic idea, e.g., by finding ways to encode tuples of logical bits (which one can think of as classical “symbols”) as longer tuples of physical bits (the codewords) such that multiple bit-flips, bit drops, or bit insertions, within the codewords are detectable and correctable to the closest legal symbols. NASA did much of the pioneering work in error-correcting codes, motivated by the needs of spacecraft to communicate reliably with Earth over exceedingly large distances and extremely noisy channels. But the field has now blossomed into a rich assortment of techniques that are used routinely in terrestrial telecommunications and data storage. Not surprisingly, the field of error-correcting codes has deep roots in Shannon information theory discussed in Chap. 11.

14.3.2 Issues Unique to Quantum Error Correcting Codes

Unfortunately, error correcting codes cannot be used in quite the same way in the quantum context as they are used in classical context. The problem is that, even if we have mapped the qubits into quantum codewords, we still cannot read a potentially corrupted quantum codeword *directly* at any intermediate step of a quantum computation in an attempt to detect an error. To do so, would cause the superposition to collapse in some unpredictable way, thereby erasing whatever remnants of correct information lay buried in the corrupted encoded state. In fact, such measurements would be likely to make the error worse rather than better. In the early years of quantum computing, this apparent prohibition on reading the corrupted encoded state to extract an error-syndrome led some researchers to speculate that quantum error-correcting codes could not exist [227, 300, 301, 502]. This cast severe doubt on the feasibility of quantum computers, because it seemed as though they would require absolute perfection in fabrication, initialization, operation, and readout—which are not likely, in practice, to be achievable. Thus, the apparent impossibility of quantum error-correcting codes seemed like a major obstacle to the development of quantum computing, because other quantum error correction schemes, such as error-correction via symmetrization, were insufficient to correct all the types of errors that were likely to arise in real quantum computing hardware.

The situation changed in 1995, however, when Peter Shor published the first account of a viable quantum error correcting code [456]. Shor’s idea was to encode each logical qubit whose state we wanted to protect within a specially crafted *entangled* state of several qubits. The encoding scheme was such that any error afflicting one of these entangled qubits thereafter could be identified by making measurements on a *subset* of the qubits to obtain what was called an “error syndrome”. Once the error syndrome was known, the error that afflicted the logical state we were trying to protect could be reversed by applying an appropriate sequence of unitary gates (i.e., error recovery operations) that were different depending on whichever error syndrome had been obtained. All subsequent quantum codes have followed this basic pattern.

In the following sections we describe the theory of quantum error correcting codes. We will start by specifying the error model as we have to know what kinds of errors can afflict our logical qubits in order to devise codes to detect and correct such errors. We then outline the properties any quantum error correcting code needs to possess to enable it to protect quantum information that is, by its very nature, unreadable without corruption. Finally, we will look at error diagnosis and recovery.

14.3.3 Modeling Errors in Terms of Error Operators

We can think of an “error” as change in the state of our logical qubit that is caused because it is not as well isolated from its environment as it is supposed to be. In this case, instead of the quantum mechanical evolution being the desired unitary evolution on the qubit alone, we obtain instead an undesired unitary evolution on the *joint* state of the qubit *and* its environment. If we then considered the state of the qubit alone, it would no longer be pure but rather now mixed. Thus, we model the error by imagining that our qubit has accidentally become part of a larger quantum system.

If we adopt this perspective, we can develop a mathematical model of how different types of errors will affect the state of our qubit. Let us imagine that the qubit starts off in state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and the environment starts off in state $|E\rangle$. As the qubit and its environment are assumed to start off independently of one another their initial joint state is a product state of the form:

$$|\Psi\rangle = |\psi\rangle \otimes |E\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes |E\rangle \quad (14.28)$$

Under a general unitary evolution, U , the $|0\rangle|E\rangle$ and $|1\rangle|E\rangle$ components would, ignoring normalization, evolve according to:

$$\begin{aligned} U(|0\rangle \otimes |E\rangle) &= |0\rangle \otimes |E_{00}\rangle + |1\rangle \otimes |E_{01}\rangle \\ U(|1\rangle \otimes |E\rangle) &= |0\rangle \otimes |E_{10}\rangle + |1\rangle \otimes |E_{11}\rangle \end{aligned} \quad (14.29)$$

where $|E_{00}\rangle$, $|E_{01}\rangle$, $|E_{10}\rangle$, and $|E_{11}\rangle$ do not have to be orthogonal to one another. Thus, a qubit in state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ evolves as:

$$\begin{aligned} U|\psi\rangle|E\rangle &= U((\alpha|0\rangle + \beta|1\rangle)|E\rangle) \\ &= \alpha(|0\rangle \otimes |E_{00}\rangle + |1\rangle \otimes |E_{01}\rangle) + \beta(|0\rangle \otimes |E_{10}\rangle + |1\rangle \otimes |E_{11}\rangle) \end{aligned} \quad (14.30)$$

We can re-write the right hand side of (14.30) in terms of distinct states for the qubit. The resulting form indicates that the state of the environment is correlated with the

state of the qubit. In fact, the two are *entangled*.

$$\begin{aligned}
 U|\psi\rangle|E\rangle &= \alpha(|0\rangle\otimes|E_{00}\rangle + |1\rangle\otimes|E_{01}\rangle) + \beta(|0\rangle\otimes|E_{10}\rangle + |1\rangle\otimes|E_{11}\rangle) \\
 &= (\alpha|0\rangle + \beta|1\rangle)\otimes\frac{|E_{00}\rangle + |E_{11}\rangle}{2} \quad (\text{no error}) \\
 &\quad + (\alpha|0\rangle - \beta|1\rangle)\otimes\frac{|E_{00}\rangle - |E_{11}\rangle}{2} \quad (\text{phase flip}) \\
 &\quad + (\alpha|1\rangle + \beta|0\rangle)\otimes\frac{|E_{01}\rangle + |E_{10}\rangle}{2} \quad (\text{bit flip}) \\
 &\quad + (\alpha|1\rangle - \beta|0\rangle)\otimes\frac{|E_{01}\rangle - |E_{10}\rangle}{2} \quad (\text{joint phase flip \& bit flip})
 \end{aligned} \tag{14.31}$$

Loosely speaking,² this allows us to *interpret* the undesired unitary evolution of the joint state of the qubit and its environment as if one of four possible events have afflicted the qubit: no-error occurred (in which case $\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|0\rangle + \beta|1\rangle$), a phase-flip error occurred (in which case $\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|0\rangle - \beta|1\rangle$), a bit-flip error occurred (in which case $\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|1\rangle + \beta|0\rangle$), or simultaneous phase flip and bit flip errors occurred (in which case $\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|1\rangle - \beta|0\rangle$ up to an unimportant overall phase).

The alert reader will recognize these four error modes as being describable by the action of one of the Pauli matrices, $\mathbb{1}$, X , Y , and Z , on the error-free qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Recognizing that a Pauli Y operation is $Y = X \cdot Z$ up to an overall phase factor, we can write:

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{\mathbb{1}} \alpha|0\rangle + \beta|1\rangle \quad (\text{no error}) \tag{14.32}$$

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{X} \alpha|0\rangle - \beta|1\rangle \quad (\text{phase flip error}) \tag{14.33}$$

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{Z} \alpha|1\rangle + \beta|0\rangle \quad (\text{bit flip error}) \tag{14.34}$$

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{X \cdot Z} \alpha|1\rangle - \beta|0\rangle \quad (\text{simultaneous phase flip \& bit flip error}) \tag{14.35}$$

where $X \cdot Z = -iY$. Thus, essentially, the error afflicting the qubit can be thought of as an “unwanted” evolution of the qubit under the action of one of the four Pauli matrices. These correspond to no-error ($\mathbb{1}$), a bit-flip error (X), a phase-flip error (Z), and a joint bit-flip and phase-flip error ($Y = iX \cdot Z$). In retrospect, this is not surprising perhaps, because the Pauli matrices form a basis for all 2×2 matrices.

That is, *any* matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ can be factored as:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \frac{a+d}{2}\mathbb{1} + \frac{b+c}{2}X + \frac{i(b-c)}{2}Y + \frac{a-d}{2}Z \tag{14.36}$$

²We say “loosely speaking” because we can only really adopt this interpretation when the states of the environment are orthogonal to one another.

We can extend the aforementioned error model to multiple qubits by assuming the various error types afflict each qubit independently. Thus, the operators describing all possible independent errors that might afflict n -qubits are precisely those of the Pauli group \mathcal{P}_n —the group consisting of all direct products of the Pauli operators $\mathbb{1}$, X , $X \cdot Z$, and Z having overall phase ± 1 or $\pm i$. Thus, if we were interested in encoding, say, one qubit into an entangled state of five qubits, there would (ignoring overall phase) be $4^5 = 1024$ distinct error operators formed from the direct product of a single qubit error operator, $\mathbb{1}$, X , Z , $X \cdot Z$, for each qubit in all possible ways. However, if we only wanted to guarantee the ability to correct up t errors amongst any of these five qubits, then we need only consider a sub-group of these Pauli operators that contained at most t Pauli terms (treating $X \cdot Z$ as *one* “Pauli” term as $X \cdot Z = -iY$).

The error operators that make up the Pauli group possess certain properties that will be of use to us later:

- The eigenvalues of $\mathcal{E}_\alpha \in \mathcal{P}_n$ are ± 1 or $\pm i$.
- Squaring an error operator is the identity up to a real phase factor of ± 1 , i.e., $\forall \mathcal{E}_\alpha \in \mathcal{P}_n : \mathcal{E}_\alpha^2 = \pm \mathbb{1}$
- The group is closed under the dot product of its elements, i.e., $\forall \mathcal{E}_\alpha, \mathcal{E}_\beta \in \mathcal{P}_n : \mathcal{E}_\alpha \cdot \mathcal{E}_\beta \in \mathcal{P}_n$
- Pairs of error operators either commute or anti-commute, i.e., $\forall \mathcal{E}_\alpha, \mathcal{E}_\beta \in \mathcal{P}_n : [\mathcal{E}_\alpha, \mathcal{E}_\beta] = 0$ (commute) or $\{\mathcal{E}_\alpha, \mathcal{E}_\beta\} = 0$ (anti-commute)

Later we will specialize our interest to sub-groups of these operators, e.g., the sub-group of error operators that contain at most one Pauli error per operator.

14.3.4 Protecting Quantum Information via Encoding

Next we turn to the question of how to encode the logical state we want to protect within a larger Hilbert space so that any errors that subsequently afflict our encoded state can be guaranteed to be detectable and correctable.

The trick, as John Preskill says, is “to use entanglement to fight entanglement”. The key idea is to entangle, in a special way, a logical qubit we want to protect with n ancillae qubits such that a subsequent measurement, in the computational basis, of just the n ancillae qubits will project the (now specially entangled) $(n + 1)$ -qubit state into a *different orthogonal subspace* depending on which type of error (“bit-flip”, “phase-flip” “joint bit-flip and phase-flip”, or “no-error”) has afflicted which of the $n + 1$ qubits. The set of quantum states that span this encoding space are called the “quantum codewords”, $\{|\psi_i\rangle\}$. The rationale for doing this is that, if the right set of measurements on n of the $(n + 1)$ qubits can project the $(n + 1)$ -qubit state into a different orthogonal subspace depending on what error occurred, we can use the outcome of these measurements to serve as a so-called “error-syndrome”, which diagnoses what error occurred. Once, the error has become known, it then becomes easy to correct it by applying the appropriate unitary operator.

To ensure our quantum codewords will behave in the way we need them to, they must be designed with the type of error we want them to protect against in mind. In fact, a little thought allows us to stipulate a criterion that the quantum codewords will have to meet in order to guarantee that we can always detect an error [270].

Quantum Codewords: Criterion for Errors to Be Detectable For an error $\mathcal{E}_\alpha \in \mathcal{E}$ that afflicts a quantum codeword to be detectable then, for every pair of valid quantum codewords $|\psi_i\rangle$ and $|\psi_j\rangle$ that span the encoding space, we require:

$$\langle \psi_j | \mathcal{E}_\alpha | \psi_i \rangle = c_\alpha \delta_{ij} \quad (14.37)$$

When this criterion is met, it will guarantee that an error-afflicted codeword, $\mathcal{E}_\alpha |\psi_i\rangle$, will be distinguishable from all the valid codewords, $|\psi_j\rangle$.

The aforementioned criterion tells us a property our codewords will need to possess to be able to *detect* errors. But what property must they possess to also be able to *correct* errors? Well, what do we need to ensure to guarantee we can correct any error? We have to be certain that we won't confuse one error with another when acting on different quantum codewords. Rather, the error syndrome has to be unique for each of the different types of errors acting on the different possible quantum codewords. This basic strategy is the foundation of all quantum error correcting codes (QECCs).

Thus, to ensure our codewords will be useful for correcting errors, in addition to detecting them, we therefore need them to satisfy the following correctability criterion.

Quantum Codewords: Criterion for Errors to Be Correctable For an error $\mathcal{E}_\alpha \in \mathcal{E}$ that afflicts a quantum codeword to be correctable, it needs to be distinguishable from all errors afflicting all other codewords. That is, if $|\psi_i\rangle$ and $|\psi_j\rangle$ are any pair of valid codewords, we require:

$$\langle \psi_j | \mathcal{E}_\beta^\dagger \mathcal{E}_\alpha | \psi_i \rangle = c_{\alpha\beta} \delta_{ij} \quad \forall \mathcal{E}_\alpha, \mathcal{E}_\beta \in \mathcal{E} \quad (14.38)$$

When this criterion is met, we can guarantee that an error $\mathcal{E}_\alpha \in \mathcal{E}$ afflicting one codeword is distinguishable from an error $\mathcal{E}_\beta \in \mathcal{E}$ afflicting a different codeword. In this case we would have $\langle \psi_i | \mathcal{E}_\beta^\dagger \mathcal{E}_\alpha | \psi_j \rangle = 0$. Furthermore, the criterion also guarantees that when different error operators afflict the same codeword, as described by $\langle \psi_i | \mathcal{E}_\beta^\dagger \mathcal{E}_\alpha | \psi_i \rangle$, that the result is independent of the codeword. This means that neither the environment nor the decoding operation learns anything about the encoded state during error detection and correction. This is an essential requirement to be sure the error detection and correction procedures do not cause more damage to the state we are trying to protect.

Thus, it should be apparent that the family of errors that we want to be able to detect and correct, and the number of qubits into which we encode each logical qubit we want to protect, will influence the options available to us for picking quantum

codewords that meet the detectability and correctability criteria. As we show below, quantum codewords having the desired properties can be constructed, and we will give examples of 9-qubit, 7-qubit, and 5-qubit coding schemes that are able to correct a single Pauli error, $\mathbb{1}$, X , Z , $X \cdot Z$, afflicting any of their qubits.

14.3.5 Digitizing and Diagnosing Errors by Measuring Error Syndromes

A striking aspect of such quantum error-correcting codes, is that the act of measuring the ancillae qubits to obtain the error-syndrome can be viewed as *determining which error has afflicted which qubit*. Prior to such measurements, which error (if any) has occurred is undetermined. In fact, pre-measurement, the state may contain a superposition of possible errors any of which are still possible outcomes. However, by making the error-syndrome measurements a particular error is determined. Forcing such an error decision is a rational thing to do, because the error then becomes known, and a large known error is entirely correctable, whereas a small unknown one is not. So the cleverness of quantum error-correcting codes is that they exploit the superposition-destroying nature of quantum measurements to render an unknown error known, and entanglement to link the measured error-syndrome to the error-type afflicting the logical qubit.

14.3.6 Reversing Errors via Inverse Error Operators

Once the error becomes known, as a result of measuring the error syndrome, it can be corrected by applying the inverse of the appropriate Pauli error operator.

14.3.7 Abstract View of Quantum Error Correcting Codes

The general approach to quantum error correcting codes outlined above, can be abstracted into a theory based on the properties of operators and sub-spaces. Stepping back a moment, the general idea is to encode a logical qubit whose state we want to protect within a set of n -qubits, i.e., within a 2^n -dimensional Hilbert space, such that there is a special sub-space \mathcal{C} , called the codespace, that is spanned by a set of quantum states, $\text{span}(\{|\psi_i\rangle\})$, i.e. the quantum codewords. The codewords are carefully chosen so that we can guarantee, for a given set of error operators, \mathcal{E} , that the error detectability and error correctability criteria are met. That is, every error operator $\mathcal{E}_\alpha \in \mathcal{E}$ takes a codeword into a state that is orthogonal to all other codewords, and the errors induced by one error operator can be distinguished from those induced by another. Thus every error is uniquely identifiable and hence correctable.

14.3.7.1 Minimal Distance of a Code

Our primary concern is how many errors a given code can correct? We approach this by determining the *distance* of the code.

Let us start with the error operators. These are all direct products of 1-qubit Pauli matrices and the identity matrix. Define the *weight* of such an operator to be the number non-identity operators in its direct product representation. We can then define the *minimum distance* of a code to be equal to the smallest weight of any operator $\mathcal{E}_\gamma \in \mathcal{E}$ such that the error correctability criterion (14.38) is violated.

What does this imply about the relationship between the distance of a code and how many errors it can correct? Well, for a QECC to be useful, it has to be able to distinguish between how different errors affect different codewords. So in the correctability criterion we use the operator $\mathcal{E}_\beta^\dagger \mathcal{E}_\alpha$. But if error operators \mathcal{E}_α and \mathcal{E}_β are in the group \mathcal{E} , then so is the operator $\mathcal{E}_\gamma = \mathcal{E}_\beta^\dagger \mathcal{E}_\alpha$. If we are working with a subgroup of error operators such that each operator contains at most t Pauli matrices, then the operator $\mathcal{E}_\beta^\dagger \mathcal{E}_\alpha$ could have weight up to $2t$. To guarantee correctability we therefore require the minimum distance d to exceed this potential weight, i.e., $d \geq 2t + 1$. This implies that our code can only be guaranteed to correct up to at most $t = \lfloor \frac{d-1}{2} \rfloor$ general errors, i.e., bit-flips, phase-flips, or joint bit-flips and phase flips.

14.3.7.2 $[n, K, d]$ Quantum Error Correcting Code

Thus, the principal characteristics of a quantum code are the number of qubits used in the encoding, n , the dimension of the codespace, K , and the minimum distances of the code d , which is related to the maximum number of errors the code can be guaranteed to correct, t_{\max} , via $t_{\max} = \lfloor \frac{d-1}{2} \rfloor$. Quantum error correcting codes are therefore often described using the notation $[n, K, d]$. An $[n, K, d]$ code can *detect* up to $(d - 1)$ errors, and *correct* up to $\lfloor \frac{d-1}{2} \rfloor$ general 1-qubit errors. The smallest quantum error correcting code able to correct a single general error is a $[5, 2, 3]$ code. In this case, $n = 5$, $K = 2$, $d = 3$ and so $t_{\max} = \lfloor \frac{d-1}{2} \rfloor = 1$.

14.3.7.3 Additive (Stabilizer) Code Versus Non-additive Code

Within the class of quantum codes, the most important distinction is between the additive codes and the non-additive ones. If the codespace of a quantum error correcting code is specified by the joint $+1$ eigenspace of an Abelian sub-group of local Pauli operators (i.e., operators writable as a direct product of Pauli matrices that all commute with one another), then the code is said to be an “additive” or “stabilizer” code. That is if the errors are specified as an Abelian sub-group of the Pauli group, and have the property that on the codewords $\{|\psi_i\rangle\}$ that $\forall \mathcal{E}_\alpha \in \mathcal{E} \in \mathcal{P}_n : \mathcal{E}_\alpha |\psi_i\rangle = +1 |\psi_i\rangle$, then the code is an additive or stabilizer code.

If the aforementioned condition on the codespace does not hold, the code is “non-additive”.

Table 14.4 Notation often used to describe classical and quantum error-correcting codes

Notation	Name	Meaning
(n, K, d)	Classical code	An n -bit classical code having a K -dimensional codespace and distance d
$\langle n, K, d \rangle$	Quantum code	An n -qubit quantum code having a K -dimensional codespace and distance d . This class includes additive and non-additive quantum codes. The latter have the potential to be more efficient than additive codes. Note that the codespace dimension of a non-additive code need not be a power of two
$\llbracket n, k, d \rrbracket$	Quantum stabilizer code	An additive (stabilizer) n -qubit quantum code having a 2^k -dimensional codespace and distance d . This class of quantum codes includes the 9-qubit Shor, 7-qubit Steane, and 5-qubit Laflamme codes. The codespace dimension of an additive code is always a power of two

The notation $\langle n, K, d \rangle$ is used to describe both additive and non-additive quantum codes. However, the codespace dimension of additive codes is always a power of two, i.e., $K = 2^k$ for some k , whereas this is not necessarily so for a non-additive code. The additive (stabilizer) codes are often described in terms of a special notation $\llbracket n, k, d \rrbracket$ (where $k = \log_2 K$). Thus, whereas we can speak of an additive code protecting k qubits within n -qubit quantum codewords, we cannot really say this for a non-additive code since $\log_2 K$ is not necessarily an integer. However, the greater complexity of non-additive codes is offset by their potential to be more efficient than additive codes. Table 14.4 summarizes the notation we just discussed.

Quantum codes have other characteristics that can be of interest including whether they are pure or impure, degenerate or non-degenerate, and perfect or imperfect.

14.3.7.4 Pure Versus Impure Code

If distinct elements of \mathcal{E} produce orthogonal results, the code is said to be pure. Otherwise it is impure.

14.3.7.5 Degenerate Versus Non-degenerate Code

If linearly independent correctable errors acting on the codespace are guaranteed to yield linearly independent states, the code is said to be *non-degenerate*. Thus, a non-degenerate code assigns a unique linearly independent error-syndrome to each possible error. Most known quantum error correcting codes are non-degenerate additive (stabilizer) codes. If a additive (stabilizer) code is also a pure code, it is guaranteed to be non-degenerate, but the converse need not be true.

Conversely, if linearly independent correctable errors acting on the codeword space can produce linearly dependent states, the code is said to be *degenerate*. Degenerate codes are interesting because they have the potential to be much more efficient than non-degenerate codes.

Theorems placing bounds on non-degenerate quantum codes do not necessarily apply to degenerate codes. Therefore, before applying a theorem, verify that the theorem holds for the type of code with which you are working.

14.3.7.6 Perfect Versus Imperfect Code

If every error syndrome corresponds to a correctable error, the code is said to be *perfect* otherwise it is imperfect.

Having re-considered quantum correcting codes in the abstract let us now turn to a concrete example of the optimal additive quantum code able to correct a single general error.

14.4 Optimal Quantum Error Correcting Code

A natural question to ask is how good a quantum error correcting code can be? That is what are the tradeoffs between the number of qubits used in the codeword, the number of qubits the code protects, and the number of general errors that such a code can correct? The following simple argument suggests a $\llbracket 5, 1, 3 \rrbracket$ code is the smallest code able to correct a single general error.

14.4.1 Laflamme-Miquel-Paz-Zurek's 5-Qubit Code

Imagine a code that encodes one logical qubit into n qubits and we wish to protect against a single error on any one of these n qubits. If we assume that errors are sufficiently rare that at most one error can afflict one of the n qubits, then each qubit can undergo one of three types of error so there are $3n$ ways the error can be introduced. Add to this the possibility that none of the qubits have an error, we obtain a total of $(3n + 1)$ possible error “diagnoses”. If we are to distinguish between the possible $(3n + 1)$ error diagnoses by making measurements on $n - 1$ qubits, i.e., the ancillae, then these can index 2^{n-1} different states, and so the code has to satisfy $(3n + 1) \leq 2^{n-1}$. The smallest integer satisfying this condition is $n = 5$. Hence, the smallest code sufficient to correct a single general error must be at least a 5-qubit code. Such a 5-qubit code was constructed by Raymond Laflamme, Cesar Miquel, Pablo Paz, and Wojciech Zurek [297] in 1996.

14.4.2 Error Operators for the 5-Qubit Code

In the Laflamme-Miquel-Paz-Zurek 5-qubit code we wish to be able to correct a general error amongst the five qubits in the encoded state. Hence, the only error

operators we need consider are:

$$\begin{aligned}
 \mathcal{E}_{\text{None}} &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \\
 \mathcal{E}_{\text{B5}} &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes X \\
 \mathcal{E}_{\text{BP5}} &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes X \cdot Z \\
 \mathcal{E}_{\text{P5}} &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes Z \\
 \mathcal{E}_{\text{B4}} &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes X \otimes \mathbb{1} \\
 \mathcal{E}_{\text{BP4}} &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes X \cdot Z \otimes \mathbb{1} \\
 \mathcal{E}_{\text{P4}} &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes Z \otimes \mathbb{1} \\
 \mathcal{E}_{\text{B3}} &= \mathbb{1} \otimes \mathbb{1} \otimes X \otimes \mathbb{1} \otimes \mathbb{1} \\
 \mathcal{E}_{\text{BP3}} &= \mathbb{1} \otimes \mathbb{1} \otimes X \cdot Z \otimes \mathbb{1} \otimes \mathbb{1} \\
 \mathcal{E}_{\text{P3}} &= \mathbb{1} \otimes \mathbb{1} \otimes Z \otimes \mathbb{1} \otimes \mathbb{1} \\
 \mathcal{E}_{\text{B2}} &= \mathbb{1} \otimes X \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \\
 \mathcal{E}_{\text{BP2}} &= \mathbb{1} \otimes X \cdot Z \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \\
 \mathcal{E}_{\text{P2}} &= \mathbb{1} \otimes Z \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \\
 \mathcal{E}_{\text{B1}} &= X \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \\
 \mathcal{E}_{\text{BP1}} &= X \cdot Z \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \\
 \mathcal{E}_{\text{P1}} &= Z \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1}
 \end{aligned} \tag{14.39}$$

which includes, you notice, the possibility of there being no errors at all, i.e., $\mathcal{E}_{\text{None}}$.

14.4.3 Encoding Scheme for the 5-Qubit Code

In the Laflamme-Miquel-Paz-Zurek code a single logical qubit is encoded in a 5-qubit entangled state of the form:

$$\begin{aligned}
 |0\rangle_L &= \frac{1}{2\sqrt{2}}(|00000\rangle + |00110\rangle + |01001\rangle - |01111\rangle \\
 &\quad + |10011\rangle + |10101\rangle + |11010\rangle - |11100\rangle) \\
 |1\rangle_L &= \frac{1}{2\sqrt{2}}(|11111\rangle + |11001\rangle + |10110\rangle - |10000\rangle \\
 &\quad - |01100\rangle - |01010\rangle - |00101\rangle + |00011\rangle)
 \end{aligned} \tag{14.40}$$

A general state of a qubit we want to protect, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, is mapped into an entangled state of the form $|\psi\rangle_L = \alpha|0\rangle_L + \beta|1\rangle_L$. Subsequently, if a single bit-flip, phase-flip, or joint bit-flip and phase-flip afflicts any of these five qubits, there is sufficient information in their entanglement to be able to determine, from the measured error syndrome, the operation that must be performed on the unmeasured qubit to restore it to its original state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

Figure 14.3 shows a quantum circuit for creating the entangled encoded state used in the Laflamme-Miquel-Paz-Zurek quantum error-correcting code. This circuit entangles a single qubit in an arbitrary state $|\psi\rangle$ with four ancilla qubits, each initially in state $|0\rangle$, to create the encoded state $|\psi\rangle_L = \alpha|0\rangle_L + \beta|1\rangle_L$, having basis vectors $|0\rangle_L$ and $|1\rangle_L$ as in (14.40), and single qubit gates L and L^\dagger defined by:

$$\begin{aligned} L &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \\ L^\dagger &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \end{aligned} \quad (14.41)$$

After such an encoding, the 5-qubit state may then be afflicted with a single bit-flip, phase-flip, or joint bit-flip and phase-flip in the region marked “ERROR” in Fig. 14.4. This would correspond to an error being introduced while an encoded qubit was being stored in a quantum memory. For example, if a single bit-flip occurs

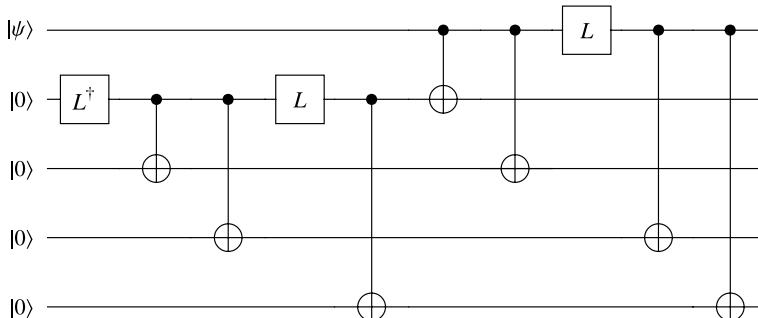


Fig. 14.3 Quantum circuit for encoding unknown quantum state $|\psi\rangle$ amongst the amplitudes of a 5-qubit entangled state such that any subsequent bit flip, phase shift or joint bit flip/phase shift error can be detected and corrected

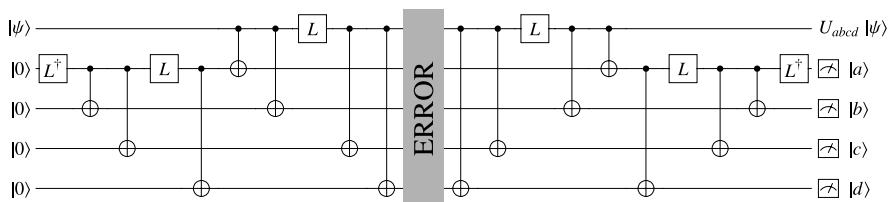


Fig. 14.4 Quantum circuit implementing the Laflamme-Miquel-Paz-Zurek 5-qubit quantum error correcting code. The left hand side of the circuit encodes a single logical qubit in an entangled 5-qubit state. In the encoded form, the state is protected against a single bit-flip, phase-flip, or joint bit-flip and phase-flip acting on any of the five qubits. To diagnose and correct the error, the encoded state must be decoded and the error syndrome measured. Depending on the outcome, $|a\rangle|b\rangle|c\rangle|d\rangle$, a unitary operator U_{abcd} is applied to the top qubit which rotates it into the original state of the logical qubit and hence the error is undone

on the third qubit of the encoded state $|\psi\rangle_L$, the state would change according to:

$$|\psi\rangle_L \xrightarrow{\mathcal{E}_{B3}} -\beta|01111\rangle + \alpha|11111\rangle \quad (14.42)$$

Likewise, if a single phase-flip occurs on the fourth qubit of the encoded state $|\psi\rangle_L$, the state would change according to:

$$|\psi\rangle_L \xrightarrow{\mathcal{E}_{P4}} \alpha|01110\rangle - \beta|11110\rangle \quad (14.43)$$

Similar state changes are induced by a bit-flip, a phase-flip, or a joint bit-flip and phase-flip on any of the qubits in the encoded state $|\psi\rangle_L$.

Now that we know how an error changes the encoded state, we next need to figure out how to diagnose what error has occurred and determine the appropriate corrective action to undo that error and restore the logical qubit to its pristine state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. To do this, we will use the part of the circuit in Fig. 14.4 to the right of the region marked “ERROR”.

14.4.4 Error Syndromes & Corrective Actions for the 5-Qubit Code

To diagnose what single error has afflicted the 5-qubit encoded state, we run the buggy encoded state through the decoding circuit shown in Fig. 14.5. This is just the encoding circuit run in the reverse direction. The decoding operation produces an output entangled state that can be factored as a *superposition* of the form $\sum_{a=0}^1 \sum_{b=0}^1 \sum_{c=0}^1 \sum_{d=0}^1 \alpha_{abcd} |\varphi_{abcd}\rangle |abcd\rangle$ where $|abcd\rangle$ is a 4-bit computational basis vector, and the $|\varphi_{abcd}\rangle$ states are unitary rotations of state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. We can, given knowledge of the encoding-decoding circuitry and the error-modes, build a lookup table that gives the required rotation to map each $|\varphi\rangle$ back into $|\psi\rangle$. Such rotation are shown in Table 14.5. Thus, by measuring the four ancillae qubits, in the computational basis, in the output from the decoding circuit we can project out a specific state $|\varphi_{abcd}\rangle$ state, and use Table 14.5 to determine the corrective action needed to recover the correct state ($|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$) of the logical qubit.

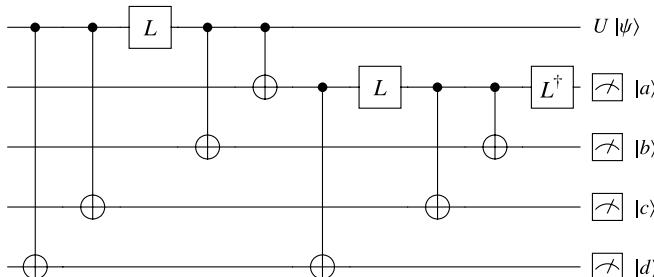


Fig. 14.5 Quantum circuit for decoding a (potentially corrupted) entangled state and measuring its error syndrome to reveal whether or not an error had occurred, and if so, what action to apply to the top qubit to undo the error

Table 14.5 To protect an unknown quantum state $|\psi\rangle$ while in storage in a quantum memory register, we entangle $|\psi\rangle$ with four ancilla qubits each prepared initially in the state $|0\rangle$, using the left hand side of the quantum circuit shown in Fig. 14.4. Once encoded, the state can be corrupted by a single bit-flip, a single phase-flip or a single phase-flip followed by a bit-flip on any of the five qubits. However, when we want to retrieve our protected state, we decode the entangled state by running it through the right hand side of the circuit shown in Fig. 14.4 and then “measure the error syndrome”, i.e., read the bit values of the four ancilla qubits. Based on the observed values we can then look up corrective action to apply to the top qubit to restore it to its pristine (yet unknown) state. In the table an error BPi means a phase flip followed by a bit flip on the i -th qubit

Error type	State produced $ \psi\rangle a\rangle b\rangle c\rangle d\rangle$	Error syndrome	Corrective action U_{abcd}	Result
None	$-\beta 01011\rangle + \alpha 11011\rangle$	{1, 0, 1, 1}	$Z \cdot X$	$(\alpha 0\rangle + \beta 1\rangle) 1011\rangle$
B1	$\beta 01000\rangle + \alpha 11000\rangle$	{1, 0, 0, 0}	X	$(\alpha 0\rangle + \beta 1\rangle) 1000\rangle$
B2	$-\beta 00010\rangle + \alpha 10010\rangle$	{0, 0, 1, 0}	$Z \cdot X$	$(\alpha 0\rangle + \beta 1\rangle) 0010\rangle$
B3	$-\beta 01111\rangle + \alpha 11111\rangle$	{1, 1, 1, 1}	$Z \cdot X$	$(\alpha 0\rangle + \beta 1\rangle) 1111\rangle$
B4	$-\beta 01001\rangle + \alpha 11001\rangle$	{1, 0, 0, 1}	$Z \cdot X$	$(\alpha 0\rangle + \beta 1\rangle) 1001\rangle$
B5	$-\beta 01010\rangle + \alpha 11010\rangle$	{1, 0, 1, 0}	$Z \cdot X$	$(\alpha 0\rangle + \beta 1\rangle) 1010\rangle$
P1	$\alpha 00110\rangle - \beta 10110\rangle$	{0, 1, 1, 0}	Z	$(\alpha 0\rangle + \beta 1\rangle) 0110\rangle$
P2	$\beta 01101\rangle + \alpha 11101\rangle$	{1, 1, 0, 1}	X	$(\alpha 0\rangle + \beta 1\rangle) 1101\rangle$
P3	$\beta 00011\rangle + \alpha 10011\rangle$	{0, 0, 1, 1}	X	$(\alpha 0\rangle + \beta 1\rangle) 0011\rangle$
P4	$\alpha 01110\rangle - \beta 11110\rangle$	{1, 1, 1, 0}	Z	$(\alpha 0\rangle + \beta 1\rangle) 1110\rangle$
P5	$\alpha 00000\rangle - \beta 10000\rangle$	{0, 0, 0, 0}	Z	$(\alpha 0\rangle + \beta 1\rangle) 0000\rangle$
BP1	$-\alpha 00101\rangle - \beta 10101\rangle$	{0, 1, 0, 1}	$Z \cdot X \cdot Z \cdot X$	$(\alpha 0\rangle + \beta 1\rangle) 0101\rangle$
BP2	$\beta 00100\rangle + \alpha 10100\rangle$	{0, 1, 0, 0}	X	$(\alpha 0\rangle + \beta 1\rangle) 0100\rangle$
BP3	$\beta 00111\rangle + \alpha 10111\rangle$	{0, 1, 1, 1}	X	$(\alpha 0\rangle + \beta 1\rangle) 0111\rangle$
BP4	$\alpha 01100\rangle - \beta 11100\rangle$	{1, 1, 0, 0}	Z	$(\alpha 0\rangle + \beta 1\rangle) 1100\rangle$
BP5	$\alpha 00001\rangle - \beta 10001\rangle$	{0, 0, 0, 1}	Z	$(\alpha 0\rangle + \beta 1\rangle) 0001\rangle$

14.4.5 Example: Correcting a Bit-Flip

Suppose the logical qubit we wish to protect is in state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. To protect this qubit against error we augment $|\psi\rangle$ state with four ancillae qubits each prepared in state $|0\rangle$ to give us the input state:

$$|\Psi_{\text{in}}\rangle = \alpha|00000\rangle + \beta|10000\rangle \quad (14.44)$$

Encoding this state using the Laflamme-Miquel-Paz-Zurek 5-qubit encoding circuit gives us the state:

$$\begin{aligned} |\Psi_{\text{middle}}\rangle = & \frac{1}{2\sqrt{2}}(\alpha|00000\rangle + \beta|00011\rangle - \beta|00101\rangle + \alpha|00110\rangle \\ & + \alpha|01001\rangle - \beta|01010\rangle - \beta|01100\rangle - \alpha|01111\rangle \\ & - \beta|10000\rangle + \alpha|10011\rangle + \alpha|10101\rangle + \beta|10110\rangle \\ & + \beta|11001\rangle + \alpha|11010\rangle - \alpha|11100\rangle + \beta|11111\rangle) \end{aligned} \quad (14.45)$$

This is an entangled state that can now protect our logical qubit from error. For example, imagine introducing a bit-flip error on the third qubit in this state creating the buggy state:

$$\begin{aligned} |\Psi_{\text{buggy}}\rangle = & \frac{1}{2\sqrt{2}}(\alpha|00100\rangle + \beta|00111\rangle - \beta|00001\rangle + \alpha|00010\rangle \\ & + \alpha|01101\rangle - \beta|01110\rangle - \beta|01000\rangle - \alpha|01011\rangle \\ & - \beta|10100\rangle + \alpha|10111\rangle + \alpha|10001\rangle + \beta|10010\rangle \\ & + \beta|11101\rangle + \alpha|11110\rangle - \alpha|11000\rangle + \beta|11011\rangle) \end{aligned} \quad (14.46)$$

Decoding the buggy state using the Laflamme-Miquel-Paz-Zurek decoding circuit gives us the state:

$$|\Psi_{\text{out}}\rangle = -\beta|01111\rangle + \alpha|11111\rangle \quad (14.47)$$

Reading ancillae state $|abcd\rangle$ then gives the error syndrome is 1111. Using the lookup table, Table 14.5, the appropriate corrective action to apply to the top (unmeasured qubit) should be $Z \cdot X$. Applying this operation, we see that we do indeed restore the top qubit to its error free state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

Quantum error correcting codes *are* therefore feasible even though we are unable to read the encoded data directly without necessarily perturbing the state being read. The proof of the feasibility of QECCs was one of the most important discoveries in the development of quantum computing because their existence means that it is not necessary to fabricate, initialize, and run quantum computers perfectly in order to obtain correct results.

14.5 Other Additive Quantum Error Correcting Codes

The 5-qubit was not the first quantum error correcting code discovered that was able to correct for a single general error amongst the encoded qubits. In fact, two other codes pre-date it, but both require more qubits to encode the data being protected.

14.5.1 Shor's 9-Qubit Code

The first quantum error-correcting code (QECC) was devised by Peter Shor in 1995 [456]. It encodes each logical qubit in nine physical qubits in such manner that a single bit-flip, phase-flip, or joint bit-flip and phase flip, afflicting any of these nine qubits can be identified, and undone by performing an appropriate unitary operation which differs depending on the outcome of the ancilla measurements.

The encoding step in the 9-qubit code involves mapping each logical qubit to encoded form according to:

$$\begin{aligned} |0\rangle_L &= \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \\ |1\rangle_L &= \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \end{aligned} \quad (14.48)$$

Once in this form the encoded data is protected against a single error in any qubit amongst any of the nine qubits.

14.5.2 Steane's 7-Qubit Code

In 1996 Andrew Steane improved upon Peter Shor's 9-qubit code with a 7-qubit code [477, 478]. The encoding step in the 7-qubit code involves mapping each logical qubit to encoded form according to:

$$\begin{aligned} |0\rangle_L &= \frac{1}{2\sqrt{2}}(|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle \\ &\quad + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle) \\ |1\rangle_L &= \frac{1}{2\sqrt{2}}(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle \\ &\quad + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle) \end{aligned} \tag{14.49}$$

Once in this form the encoded data is protected against a single error in any qubit amongst any of the seven qubits.

14.6 Stabilizer Formalism for Quantum Error Correcting Codes

The foregoing quantum error correcting codes were either constructed based on analogies with pre-existing classical codes, or discovered via extensive computer searches. As a result, each code was created in a somewhat makeshift fashion and few, if any, general design principles for quantum codes were learned. One could easily get the impression, therefore, that quantum error correcting codes are discovered serendipitously rather than being constructed systematically to meet desired criteria. Furthermore, one could also get the impression, from our description of the 5-qubit Laflamme-Miquel-Paz-Zurek code, that error correction requires that we periodically map the encoded (and therefore protected qubit) back to its (unprotected) logical basis at which times the qubit is exposed to uncorrectable errors. Neither of these impressions is correct.

In 1996 Daniel Gottesman invented a unified way to think about an important sub-class of QECCs that allows them to be constructed in a more systematic fashion and to perform error correction entirely within the encoded basis so we never re-expose the protected quantum information during the error correction procedure. The 9-qubit Shor, 7-qubit Steane, and 5-qubit Laflamme-Miquel-Paz-Zurek codes as special cases of Gottesman's formalism, which later became known as the "stabilizer formalism" [207, 208]. Using the stabilizer formalism it becomes straightforward to design QECCs to protect against a given set of errors, and to find quantum circuits that will perform the required error diagnosis and recovery operations while staying entirely within the encoded (and therefore protected) basis.

Rather than discuss the stabilizer formalism in the abstract, we will use it to re-analyze the 5-qubit Laflamme-Miquel-Paz-Zurek code, which is the best QECC capable of correcting a single bit-flip, phase-flip, or joint bit-flip and phase-flip afflicting any one of five qubits.

14.6.1 Group Theory for Stabilizer Codes

As the stabilizer formalism relies upon ideas from group theory, we will begin with a brief summary of the key ideas of group theory.

A “group”, \mathcal{G} , is a collection of objects, $g_1, g_2, \dots \in \mathcal{G}$, together with a multiplication operation “ \cdot ”, which possess the following properties:

Group Theory

- **Closure:** the group is closed under “ \cdot ”, i.e., if $g_i, g_j \in \mathcal{G}$ then $g_i \cdot g_j \in \mathcal{G}$.
- **Associativity:** i.e., $(g_i \cdot g_j) \cdot g_k = g_i \cdot (g_j \cdot g_k)$.
- **Existence of Identity:** the group contains an identity element, i.e., $\exists e \in \mathcal{G}$ such that $\forall g_i \in \mathcal{G}, e \cdot g_i = g_i$.
- **Existence of Inverse:** each member of the group has an inverse, i.e., $\forall g_i \in \mathcal{G}, \exists g_i^{-1} \in \mathcal{G}$ s.t. $g_i \cdot g_i^{-1} = e$.

In the context of quantum error correcting codes, the following types of groups and concepts are the most important.

Types of Groups

- **Pauli group:** the group consisting of tensor products of the Pauli matrices, $\mathbb{1}$, X , Y , Z , with an overall phase of ± 1 or $\pm i$.
- **Finite group:** a group \mathcal{G} is finite if the number of elements in it is finite, i.e., the group contains only the elements $g_1, g_2, \dots, g_n \in \mathcal{G}$ for some finite positive integer n .
- **Abelian group:** a group is “Abelian” iff $\forall g_i, g_j \in \mathcal{G}, g_i \cdot g_j = g_j \cdot g_i$.
- **Sub-group:** \mathcal{S} is a sub-group of \mathcal{G} iff the elements $s_1, s_2, \dots \in \mathcal{S}$ are a subset of the elements of $g_1, g_2, \dots \in \mathcal{G}$, and obey the rules for a group in their own right under the same group multiplication operator as that of \mathcal{G} .

The final concept we shall need is that of a “group generator”. The generator, $\{g_1, g_2, \dots, g_\ell\}$, of a group \mathcal{G} is the smallest subset of elements of \mathcal{G} sufficient to generate every member of \mathcal{G} under the multiplication operation for \mathcal{G} . That is we can obtain any element of \mathcal{G} from products of the elements in $\{g_1, g_2, \dots, g_\ell\}$ with repetitions allowed.

We can now describe the basic machinery of the stabilizer formalism using these group-theoretic concepts.

14.6.2 The Stabilizer

A “stabilizer” $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_K\}$ is a carefully chosen group of tensor products of the Pauli operators, $\mathcal{S}_i \in \{\mathbb{1}, X, Z\}^{\otimes n}$ whose elements are required have a simultaneous eigenvalue of $+1$. That is, for some family of states $|\psi\rangle_L$ the stabilizer \mathcal{S} is

a group of tensor products of Pauli operators such that:

$$\begin{aligned} \mathcal{S}_1 |\psi\rangle_L &= +1 |\psi\rangle_L \\ \mathcal{S}_2 |\psi\rangle_L &= +1 |\psi\rangle_L \\ &\vdots \\ \mathcal{S}_K |\psi\rangle_L &= +1 |\psi\rangle_L \end{aligned} \tag{14.50}$$

Furthermore, it is known from pure mathematics that a group of operators can only share a simultaneous eigenvalue when the operators commute with one another. This means that the stabilizer group has to be a finite *Abelian* sub-group of the Pauli group. That is, for the operators in a valid stabilizer, $\mathcal{S}_i \cdot \mathcal{S}_j = \mathcal{S}_j \cdot \mathcal{S}_i$.

14.6.3 Example: A Stabilizer for the 5-Qubit Code

There are many sets of tensor products that we could pick as stabilizers, and different choices would induce different quantum error correcting codes. If we focus on the case of QECCs that involve just five physical qubits, then all the relevant stabilizers must involve only five Pauli matrices. But remember, we don't accept just any old set of tensor products. We are specifically looking for sets of tensor products that form an Abelian sub-group.

Of the many alternatives available to us, suppose we had picked the following set of tensor products of Pauli, $\mathbb{1}$, X , and Z , matrices as our stabilizer:

$$\begin{aligned} \mathcal{S}_1 &= X \otimes X \otimes Z \otimes X \otimes \mathbb{1} \\ \mathcal{S}_2 &= X \otimes Z \otimes X \otimes \mathbb{1} \otimes X \\ \mathcal{S}_3 &= Z \otimes \mathbb{1} \otimes X \otimes X \otimes Z \\ \mathcal{S}_4 &= Z \otimes X \otimes \mathbb{1} \otimes Z \otimes X \\ \mathcal{S}_5 &= \mathbb{1} \otimes Z \otimes Z \otimes Z \otimes Z \\ \mathcal{S}_6 &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \end{aligned}$$

This set of tensor products satisfies all the criteria for a group given above. Moreover, the group is Abelian because all its elements commute with one another. However, we can whittle this group down a little further and work just with its generators, i.e. a minimal set of group elements sufficient to generate all members of the group via their products, with repetitions allowed. In particular, we can immediately see that we do not need the element $\mathcal{S}_6 = \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1}$ because the square of any element of the stabilizer is \mathcal{S}_6 . For example, in particular we have:

$$\mathcal{S}_1 \cdot \mathcal{S}_1 = (X \otimes X \otimes Z \otimes X \otimes \mathbb{1})^2 = (\mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1}) = \mathcal{S}_6$$

Likewise, we can drop any one of the remaining five group elements from this sub-group too. For example, we do not need (say) $\mathcal{S}_5 = \mathbb{1} \otimes Z \otimes Z \otimes Z \otimes Z$ because:

$$\begin{aligned}\mathcal{S}_1 \cdot \mathcal{S}_2 \cdot \mathcal{S}_3 \cdot \mathcal{S}_4 &= (X \otimes X \otimes Z \otimes X \otimes \mathbb{1}) \cdot (X \otimes Z \otimes X \otimes \mathbb{1} \otimes X) \\ &\quad \cdot (Z \otimes \mathbb{1} \otimes X \otimes X \otimes Z) \cdot (Z \otimes X \otimes \mathbb{1} \otimes Z \otimes X) \\ &= (\mathbb{1} \otimes Z \otimes Z \otimes Z \otimes Z) = \mathcal{S}_5\end{aligned}\tag{14.51}$$

Thus, to define the stabilizer, we need only work with the generators of the associated Abelian sub-group, namely the tensor products $\langle \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4 \rangle$ where:

$$\begin{aligned}\mathcal{S}_1 &= X \otimes X \otimes Z \otimes X \otimes \mathbb{1} \\ \mathcal{S}_2 &= X \otimes Z \otimes X \otimes \mathbb{1} \otimes X \\ \mathcal{S}_3 &= Z \otimes \mathbb{1} \otimes X \otimes X \otimes Z \\ \mathcal{S}_4 &= Z \otimes X \otimes \mathbb{1} \otimes Z \otimes X\end{aligned}$$

With these definitions, the tensor products in the generator $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$ is an Abelian sub-group of the Pauli group and therefore meets all the criteria needed to be a stabilizer.

14.6.4 Using a Stabilizer to Find the Codewords It Stabilizes

Given a choice of stabilizer $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_K\}$ we can find the family of states $|\psi\rangle_L$ it stabilizes quite easily. As the \mathcal{S}_j are all hermitian, we can characterize the states we seek as the +1 simultaneous eigenstates of the operators $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$. These are the states spanned by the encoded basis vectors $|0\rangle_L$ and $|1\rangle_L$ that correspond to the simultaneous +1 eigenstates of every element of the stabilizer, when the inputs are $|00000\rangle$ and $|11111\rangle$ respectively.

In the case of the 5-qubit Laflamme-Miquel-Paz-Zurek code the stabilizer has four elements $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$. We can measure the eigenvalue of each of these operators individually using the circuit shown in Fig. 14.6. The encoded basis state $|0\rangle_L$ is the output when the eigenvalues are all measured to be +1 (indicated by finding the output in state $|0\rangle$) when the input state is $|00000\rangle$.

Likewise, the encoded basis state $|1\rangle_L$ is the output from the circuit in Fig. 14.7 when the eigenvalues are all measured to be +1 (again, indicated by finding the output in state $|0\rangle$) when the input state is $|11111\rangle$.

Hence, the codespace that is invariant with respect to this stabilizer $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$ is the set of states spanned by:

$$\begin{aligned}|0\rangle_L &= \frac{1}{2\sqrt{2}}(|00000\rangle + |00110\rangle + |01001\rangle - |01111\rangle \\ &\quad + |10011\rangle + |10101\rangle + |11010\rangle - |11100\rangle) \\ |1\rangle_L &= \frac{1}{2\sqrt{2}}(|11111\rangle + |11001\rangle + |10110\rangle - |10000\rangle \\ &\quad - |01100\rangle - |01010\rangle - |00101\rangle + |00011\rangle)\end{aligned}\tag{14.52}$$

Fig. 14.6 Quantum circuit for using the generators, $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$, of a stabilizer sub-group \mathcal{S} to find the corresponding logical 0 codeword, i.e., $|0\rangle_L$. The logical 0 has the property that $\mathcal{S}_j|0\rangle_L = +1|0\rangle_L$ for all $\mathcal{S}_j \in \mathcal{S}$ and is therefore a +1 eigenstate of the stabilizer sub-group \mathcal{S} . As in the eigenvalue estimation algorithm, when the measurement outcomes made on the ancillae are all 0000 the state on the remaining unmeasured five qubits will be projected into a simultaneous eigenstate of the operators $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$, and \mathcal{S}_4 with eigenvalue +1

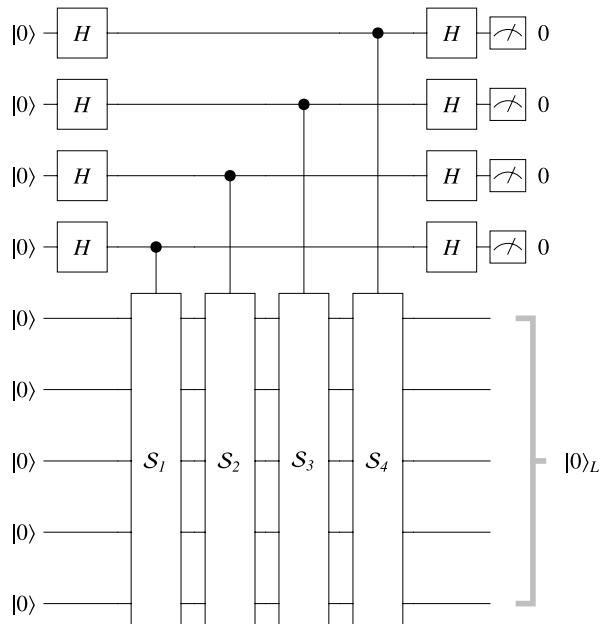
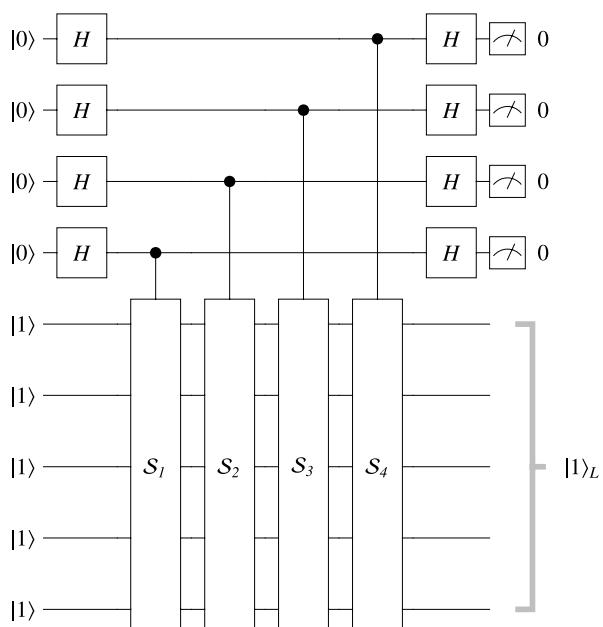


Fig. 14.7 Quantum circuit for using the generators, $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$, of a stabilizer sub-group \mathcal{S} to find the corresponding logical 1 codeword, i.e., $|1\rangle_L$. The logical 1 has the property that $\mathcal{S}_j|1\rangle_L = +1|1\rangle_L$ for all $\mathcal{S}_j \in \mathcal{S}$ and is therefore a +1 eigenstate of the stabilizer sub-group \mathcal{S} . As in the eigenvalue estimation algorithm, when the measurement outcomes made on the ancillae are all 0000 the state on the remaining unmeasured five qubits will be projected into a simultaneous eigenstate of the operators $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$, and \mathcal{S}_4 with eigenvalue +1



These “happen to be” exactly the logical qubits we used on our 5-qubit Laflamme-Miquel-Paz-Zurek code! This means that any state of the form:

$$|\psi\rangle_L = \alpha|0\rangle_L + \beta|1\rangle_L \quad (14.53)$$

such that $|\alpha|^2 + |\beta|^2 = 1$ is stabilized by our stabilizer $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$.

14.6.5 How the Stabilizer is Related to the Error Operators

So far in our discussion I have treated a stabilizer as nothing more than an arbitrary Abelian sub-group of the Pauli group, and I “happened to pick” a stabilizer whose codewords matched those of Sam Braunstein and John Smolin’s version of the pre-existing 5-qubit Laflamme-Miquel-Paz-Zurek code [79, 297]. This was intended to make the connection between stabilizer codes and the 5-qubit code explicit. But, clearly, this is cheating—as I had foreknowledge of the codewords sought, and I worked backwards to find a stabilizer that produced those codewords! It would be more honest to start with the *error operators* we want a quantum error correcting code to correct, and use the *error operators* to derive a stabilizer code able to protect against them. This is the purpose of this section.

In the case of the 5-qubit Laflamme-Miquel-Paz-Zurek code, our intention is to encode a single logical qubit within an entangled 5-qubit state so that the encoded qubit is protected against a single bit-flip, phase-flip, or joint bit-flip and phase flip on any of these five qubits. In this case, the family of error operators we need to protect against is, as we explained earlier, given by:

$$\begin{aligned} \mathcal{E}_{\text{None}} &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \\ \mathcal{E}_{B5} &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes X \\ \mathcal{E}_{BP5} &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes X \cdot Z \\ \mathcal{E}_{P5} &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes Z \\ \mathcal{E}_{B4} &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes X \otimes \mathbb{1} \\ \mathcal{E}_{BP4} &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes X \cdot Z \otimes \mathbb{1} \\ \mathcal{E}_{P4} &= \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes Z \otimes \mathbb{1} \\ \mathcal{E}_{B3} &= \mathbb{1} \otimes \mathbb{1} \otimes X \otimes \mathbb{1} \otimes \mathbb{1} \\ \mathcal{E}_{BP3} &= \mathbb{1} \otimes \mathbb{1} \otimes X \cdot Z \otimes \mathbb{1} \otimes \mathbb{1} \\ \mathcal{E}_{P3} &= \mathbb{1} \otimes \mathbb{1} \otimes Z \otimes \mathbb{1} \otimes \mathbb{1} \\ \mathcal{E}_{B2} &= \mathbb{1} \otimes X \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \\ \mathcal{E}_{BP2} &= \mathbb{1} \otimes X \cdot Z \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \\ \mathcal{E}_{P2} &= \mathbb{1} \otimes Z \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \\ \mathcal{E}_{B1} &= X \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \\ \mathcal{E}_{BP1} &= X \cdot Z \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \\ \mathcal{E}_{P1} &= Z \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \end{aligned} \quad (14.54)$$

which includes the possibility of there being no errors at all, i.e., $\mathcal{E}_{\text{None}}$.

Intuitively, it seems reasonable to expect that our error diagnosis and recovery operations must somehow be related to these error operators. This intuition is indeed correct. The connection is made by way of the stabilizer. Specifically, we want to pick a stabilizer such that every error operator we want to protect against, $\mathcal{E}_\alpha \in \mathcal{E}$, *anti-commutes* with at least one element of the stabilizer, $\mathcal{S}_i \in \mathcal{S}$.

The motivation for this requirement is the following. If an error operator $\mathcal{E}_\alpha \in \mathcal{E}$ *commutes* with an element of the stabilizer $\mathcal{S}_i \in \mathcal{S}$, we have $\mathcal{S}_i \cdot \mathcal{E}_\alpha = \mathcal{E}_\alpha \cdot \mathcal{S}_i$ and so:

$$\mathcal{S}_i \cdot \mathcal{E}_\alpha |\psi\rangle_L = \mathcal{E}_\alpha \cdot \mathcal{S}_i |\psi\rangle_L = +1 \mathcal{E}_\alpha |\psi\rangle_L \quad (14.55)$$

and thus has the eigenvalue $+1$. This means that when we measure the eigenvalue of the operator \mathcal{S}_i whether the input state is pristine, i.e., $|\psi\rangle_L$ or error-afflicted, i.e., $\mathcal{E}_\alpha |\psi\rangle_L$, the eigenvalue will be $+1$ either way. So a good input and a corrupted input will not be distinguishable.

However, if an error operator $\mathcal{E}_\alpha \in \mathcal{E}$ *anti-commutes* with an element of the stabilizer $\mathcal{S}_i \in \mathcal{S}$, we have $\mathcal{S}_i \cdot \mathcal{E}_\alpha = -1 \mathcal{E}_\alpha \cdot \mathcal{S}_i$ and so:

$$\mathcal{S}_i \cdot \mathcal{E}_\alpha |\psi\rangle_L = - \mathcal{E}_\alpha \cdot \mathcal{S}_i |\psi\rangle_L = -1 \mathcal{E}_\alpha |\psi\rangle_L \quad (14.56)$$

and thus has the eigenvalue -1 . In this case, the presence of an error is signalled by the fact that the eigenvalue of the operator \mathcal{S}_i has become -1 .

Hence by measuring the eigenvalue of each element of the stabilizer we can detect whether or not an error has occurred. Moreover, the pattern of anti-commutativity over all the elements in the stabilizer is unique to each different type of error. This allows the pattern of eigenvalues to be used as an *error syndrome* that diagnoses what error occurred unambiguously.

14.6.6 Example: Stabilizers and Error Operators for the 5-Qubit Code

Let us make this concrete in the case of the 5-qubit Laflamme-Miquel-Paz-Zurek code. In this case the error operators we want to protect against are those defined in (14.54). As we are interested in a 5-qubit code, the potential stabilizer elements are therefore all tensor products of any five Pauli matrices taken from the set $\{\mathbb{1}, X, Z\}^{\otimes 5}$. There are 243 distinct possibilities, namely:

$11111 \ 1111X \ 1111Z \ 111X1 \ 111XX \ 111XZ \ 111Z1 \ 111ZX \ 111ZZ$
 $11X11 \ 11X1X \ 11X1Z \ 11XX1 \ 11XXX \ 11XXZ \ 11XZ1 \ 11XZX \ 11XZZ$
 $11Z11 \ 11Z1X \ 11Z1Z \ 11ZX1 \ 11ZXX \ 11ZXZ \ 11ZZ1 \ 11ZZX \ 11ZZZ$
 $1X111 \ 1X11X \ 1X11Z \ 1X1X1 \ 1X1XX \ 1X1XZ \ 1X1Z1 \ 1X1ZX \ 1X1ZZ$
 $1XX11 \ 1XX1X \ 1XX1Z \ 1XXX1 \ 1XXXX \ 1XXXXZ \ 1XXZ1 \ 1XXZX \ 1XXZZ$

$1XZ11 \ 1XZ1X \ 1XZ1Z \ 1XZX1 \ 1XZXX \ 1XZXZ \ 1XZZ1 \ 1XZZX \ 1XZZZ$
 $1Z111 \ 1Z11X \ 1Z11Z \ 1Z1X1 \ 1Z1XX \ 1Z1XZ \ 1Z1Z1 \ 1Z1ZX \ 1Z1ZZ$
 $1ZX11 \ 1ZX1X \ 1ZX1Z \ 1ZXX1 \ 1ZXXX \ 1ZXXX \ 1ZXZ1 \ 1ZXZX \ 1ZXZZ$
 $1ZZ11 \ 1ZZ1X \ 1ZZ1Z \ 1ZZX1 \ 1ZZXX \ 1ZZXZ \ 1ZZZ1 \ 1ZZZX \ 1ZZZZ$
 $X1111 \ X111X \ X111Z \ X11X1 \ X11XX \ X11XZ \ X11Z1 \ X11ZX \ X11ZZ$
 $X1X11 \ X1X1X \ X1X1Z \ X1XX1 \ X1XXX \ X1XXZ \ X1XZ1 \ X1XZX \ X1XZZ$
 $X1Z11 \ X1Z1X \ X1Z1Z \ X1ZX1 \ X1ZXX \ X1ZXZ \ X1ZZ1 \ X1ZZX \ X1ZZZ$
 $XX111 \ XX11X \ XX11Z \ XX1X1 \ XX1XX \ XX1XZ \ XX1Z1 \ XX1ZX \ XX1ZZ$
 $XXX11 \ XXX1X \ XXX1Z \ XXXX1 \ XXXXX \ XXXXX \ XXXZ1 \ XXXZX \ XXXZZ$
 $XXZ11 \ XXZ1X \ XXZ1Z \ XXZX1 \ XXZXX \ XXZXZ \ XXZZ1 \ XXZZX \ XXZZZ$
 $XZ111 \ XZ11X \ XZ11Z \ XZ1X1 \ XZ1XX \ XZ1XZ \ XZ1Z1 \ XZ1ZX \ XZ1ZZ$
 $XZX11 \ XZX1X \ XZX1Z \ XZXX1 \ XZXXX \ XZXXX \ XZXZ1 \ XZXZX \ XZXZZ$
 $XZZ11 \ XZZ1X \ XZZ1Z \ XZZX1 \ XZZXX \ XZZXX \ XZZZ1 \ XZZZX \ XZZZZ$
 $Z1111 \ Z111X \ Z111Z \ Z11X1 \ Z11XX \ Z11XZ \ Z11Z1 \ Z11ZX \ Z11ZZ$
 $Z1X11 \ Z1X1X \ Z1X1Z \ Z1XX1 \ Z1XXX \ Z1XXZ \ Z1XZ1 \ Z1XZX \ Z1XZZ$
 $Z1Z11 \ Z1Z1X \ Z1Z1Z \ Z1ZX1 \ Z1ZXX \ Z1ZXZ \ Z1ZZ1 \ Z1ZZX \ Z1ZZZ$
 $ZX111 \ ZX11X \ ZX11Z \ ZX1X1 \ ZX1XX \ ZX1XZ \ ZX1Z1 \ ZX1ZX \ ZX1ZZ$
 $ZXX11 \ ZXX1X \ ZXX1Z \ ZXXX1 \ ZXXXX \ ZXXXZ \ ZXXZ1 \ ZXXZX \ ZXXZZ$
 $ZXZ11 \ ZXZ1X \ ZXZ1Z \ ZXZX1 \ ZXZXX \ ZXZXZ \ ZXZZ1 \ ZXZZX \ ZXZZZ$
 $ZZ111 \ ZZ11X \ ZZ11Z \ ZZ1X1 \ ZZ1XX \ ZZ1XZ \ ZZ1Z1 \ ZZ1ZX \ ZZ1ZZ$
 $ZZX11 \ ZZX1X \ ZZX1Z \ ZZZX1 \ ZZZXX \ ZZZXX \ ZZXZ1 \ ZZXZX \ ZZXZZ$
 $ZZZ11 \ ZZZ1X \ ZZZ1Z \ ZZZX1 \ ZZZXX \ ZZZZX \ ZZZZ1 \ ZZZZX \ ZZZZZ$

Next we determine which of these *potential* stabilizer elements anti-commute with each error operator $\mathcal{E}_\alpha \in \mathcal{E}$. Two operators anti-commute when $\{A, B\} = A \cdot B + B \cdot A = 0$. Error operators that anti-commute with any element of the stabilizer, correspond to errors that are *detectable* by the corresponding stabilizer code. However, to be *correctable*, each error operator needs to have a different pattern of anti-commutativity with the elements of the stabilizer. We can find the patterns of anti-commutativity by computer search very easily. The result will look something like:

$$\begin{aligned}
\mathcal{E}_{\text{None}} &= 11111 \text{ anti-commutes with } F_1 = \{\} \\
\mathcal{E}_{B5} &= 1111X \text{ anti-commutes with } F_2 = \{1111Z, 111XZ, 111ZZ, 11X1Z, \dots\} \\
\mathcal{E}_{BP5} &= 1111(XZ) \text{ anti-commutes with } F_3 = \{1111X, 1111Z, 111XX, 111XZ, \dots\} \\
\mathcal{E}_{P5} &= 1111Z \text{ anti-commutes with } F_4 = \{1111X, 111XX, 111ZX, 11X1X, \dots\} \\
&\vdots \\
\mathcal{E}_{P1} &= Z1111 \text{ anti-commutes with } F_{15} = \{X1111, X111X, X111Z, X11X1, \dots\}
\end{aligned} \tag{14.57}$$

This gives us sets of tensor products of Pauli operators, $\{F_1, F_2, \dots, F_{15}\}$, that anti-commute with the different error operators. The required stabilizer will then be a *minimum hitting set* of the sets $\{F_1, F_2, \dots, F_{15}\}$. By minimum hitting set we mean that the desired stabilizer $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$ is the smallest set that intersects with at least one element in every set F_1, F_2, \dots, F_{15} . Given the results of a computer

search shown in (14.57) a minimal hitting set is \mathcal{S} is found to require only four tensor products, $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$, where:

$$\mathcal{S}_1 = X \otimes X \otimes Z \otimes X \otimes 1$$

$$\mathcal{S}_2 = X \otimes Z \otimes X \otimes 1 \otimes X$$

$$\mathcal{S}_3 = Z \otimes 1 \otimes X \otimes X \otimes Z$$

$$\mathcal{S}_4 = Z \otimes X \otimes 1 \otimes Z \otimes X$$

which coincides with the stabilizer we picked to generate codewords that match those used in the 5-qubit Laflamme-Miquel-Paz-Zurek code.

The pattern of anti-commutativity between each error operator $\mathcal{E}_\alpha \in \mathcal{E}$ and the elements of the stabilizer, $\mathcal{S}_i \in \mathcal{S}$, is shown in Table 14.6. In the table a check mark signifies \mathcal{E}_α and \mathcal{S}_i anti-commute whereas a cross signifies they do not. As you can see, each error operator anti-commutes with at least one element of the stabilizer. Moreover, the pattern of anti-commutativity is unique to each error operator. We can exploit this property to associate each error type with a different error syndrome.

So to sum up, we can either pick a stabilizer as a random finite Abelian sub-group of the Pauli group and then see whatever errors it protects against. Alternatively, we can fix the set of errors we want to protect against and use them to induce an

Table 14.6 Two operators anti-commute when $\{A, B\} = A \cdot B + B \cdot A = 0$. Each error operator, $\mathcal{E}_{\text{None}}$, \mathcal{E}_{B5} , \mathcal{E}_{BP5} , ... etc., describing a single error amongst five qubits, anti-commutes with at least one element of the stabilizer $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$. Furthermore, the pattern of anti-commutativity is unique to each operator. This property can be exploited to associate each type of error with a unique error syndrome

Error type	Error operator	$\{\mathcal{E}_\alpha, \mathcal{S}_1\} = 0?$	$\{\mathcal{E}_\alpha, \mathcal{S}_2\} = 0?$	$\{\mathcal{E}_\alpha, \mathcal{S}_3\} = 0?$	$\{\mathcal{E}_\alpha, \mathcal{S}_4\} = 0?$
$\mathcal{E}_{\text{None}}$	$1 \otimes 1 \otimes 1 \otimes 1 \otimes 1$	×	×	×	×
\mathcal{E}_{B5}	$1 \otimes 1 \otimes 1 \otimes 1 \otimes X$	×	×	✓	✗
\mathcal{E}_{BP5}	$1 \otimes 1 \otimes 1 \otimes 1 \otimes X \cdot Z$	✗	✓	✓	✓
\mathcal{E}_{P5}	$1 \otimes 1 \otimes 1 \otimes 1 \otimes Z$	✗	✓	✗	✓
\mathcal{E}_{B4}	$1 \otimes 1 \otimes 1 \otimes X \otimes 1$	✗	✗	✗	✓
\mathcal{E}_{BP4}	$1 \otimes 1 \otimes 1 \otimes X \cdot Z \otimes 1$	✓	✗	✓	✓
\mathcal{E}_{P4}	$1 \otimes 1 \otimes 1 \otimes Z \otimes 1$	✓	✗	✓	✗
\mathcal{E}_{B3}	$1 \otimes 1 \otimes X \otimes 1 \otimes 1$	✓	✗	✗	✗
\mathcal{E}_{BP3}	$1 \otimes 1 \otimes X \cdot Z \otimes 1 \otimes 1$	✓	✓	✓	✗
\mathcal{E}_{P3}	$1 \otimes 1 \otimes Z \otimes 1 \otimes 1$	✗	✓	✓	✗
\mathcal{E}_{B2}	$1 \otimes X \otimes 1 \otimes 1 \otimes 1$	✗	✓	✗	✗
\mathcal{E}_{BP2}	$1 \otimes X \cdot Z \otimes 1 \otimes 1 \otimes 1$	✓	✓	✗	✓
\mathcal{E}_{P2}	$1 \otimes Z \otimes 1 \otimes 1 \otimes 1$	✓	✗	✗	✓
\mathcal{E}_{B1}	$X \otimes 1 \otimes 1 \otimes 1 \otimes 1$	✗	✗	✓	✓
\mathcal{E}_{BP1}	$X \cdot Z \otimes 1 \otimes 1 \otimes 1 \otimes 1$	✓	✓	✓	✓
\mathcal{E}_{P1}	$Z \otimes 1 \otimes 1 \otimes 1 \otimes 1$	✓	✓	✗	✗

acceptable stabilizer as the solution to a minimum hitting set problem. Given the stabilizer, the quantum codewords it stabilizes, and the errors it protects against, can be obtained automatically.

Next we see how the stabilizer formalism also simplifies the search for the required encoding and decoding circuits, and allows us to perform error correction while staying entirely within the encoded basis.

14.6.7 Stabilizer-Based Error Correction: The Encoding Step

To protect a qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ we encode it into the state $|\psi\rangle_L = \alpha|0\rangle_L + \beta|1\rangle_L$ using the Laflamme-Miquel-Paz-Zurek encoding circuit shown in Fig. 14.3. Once in encoded form the logical qubit is protected against a single error amongst any of the five qubits.

14.6.8 Stabilizer-Based Error Correction: Introduction of the Error

We model the introduction of an error on our encoded state as the application of one of the error operators, $\mathcal{E}_\alpha \in \mathcal{E}$, our stabilizer is known to correct.

14.6.9 Stabilizer-Based Error Correction: Error Diagnosis & Recovery

We use the same quantum circuit to perform the actual error correction as we use to find the encoded basis states $|0\rangle_L$ and $|1\rangle_L$. As illustrated in Fig. 14.8, we imagine that we have an encoded state entering the circuit, which has been afflicted with an unknown error, $\mathcal{E}_\alpha \in \mathcal{E}$, where:

$$\mathcal{E} = \{\mathcal{E}_{\text{None}}, \mathcal{E}_{B5}, \mathcal{E}_{P5}, \mathcal{E}_{BP5}, \mathcal{E}_{B4}, \mathcal{E}_{P4}, \mathcal{E}_{BP4}, \mathcal{E}_{B3}, \mathcal{E}_{P3}, \mathcal{E}_{BP3}, \mathcal{E}_{B2}, \mathcal{E}_{P2}, \mathcal{E}_{BP2}, \mathcal{E}_{B1}, \mathcal{E}_{P1}, \mathcal{E}_{BP1}\} \quad (14.58)$$

which includes the possibly no error whatsoever. The circuit essentially measures the eigenvalue of each element of the stabilizer, $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$ with respect to the incoming state $\mathcal{E}_\alpha|\psi\rangle_L$. If the stabilizer element commutes with the (unknown) error operator, the eigenvalue will be $+1$. But if the error-operator anti-commutes with the element of the stabilizer, the eigenvalue will be -1 . Thus, the pattern of anti-commutativity can therefore be used as an error syndrome $a\ b\ c\ d$, which can diagnose that the error afflicting $|\psi\rangle_L$ is \mathcal{E}_α and hence the appropriate corrective action needed to restore the state is \mathcal{E}_α^{-1} .

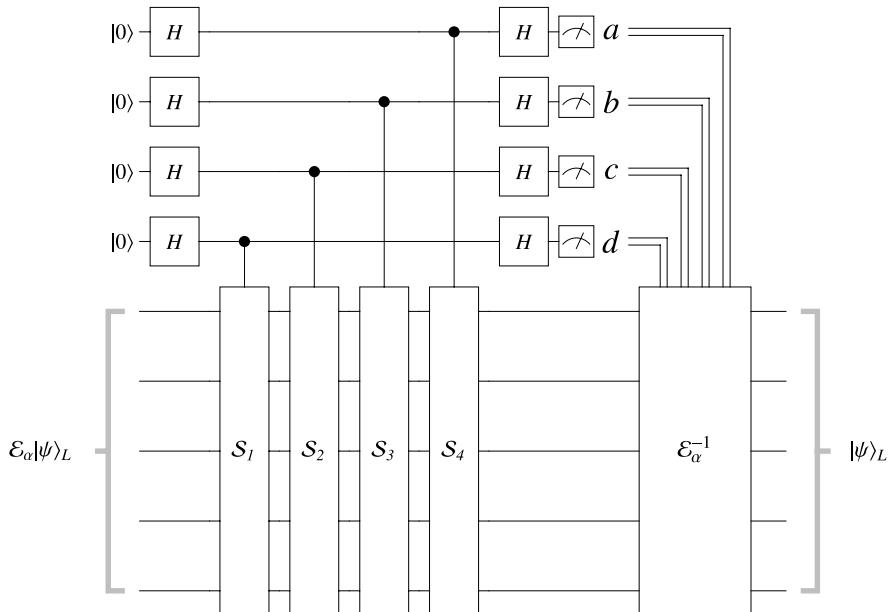


Fig. 14.8 Quantum circuit for error recovery based on the stabilizer formalism. An error-afflicted encoded state, $\mathcal{E}_\alpha |\psi\rangle_L$, enters the circuit. At this point we have no idea what error has occurred, i.e., we do not know \mathcal{E}_α . To discover the identity of \mathcal{E}_α , we measure the eigenvalue of each element of the stabilizer with respect to the state fed into the circuit. If the error-afflicted state commutes with the element of the stabilizer, the eigenvalue is $+1$. If, on the other hand, the error-afflicted state anti-commutes with the element of the stabilizer, the eigenvalue is -1 . Hence, the pattern of anti-commutativity revealed in the results $a \ b \ c \ d$, provides sufficient information diagnose what error occurred. That is, after these measurements we now know \mathcal{E}_α . It is then straightforward to predict the error-restoration operation, \mathcal{E}_α^{-1} , needed to restore the encoded qubit to its pristine, and still encoded, state $|\psi\rangle_L$. Note that during this error-diagnosis and error-correction process the single logical qubit remains in its encoded basis throughout. Hence, the stabilizer formalism is especially good because we never need to re-expose the logical qubit in an unprotected form at any time

Notice that, whereas in the original Laflamme-Miquel-Paz-Zurek scheme we periodically decoded the encoded state back to a single logical qubit, and thereby exposed it to an uncorrectable error, in the stabilizer formalism once the state has been encoded it is never re-exposed as a single logical qubit. Rather, in the stabilizer formalism, the whole error-correction procedure takes place within the encoded subspace. This is a very smart thing to do because it avoids having to periodically re-expose the logical qubit in order to error correct it.

14.6.10 Stabilizers for Other Codes

An $\llbracket n, k, d \rrbracket$ quantum code (with square parentheses and a lowercase letter k) is a special notation for quantum stabilizer codes. Such as code uses n physical qubits

to encode $k < n$ logical qubits within a $K = 2^k$ -dimensional codespace and has minimum distance d . Hence, the number of 1-qubit changes needed to get from one codeword to another is at least d , which means that the code can correct up to $t = \lfloor \frac{d-1}{2} \rfloor$ single qubit errors.

The 9-qubit Shor, the 7-qubit Steane, and the 5-qubit Laflamme-Miquel-Paz-Zurek codes are respectively $\llbracket 9, 1, 3 \rrbracket$, $\llbracket 7, 1, 3 \rrbracket$, and $\llbracket 5, 1, 3 \rrbracket$ stabilizer codes, which can each correct at most $t = \lfloor (3-1)/2 \rfloor = 1$ error within their respective blocks of 9, 7, and 5 physical qubits.

A stabilizer for Shor's 9-qubit code is [141]:

$$\begin{aligned}\mathcal{S}_1 &= ZZ\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1} \\ \mathcal{S}_2 &= Z\mathbb{1}Z\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1} \\ \mathcal{S}_3 &= \mathbb{1}\mathbb{1}\mathbb{1}ZZ\mathbb{1}\mathbb{1}\mathbb{1} \\ \mathcal{S}_4 &= \mathbb{1}\mathbb{1}\mathbb{1}Z\mathbb{1}Z\mathbb{1}\mathbb{1} \\ \mathcal{S}_5 &= \mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}ZZ\mathbb{1} \\ \mathcal{S}_6 &= \mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}Z\mathbb{1}Z \\ \mathcal{S}_7 &= XXXXXX\mathbb{1}\mathbb{1}\mathbb{1} \\ \mathcal{S}_8 &= XXX\mathbb{1}\mathbb{1}\mathbb{1}XXX\end{aligned}\tag{14.59}$$

and one for Steane's 7-qubit code is [141]:

$$\begin{aligned}\mathcal{S}_1 &= \mathbb{1}\mathbb{1}\mathbb{1}XXXX \\ \mathcal{S}_2 &= X\mathbb{1}X\mathbb{1}X\mathbb{1}X \\ \mathcal{S}_3 &= \mathbb{1}XX\mathbb{1}\mathbb{1}XX \\ \mathcal{S}_4 &= \mathbb{1}\mathbb{1}\mathbb{1}ZZZZ \\ \mathcal{S}_5 &= Z\mathbb{1}Z\mathbb{1}Z\mathbb{1}Z \\ \mathcal{S}_6 &= \mathbb{1}ZZ\mathbb{1}\mathbb{1}ZZ\end{aligned}\tag{14.60}$$

Notice that the stabilizers for the Shor and Steane codes have only X 's or only Z 's within their respective stabilizer elements, making them so-called Calderbank-Shor-Steane ("CSS") codes. By comparison, the 5-qubit code is also a stabilizer code but it is not a CSS code because some of its stabilizer elements mix Z 's and X 's together.

14.7 Bounds on Quantum Error Correcting Codes

One can gain an intuition for the tradeoffs between the number of physical qubits (n), the number of logical qubits (k), and the maximum number of correctable errors $t = \lfloor \frac{d-1}{2} \rfloor$ by finding those tuples of values of n , k , and d that simultaneously satisfy three important bounds on quantum codes: the quantum Hamming, Gilbert-Varshamov and Singleton bounds.

14.7.1 Quantum Hamming Bound

The quantum Hamming bound was discovered by Artur Ekert and Chiara Macchiavello [168]. It places an upper bound on the number of codewords we can have in a quantum code if the code has to be guaranteed to be able to encode k logical qubits in n physical qubits and protect against up to t single qubit errors.

The bound is obtained via a counting argument on the number of buggy states in comparison to the number of states we can fit in a Hilbert space of dimension 2^n . The argument goes as follows. If a code is to correct up to t errors, then each codeword must be able to tolerate up to t errors and yet still be distinct from every other codeword and every other potentially corrupted codeword. We can therefore imagine each codeword as being surrounded by a “cloud” of buggy states that have anywhere from zero to t errors in them. All these states need to be distinct from the other buggy states in similar clouds around all the other codewords. All these buggy codewords have to fit within our Hilbert space of n qubits.

Making this argument more quantitative, consider a single codeword of length n qubits. We can introduce i errors to this codeword by picking a particular subset of i out of n qubits, and assign single qubit errors to those qubits in all possible ways. There are $\binom{n}{i}$ ways to pick a particular subset of i qubit locations, and there are three types of error (X , Z , and $(X \cdot Z)$) possible per location. Hence, there are $3^i \binom{n}{i}$ states describing i errors to our codeword. But we want to protect against up to t errors. Therefore, we can think of each codeword as being surrounded by a cloud of $\sum_{i=0}^t 3^i \binom{n}{i}$ “buggy” codewords. There are a total of 2^k such codewords. And the union of all these clouds of states needs to fit within the dimension of our n -qubit Hilbert space. Hence, we arrive at the quantum Hamming bound which places an upper bound on the number of codewords (2^k) or equivalently the number of logical qubits (k), that we can have in a quantum code that uses n physical qubits. Hence, for a non-degenerate $\llbracket n, k, d \rrbracket$ code we must have:

$$2^k \sum_{i=0}^t 3^i \binom{n}{i} \leq 2^n \quad (14.61)$$

where $t = \lfloor \frac{d-1}{2} \rfloor$. Note that this bound gives a *necessary* condition for the existence of a quantum code and is really no more than a generalization of the argument we gave in Sect. 14.4.1 to deduce the allowed relationships between n , k and d for the 5-qubit Laflamme-Miquel-Paz-Zurek code.

14.7.2 Quantum Singleton Bound

The quantum Singleton bound was discovered by Raymond Laflamme and Manny Knill [292]. This states that if a pure or impure $\llbracket n, k, d \rrbracket$ code exists then:

$$n - k \geq 4 \lfloor (d - 1)/2 \rfloor \quad (14.62)$$

The quantum Singleton bound for pure codes was strengthened, slightly, by Calderbank, Rains, Shor, and Sloane [95] to:

$$n - k \geq 2(d - 1) \quad (14.63)$$

This reduces to the Laflamme and Knill formula when d is odd, but is slightly stronger when d is even. It is also a necessary condition for the existence of a quantum code.

14.7.3 Quantum Gilbert-Varshamov Bound

The quantum Gilbert-Varshamov bound was discovered by Artur Ekert and Chiara Macchiavello [168]. It states that for an $\llbracket n, k, d \rrbracket$ code:

$$2^k \sum_{i=0}^{2t} 3^i \binom{n}{i} \geq 2^n \quad (14.64)$$

where the number of errors that can be corrected, t , is given by $t = \lfloor \frac{d-1}{2} \rfloor$. This bound gives a *sufficient* condition on the existence of a code but it is not necessary. The bound states that the number of codewords times the number of buggy codewords reachable in up to $2t$ errors must not be smaller than the dimension of the Hilbert space for n physical qubits.

14.7.4 Predicting Upper and Lower Bounds on Additive Codes

The quantum Hamming, Singleton, and Gilbert-Varshamov bounds can be used to find upper and lower bounds on the minimum distance d of feasible quantum codes. This in turn bounds the maximum possible number of errors such codes can correct, t , because we have $t = \lfloor \frac{d-1}{2} \rfloor$. Note that the quantum Hamming and Singleton bounds give us an upper bound on d , whereas the quantum Gilbert-Varshamov bound gives us a (loose) lower bound on d . Nevertheless, the bounds are tight enough that we can use them to gain a rough intuition for the tradeoffs between the number of physical qubits, the number of logical qubits, the minimum distance and hence the maximum number of correctable errors.

Table 14.7 shows the approximate upper and lower bounds on the minimal distance d in any $\llbracket n, k, d \rrbracket$ pure quantum error-correcting code as constrained by the quantum Hamming, quantum Singleton and quantum Gilbert-Varshamov bounds. The lower bounds are obtained from the quantum Gilbert-Varshamov inequality (14.64) and the upper bound is obtained from the lesser of the quantum Hamming (14.61) and quantum Singleton bound for pure codes (14.63). In Table 14.7 when a range of values is given these are lower and upper bounds on d . As the quantum Hamming and quantum Singleton bound provide a *necessary* condition on d ,

Table 14.7 Approximate bounds on the highest achievable minimal distance d for any pure $[[n, k, d]]$ quantum error-correcting code. The upper bound is obtained by finding the smallest value of distance d , for given values of n and k , able to satisfy the quantum Hamming bound (14.61) and the quantum Singleton bound (for pure codes) (14.63) simultaneously. These provide a necessary upper bound on d for the existence of an $[[n, k, d]]$ code. Hence the stated upper bound on d is a hard constraint. The lower bound is obtained by finding the largest value of d , for given values of n and k , able to satisfy the quantum Gilbert–Varshamov bound. The latter bound is sufficient to guarantee the existence of an $[[n, k, d]]$ code but it is not necessary. Hence, the lower bound given is loose.

$n \setminus k$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
3	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	3	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	3	3	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	3-4	3	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	3-4	3-4	3	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	3-4	3-4	3-4	3	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	5	3-4	3-4	3-4	3	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	5-6	5	3-4	3-4	3-4	3	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
11	5-6	5-6	3-4	3-4	3-4	3-4	2	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	
12	5-6	5-6	5-6	3-4	3-4	3-4	3-4	2	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	
13	5-6	5-6	5-6	5-6	3-4	3-4	3-4	3-4	2	2	2	2	1	1	0	0	0	0	0	0	0	0	0	
14	5-8	5-6	5-6	5-6	5-6	3-4	3-4	3-4	3-4	2	2	2	2	1	1	0	0	0	0	0	0	0	0	
15	5-8	5-8	5-6	5-6	5-6	5-6	5-6	3-4	3-4	3-4	3-4	2	2	2	2	1	1	0	0	0	0	0	0	
16	5-8	5-8	5-8	5-6	5-6	5-6	3-4	3-4	3-4	3-4	3-4	2	2	2	2	1	1	0	0	0	0	0	0	
17	5-8	5-8	5-8	5-6	5-6	5-6	5-6	3-4	3-4	3-4	3-4	3-4	2	2	2	2	1	1	0	0	0	0	0	
18	5-8	5-8	5-8	5-8	5-8	5-6	5-6	5-6	5-6	3-4	3-4	3-4	3-4	3-4	2	2	2	2	1	1	0	0	0	
19	7-10	5-8	5-8	5-8	5-8	5-8	5-8	5-8	5-8	5-6	5-6	5-6	5-6	3-4	3-4	3-4	3-4	2	2	2	1	1	0	
20	7-10	7-10	5-8	5-8	5-8	5-8	5-8	5-8	5-8	5-6	5-6	5-6	5-6	3-4	3-4	3-4	3-4	2	2	2	1	1	0	

Table 14.7 (Continued)

$n \setminus k$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
21	7-10	7-10	7-10	5-8	5-8	5-8	5-6	5-6	5-6	5-6	5-6	5-6	3-4	3-4	3-4	2	2	2	1	1	0	0	0	
22	7-10	7-10	7-10	5-8	5-8	5-8	5-8	5-6	5-6	5-6	5-6	3-4	3-4	3-4	2	2	2	2	1	1	0			
23	7-10	7-10	7-10	7-10	5-8	5-8	5-8	5-8	5-6	5-6	5-6	5-6	3-4	3-4	3-4	2	2	2	2	1	1	1		
24	7-12	7-10	7-10	7-10	7-10	7-10	5-8	5-8	5-8	5-8	5-6	5-6	5-6	3-4	3-4	3-4	2	2	2	2	2	1		
25	7-12	7-12	7-10	7-10	7-10	5-8	5-8	5-8	5-8	5-8	5-6	5-6	5-6	3-4	3-4	3-4	3-4	2	2	2	2	2		
26	7-12	7-12	7-10	7-10	7-10	7-10	5-8	5-8	5-8	5-8	5-6	5-6	5-6	5-6	3-4	3-4	3-4	3-4	2	2	2	2		
27	7-12	7-12	7-12	7-10	7-10	7-10	7-10	5-8	5-8	5-8	5-8	5-6	5-6	5-6	5-6	3-4	3-4	3-4	3-4	2	2	2		
28	7-12	7-12	7-12	7-10	7-10	7-10	7-10	5-8	5-8	5-8	5-8	5-6	5-6	5-6	5-6	3-4	3-4	3-4	3-4	3-4	2	2		
29	9-14	7-12	7-12	7-12	7-12	7-12	7-10	7-10	7-10	7-10	5-8	5-8	5-8	5-8	5-6	5-6	5-6	5-6	3-4	3-4	3-4	3-4	2	
30	9-14	9-14	7-12	7-12	7-12	7-12	7-10	7-10	7-10	7-10	5-8	5-8	5-8	5-8	5-6	5-6	5-6	5-6	5-6	3-4	3-4	3-4	3-4	

Fig. 14.9 Plot of the approximate upper bounds on minimal distance d of an $\llbracket n, k, d \rrbracket$ quantum error correcting code for $1 \leq n \leq 30$ and $0 \leq k \leq 28$. The data correspond to the upper bounds given in Table 14.7, which come from finding the largest value of d , for given values of n and k , such that the quantum Hamming bound, and quantum Singleton bound are satisfied simultaneously. This is a *necessary* condition on the existence of the corresponding $\llbracket n, k, d \rrbracket$ code, so this upper bound on the minimal distance cannot be beaten

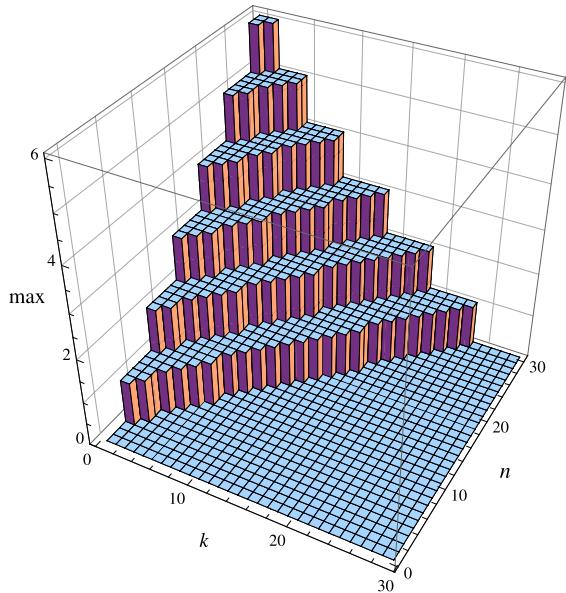
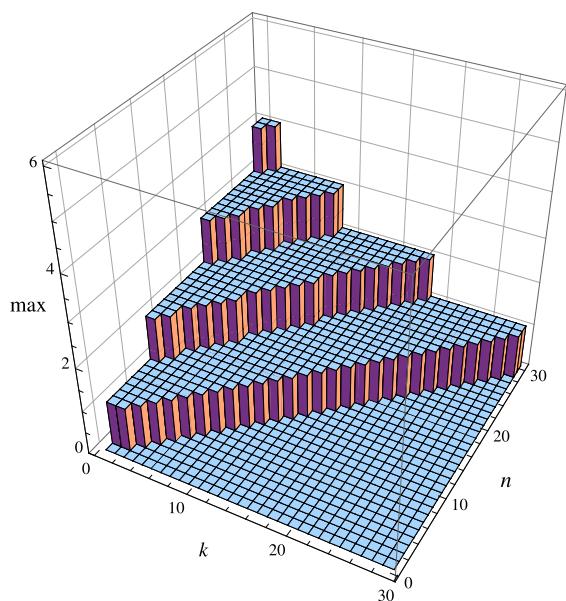


Fig. 14.10 Plot of the approximate (very loose) lower bounds on minimal distance d of an $\llbracket n, k, d \rrbracket$ quantum error correcting code for $1 \leq n \leq 30$ and $0 \leq k \leq 28$. The data correspond to the lower bounds given in Table 14.7, which come from finding the smallest value of d , for given values of n and k , such that the quantum Gilbert-Varshamov bound is satisfied. This is only a *sufficient* condition on the existence of the corresponding $\llbracket n, k, d \rrbracket$ code, so this lower bound on the minimal distance can be beaten



whereas the quantum Gilbert-Varshamov bound provides a *sufficient* condition on d , the Hamming and Singleton bounds take precedence on upper bounding d .

The upper and lower bound data on predicted minimum distance in Table 14.7 is visualized in Figs. 14.9 and 14.10.

14.7.5 Tightest Proven Upper and Lower Bounds on Additive Codes

It is naturally to ask whether it is possible that more efficient codes could exist, i.e., codes that can correct more than one error per block. Indeed they can, but the complexity of the quantum circuits needed to implement them grows rapidly.

Using far more sophisticated methods, one can obtain tighter upper bounds, as well as proper lower bounds, on the highest achievable minimal distance d of any $\llbracket n, k, d \rrbracket$ quantum error-correcting code (see Table 14.8). Comparing Table 14.8 with (the much more easily obtained) Table 14.7 shows the estimated bounds on d from necessary and sufficient conditions are pretty good.

Plots of the tightest proven upper and lower bounds on minimum distance are shown in Figs. 14.11 and 14.12.

14.8 Non-additive (Non-stabilizer) Quantum Codes

The original 9-qubit Shor, 7-qubit Steane, and the 5-qubit Laflamme-Miquel-Paz-Zurek codes were all additive (stabilizer) codes. However, it is possible to have codes that possess a fundamentally different structure than the stabilizer codes. These so-called “non-additive” codes may be harder to find, but they are potentially more efficient than the stabilizer codes.

The first non-additive quantum error correcting code, that was provably better than an additive (stabilizer) code was the $\llbracket 5, 6, 2 \rrbracket$ code discovered using numerical techniques by E.M. Rains, R.H. Hardin, P.W. Shor, and N.J.A. Sloane in 1997 [407]. This generalizes to a family of codes of the form $\llbracket 2n + 1, 3 \times 2^{2n-3}, 2 \rrbracket$. Thomas Beth and Markus Grassl showed that the $\llbracket 5, 6, 2 \rrbracket$ code could be obtained from union of additive codes [212]. That is, if \mathcal{C}_1 and \mathcal{C}_2 are respectively $\llbracket n, K_1, d_1 \rrbracket$ and $\llbracket n, K_2, d_2 \rrbracket$ quantum codes, the union of these codes is an $\llbracket n, K_1 + K_2, \min(d_1, d_2) \rrbracket$ quantum code such that the set of errors the new code can correct is the intersection of the sets of errors the old codes could correct. Note that, whereas the dimension for the codespace of an additive code is always a power of two, the dimension of the codespace of a non-additive code built from the union of two additive codes need not be a power of two.

The first non-additive code found that can outperform the *optimal* $\llbracket 5, 1, 3 \rrbracket$ (Laflamme-Miquel-Paz-Zurek) stabilizer code was the $\llbracket 9, 12, 3 \rrbracket$ non-additive code found by Sixia Yu, Qing Chen, C. Lai, and C. Oh [554].

14.9 Fault-Tolerant Quantum Error Correcting Codes

The discussions of quantum error correcting codes given in the preceding sections have made implicit assumptions about where and when errors occur. For example,

Table 14.8 Known bounds on the highest achievable minimal distance d for any $[\![n, k, d]\!]$ quantum error-correcting code [211]

$n \setminus k$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
3	2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	3	3	2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	4	3	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	3	3	2	2	2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	4	3	3	3	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	4	3	3	3	2	2	2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	4	4	4	3	3	2	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
11	5	5	4	3	3	3	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
12	6	5	4	4	4	3	3	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	
13	5	5	4	4	4	3	3	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	
14	6	5	5	4-5	4	4	3	3	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	
15	6	5	5	5	4	4	4	3	3	3	2	2	2	1	1	0	0	0	0	0	0	0	0	
16	6	6	6	5	5	4-5	4	4	3	3	3	2	2	2	2	1	1	0	0	0	0	0	0	
17	7	7	6	5-6	5	4-5	4	4	4	3	3	3	2	2	2	1	1	0	0	0	0	0	0	
18	8	7	6	5-6	5-6	5	5	4	4	3	3	3	2	2	2	2	1	1	0	0	0	0	0	
19	7	7	6	5-6	5-6	5	4-5	4	4	3-4	3	3	2	2	2	2	1	1	1	0	0	0	0	
20	8	7	6-7	6-7	6	5-6	5-6	4-5	4-5	4	4	3-4	3	3	2	2	2	2	1	1	0	0	0	

Table 14.8 (Continued)

$n \setminus k$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
21	8	7	6-7	6-7	6-7	6	5-6	5-6	4-5	4-5	4	4	3-4	3	3	3	2	2	2	1	1	0	0	
22	8	7-8	6-8	6-7	6-7	6-7	5-6	5-6	4-5	4-5	4	4	3-4	3	3	2	2	2	2	1	1	0		
23	8-9	7-9	7-8	6-8	6-7	6-7	5-6	5-6	4-6	4-5	4-5	4	4	3-4	3	3	2	2	2	1	1	1		
24	8-10	8-9	7-8	7-8	6-8	6-7	6-7	5-7	5-6	5-6	4-5	4-5	4	4	3-4	3	3	2	2	2	2	2	1	
25	8-9	9	7-8	7-8	7-8	7-8	6-7	5-7	5-6	5-6	4-6	4-5	4-5	4	4	3-4	3	3	2	2	2	2	1	
26	8-10	9	8-9	8-9	8	7-8	6-8	6-8	6-7	5-7	5-6	5-6	4-5	4-5	4	4	3-4	3	3	2	2	2	2	
27	9-10	9	9	8-9	8-9	7-8	6-8	6-8	6-7	5-7	5-6	5-6	5	4-5	4-5	4	4	3-4	3	3	2	2	2	
28	10	10	9	8-9	8-9	7-9	6-8	6-8	6-8	6-7	5-7	5-6	5-6	4-5	4	4	4	3-4	3	3	2	2		
29	11	11	10	9-10	8-9	7-9	7-9	6-8	6-8	6-7	5-7	5-6	5-6	4-5	4	4	4	3-4	3	3	2			
30	12	11	10	9-10	8-10	8-9	7-9	7-9	7-9	6-8	6-8	6-7	5-6	5-6	5	4-5	4	4	4	3-4	3	3		

Fig. 14.11 Plot of the known upper bounds on minimal distance d of an $\llbracket n, k, d \rrbracket$ quantum error correcting code for $1 \leq n \leq 30$ and $0 \leq k \leq 28$. The data correspond to the upper bounds given in Table 14.8, which come from Markus Grassl's curated database of code parameters [211]

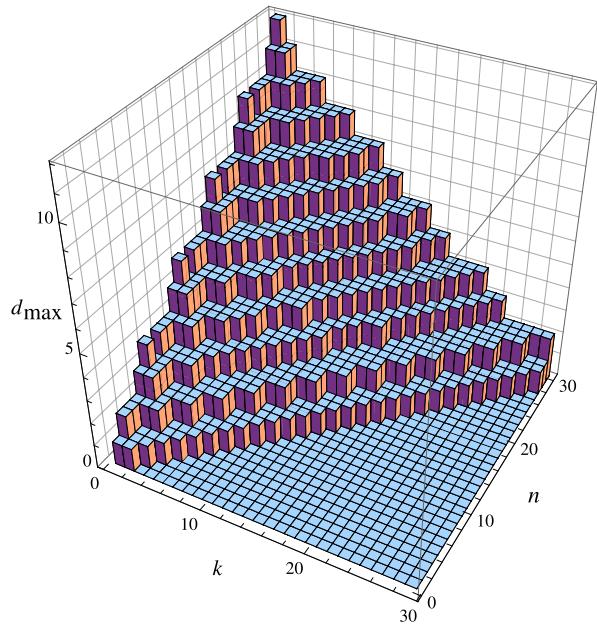
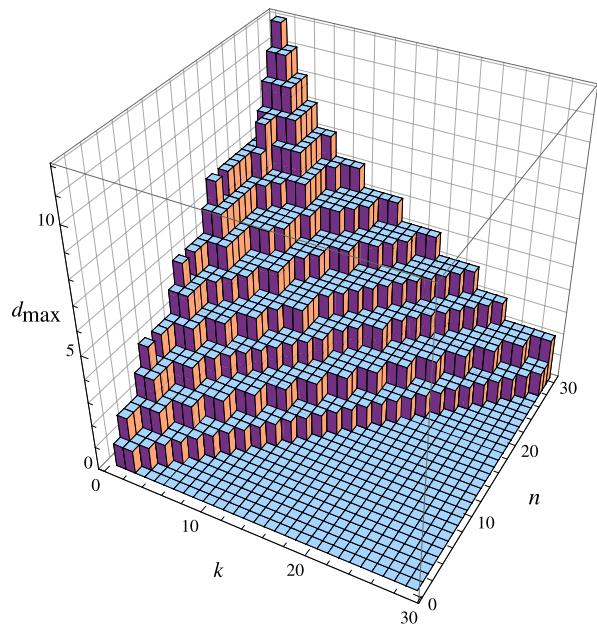


Fig. 14.12 Plot of the known lower bounds on minimal distance d of an $\llbracket n, k, d \rrbracket$ quantum error correcting code for $1 \leq n \leq 30$ and $0 \leq k \leq 28$. The data correspond to the lower bounds given in Table 14.8, which come from Markus Grassl's curated database of code parameters [211]



in the original formulation of the 5-qubit quantum error-correcting code, given in Sect. 14.4, error correction required us to map the encoded (protected) logical qubit back to its unprotected form periodically. Once error-free, the logical qubit would be

re-encoded to protect it again. If the error occurs when the qubit is back in its logical (unprotected) state the logical qubit will be vulnerable to irreversible corruption. For this strategy to work, we must implicitly assume that no errors can arise while the qubit is re-exposed in the unencoded basis. If this assumption holds, we will be able to store the state of a logical qubit indefinitely without error. Unfortunately, such an assumption is clearly unjustified by the physics of the situation. There is no good reason to expect errors should only afflict encoded qubits. However, by using the stabilizer formulation of the 5-qubit code, you will remember that error correction can be performed entirely within the encoded basis, never needing to re-expose the logical qubit to potential uncorrectable errors. Nevertheless, this is still not yet a complete solution, because we don't just want to protect quantum information when in storage, but also during quantum computation itself. This means that we need perform gate operations directly on the encoded data.

The second assumption we need to question is where do the errors occur? So far, we have implicitly assumed that the gates implementing the error correction operations are perfect. But what happens if they are imperfect? Can we error correct a quantum computation using imperfect quantum gates?

These concerns prompted further research into quantum error correcting codes that revealed how to them work even when the underlying error correction hardware is itself imperfect. The result is so-called *fault-tolerant* quantum error correction [209, 270, 400, 457].

A quantum circuit is deemed “fault-tolerant” when it can be made to output the correct result even though errors arise during its operation. John Preskill of the California Institute of Technology has identified five principles of quantum circuit design, distilled from Peter Shor’s original paper on fault-tolerant quantum computation [457], which will make for fault-tolerant quantum circuits [399].

1. *Don’t use the same ancilla qubit twice.* The intuition behind this principle is that if an ancilla qubit becomes corrupted, we want to limit the damage it can do by limiting the number of other gate operations that rely on the same ancilla. Thus, examples of good and bad quantum circuit structures that use ancillae are shown in Fig. 14.13. Error propagation in quantum circuits is much more problematic than in classical circuits because in controlled quantum gates errors can propagate in both directions, i.e., from control qubits to targets (as happens classically) and from target

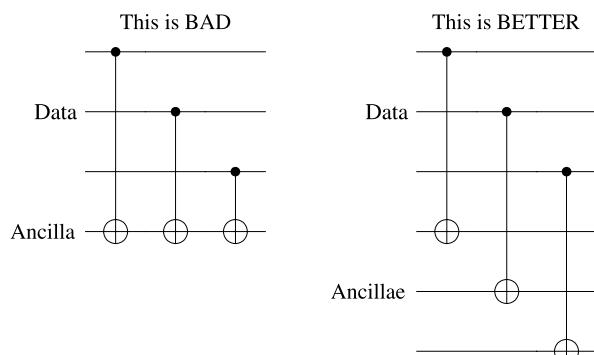


Fig. 14.13 The first principle of fault-tolerant quantum computing: “do not use the same ancilla twice.” This suppresses correlated error propagation from bad ancillae

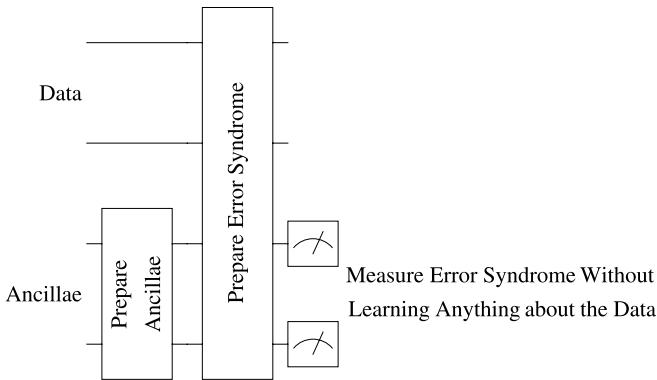


Fig. 14.14 The second principle of fault-tolerant quantum computing: “Copy the errors not the data.” The ancillae measurements must not extract any information about the logical state of the qubits being protected, only the errors that have afflicted them

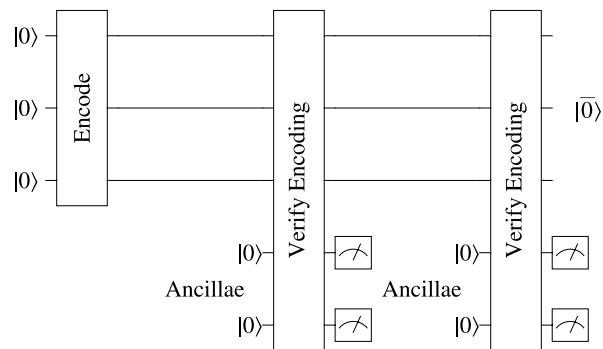
qubits to controls (which does not happen classically). So quantum controlled gates are especially susceptible to the spread of error.

2. *Error syndrome measurements should reveal the error but not the data.* We need to be careful to prepare the ancilla qubit in such a way that when we measure the ancilla to obtain an error-syndrome we do not learn anything about the state we seek to protect, but only an error that may have afflicted it. This requires us to prepare the ancilla in a special entangled state prior to linking it to the state we wish to protect. A diagram of this is shown in Fig. 14.14.

3. *Verify when encoding a known quantum state.* The potential for corruption is greatest when qubits are exposed in their raw state before they been protected using some quantum error-correcting encoding. However, whenever we do know the complete description of the quantum state with which we are dealing, and we do know the operation we intend to perform on it, we have an opportunity to verify that we synthesized the correct state before using it further. This situation can arise, e.g., when we start off with some ancillae qubits in a known state, and we entangle them in some prescribed way. In such circumstances it is worth taking the time to verify the entangled state is correct before making use of it in subsequent quantum computations. For example, if our intent is to encode three physical qubits $|0\rangle|0\rangle|0\rangle$ into some encoded block of three qubits, which we will call $|\bar{0}\rangle$, we might perform a test to convince ourselves that we synthesized the block $|\bar{0}\rangle$ correctly before using it in subsequent quantum computations. This idea is illustrated in Fig. 14.15.

4. *Repeat operations.* Figure 14.15 also illustrates a fourth principle of fault-tolerant quantum computation. Just because we have verified the encoding of a state once does not mean necessarily that it is correct as our error syndrome measurement could be faulty. It would be just as disastrous to correct an error, or non-error, in the wrong way as it would to miss an error in the first place. However, by *repeating* measurements, we can increase our confidence that the error syndrome is actually what we think it is. Thus, repeating quantum measurements to a good habit as illustrated in Fig. 14.15.

Fig. 14.15 The third principle of fault-tolerant quantum computing: “Verify when encoding a known quantum state.” A known state should be verified (perhaps repeatedly) before being deemed fit for use



5. *Stay in the encoded basis.* The 9-qubit, 7-qubit and 5-qubit quantum codes we described above are, as presented thus far, geared towards error-correcting qubits while they are inactive, i.e., merely stored in memory. However, typically, we want to do more than merely *store* qubits—we want to *compute* with them. That is, we anticipate needing to apply quantum logic gates in order to perform a purposeful quantum computation. However, the theory of quantum error correcting codes outlined above, does not describe how to perform quantum gates on the encoded qubits. Instead, when one wants to perform a quantum gate, one would need to map the encoded qubits back to the logical basis, apply the quantum gate, and then re-encode the result back into the encoded basis. Such a strategy is at least cumbersome, and worse, periodically exposes the qubits to corruption as the quantum gates are being applied. To circumvent this problem Wojciech Zurek and Raymond Laflamme, and Peter Shor independently devised a schemes for performing quantum gate operations on the encoded qubits *directly*, without removing them from the safety of the encoded basis [457, 566].

Obeying these five principles of fault-tolerant quantum circuit design will help to ensure that a quantum computer will operate reliably.

14.9.1 Concatenated Codes and the Threshold Theorem

So far, we have seen that quantum error codes are possible in principle, and codes that can correct up an arbitrary number of errors exist. Moreover, we have seen it is possible to use such codes in an intelligent way by employing a fault-tolerant architecture. Unfortunately, there is still a problem. Although it is indeed possible to devise more complex quantum codes that can correct up to t errors in a block, the complexity of the quantum circuits needed to implement such codes rises rapidly. In fact, before long, we have to use so many gates that the probability of making an error within the error correcting circuitry becomes higher than the probability of the original error. So merely increasing the code complexity to correct for more errors per block is not necessarily the best way to improve reliability.

Julia Kempe has provided the following intuitive analysis of the tradeoffs between how many errors a quantum error correcting code can correct and the complexity of its required quantum circuitry [270]. If the original probability of failure per gate operation or per measurement is ϵ then in the t -error resilient code the failure rate would change to ϵ^{t+1} , which is good. But the price we pay is that the number of gates needed in the error correction circuitry grows too, typically as some polynomial in t , t^a , with $a > 1$. So overall, the probability of having $t + 1$ errors occur before error correction has completed grows as $(t^a \epsilon)^{t+1}$. This expression is minimized when $t = c\epsilon^{-\frac{1}{a}}$, for some constant c , and the value of the failure probability is then $p_{\text{fail}} \geq \exp(-ca\epsilon^{-\frac{1}{a}})$. If we repeat t -error resilient error correction N times, the failure probability will therefore become $Np_{\text{fail}} = N \exp(-ca\epsilon^{-\frac{1}{a}}) = \exp(-ca(\log N)\epsilon^{-\frac{1}{a}})$. For this overall failure probability to be much less than 1, we will therefore need ϵ to scale as $1/(\log N)^a$. In other words, the longer the computation, the smaller ϵ needs to be. Unfortunately, this is not practical. Instead we need an error correction scheme that allows the error probability per gate operation to be held constant whilst allowing longer and longer computations are performed reliably.

An alternative way to improve the reliability is to *concatenate* the simpler quantum codes we know about [7, 11]. The idea is that each logical qubit is encoded in n physical qubits (to make a “level-1” encoding), and each of these n physical qubits are themselves encoded in n other physical qubits (to make a “level-2” encoding), and so on. The number of levels of concatenation can be chosen so as to achieve any desired probability in the correctness of the final result. Figure 14.16 shows a schematic illustrating the basic idea.

It is fairly involved to calculate the exact effects of concatenation on the overall reliability of the circuit, although people have done so for different physical schemes and quantum computer architectures [31, 291, 350, 479, 484, 486]. In part, this is

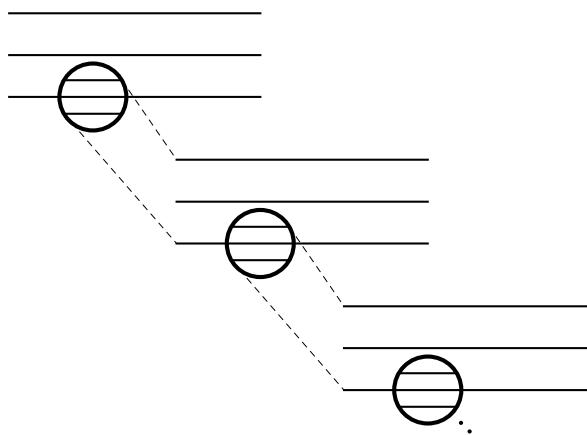


Fig. 14.16 Schematic view of concatenated coding: each qubit is encoded in several qubits, which are each encoded in several qubits, which are each encoded in several qubits, etc.

because the details of the calculation depend upon many factors such as the actual code used, the error model assumed, the degree to which fault-tolerant design principles have been followed, the architectural assumptions made, and the extent to which opportunities for gate-parallelism have been exploited. However, a simple back-of-the-envelope argument is sufficient to convey the main idea, that the use of concatenation is beneficial provided the error probability per qubit per gate is less than a certain threshold.

Think of it this way. Suppose we are using a code that encodes each logical qubit in n physical qubits. The quantum error-correcting codes we looked at earlier can correct a single arbitrary error (bit-flip, phase-flip, or joint bit-flip and phase-flip) in a coding block. So for the logical qubit to be in error at the end of some error-correcting cycle we will have had to have had *two* or more errors introduced into a block. If the probability of an error per qubit per gate operation is p , then (since all the 1-qubit errors are fixable), by using the code the error probability becomes $p_{\text{fail}}^{\text{level-1}} = cp^2$ where c counts the number of ways pairs of errors can be inserted amongst the physical qubits in a coding block.

Now concatenate this process. For each physical qubit in the coding block, imagine using the same code to encode it into n more physical qubits. Now n^2 physical qubits are involved in encoding one logical qubit. What will it take for our logical qubit to be in error now? Let us call this the level-2 encoding. We have $p_{\text{fail}}^{\text{level-2}} = c(p_{\text{fail}}^{\text{level-1}})^2 = c^3 p^4$.

Repeating concatenation steps in this fashion we can write down the error probability as a function of the number of levels of concatenation as follows:

$$\begin{aligned} p_{\text{fail}}^{\text{level-1}} &= cp^2 \\ p_{\text{fail}}^{\text{level-2}} &= c(p_{\text{fail}}^{\text{level-1}})^2 = c^3 p^4 \\ p_{\text{fail}}^{\text{level-3}} &= c(p_{\text{fail}}^{\text{level-2}})^2 = c^7 p^8 \\ &\vdots \\ p_{\text{fail}}^{\text{level-}k} &= c(p_{\text{fail}}^{\text{level-}(k-1)})^2 = \frac{c^{2^k} p^{2^k}}{c} \end{aligned} \tag{14.65}$$

Thus, successive levels of concatenation will tend to suppress the error in the logical qubit provided $c p < 1$, where p is the probability of error per physical qubit per gate operation. Hence, there is a *threshold* in error probability of:

$$p < p_{\text{threshold}} \equiv \frac{1}{c} \tag{14.66}$$

in which case the error will be reduced with successive levels of concatenation. Hence, provided this error probability per qubit per gate threshold is met, it will be possible to implement quantum computations of arbitrary length to arbitrary accuracy. That is, one can quantum compute forever without error!

But what is the overhead in gate count we have to pay to achieve k levels of concatenation? Again following Julia Kempe's intuitive argument [270], if the circuit

we wish to implement has N gates when done without error correction, and we desire a final success probability of order $1 - p$, then in such a circuit each gate has to have a failure probability of less than or equal to p/N because errors compound. Hence, if we concatenate k times we will require:

$$p_{\text{fail}}^{\text{level-}k} = \frac{c^{2^k} p^{2^k}}{c} = p_{\text{threshold}} \left(\frac{p}{p_{\text{threshold}}} \right)^{2^k} \leq \frac{p}{N} \quad (14.67)$$

which implies

$$2^k \leq \frac{\log(N\epsilon_{\text{th}}/p)}{\log(\epsilon_{\text{th}}/\epsilon)} \quad (14.68)$$

So after k levels of concatenation, each gate turns into G^k gates where:

$$G^k = 2^{k \log G \leq (\frac{\log(N\epsilon_{\text{th}}/p)}{\log(\epsilon_{\text{th}}/\epsilon)}) \log G} = \text{poly}(\log N) \quad (14.69)$$

and so its final size will be $N \text{poly}(\log N)$, which is only polylogarithmically larger.

Opinions as to actual values of this error threshold have varied widely over the years. Initially, error rates per gate of around 10^{-4} – 10^{-7} were thought necessary, but the threshold has steadily been climbing [31, 291, 350, 400, 479, 484, 486]. The truth is, although we have talked about *the* threshold, in reality it is not unique: one can obtain different thresholds if one specializes the theory to different quantum computer architectures. Such considerations take into account the specific characteristics of different physical embodiments of quantum information processing wherein some error mechanisms are more prevalent than others. If one does this, one can obtain different assessments of the error rate per gate operation needed to sustain quantum computations of arbitrary length. Some recent studies suggest in certain architectures and schemes, the threshold could be as high as 3% [291].

14.10 Errors as Allies: Noise-Assisted Quantum Computing

We end this chapter with an observation. The prevailing opinion of quantum computer scientists is that quantum error correction is essential to achieving a useful quantum computer. However, is this necessarily true? For certain computations, such as factoring composite integers, where we seek an exact solution that is either plainly right or plainly wrong, we are indeed obliged to imbue our quantum computations with the ability to either avoid errors (e.g., using decoherence-free or topological encodings we shall describe in Chap. 15) or undo errors (e.g., using quantum error correcting codes). But there are many other computations in which we seek not a right or wrong answer, but instead, a “pretty good” answer. For example, in a maximum satisfiability problem, an ideal solution is one that satisfies the greatest number of constraints. However, in practice, we might be content with a solution that comes close to this ideal but not quite. In this case, the pragmatic measure of whether a quantum computer is better than a classical computer, is whether

it finds an equally good, i.e., equally sub-optimal, solution in less time, or whether it finds a better, albeit still sub-optimal, solution in the same time as required by a classical computer. Given the relative importance and ubiquity of such problems in comparison to integer factorization, a greater degree of investigation is warranted.

Moreover, surprisingly, there are a handful of results that suggest that noise, dissipation and decoherence can sometimes be an ally of quantum computation! For example, noise can be harnessed productively in entangled state preparation [342, 551], to effect quantum gate operations [39–41], and to enhance quantum transport in networks including, e.g., the light harvesting structures in plants [105, 359, 394, 413, 414]. It seems worthwhile to pursue such avenues to determine whether there is a strategy for quantum computation that makes noise a friend rather than an enemy.

14.11 Summary

There is an inherent contradiction amongst the ideal requirements for a quantum computing device. On the one hand the machine needs to be well isolated from the external world to permit it to evolve unitarily while executing some desired quantum computation. On the other hand, the machine needs to be strongly coupled to the external world to allow us to initialize it in an arbitrary starting state, or command it to perform a particular sequence of unitary gate operations. Switching the interaction with the external world on and off cleanly is extremely challenging experimentally. Hence errors are likely to arise in real quantum computing hardware.

In this chapter we have looked at several approaches to dealing with errors in quantum computations. We found that it is not as easy to detect an error in a quantum computation as it is in a classical computation because errors may exist along a continuum of possibilities and our ability and we are not even allowed to read a corrupted state directly, because such direct observations would make matters worse rather than better.

In the early days of quantum computing it was felt that such obstacles appeared to preclude the possibility of error correcting codes for quantum information. However, it turns out that quantum error correcting codes are possible. The trick is to entangle the qubit whose state we want to protect (the logical qubit) with several other physical qubits (i.e., ancillae) in such a manner that subsequent measurements on the ancillae qubits will reveal what error has afflicted the encoded data, and hence the corrective action needed to restore the logical qubit to its correct state. Crucially, these measurements on the ancillae only reveal information about the error and nothing about the state we wish to protect. Once the error is known it can be undone using the appropriate inverse unitary operation.

Various families of quantum codes are now known. We can estimate the tradeoffs different codes make regarding the number of logical qubits protected, the number of physical qubits into which they are encoded, and the maximum number of errors that can be corrected by way of the quantum Hamming, Singleton and Gilbert–Varshamov bounds. Sometimes tighter bounds have now been determined for many codes using more sophisticated methods. A database of known results is maintained

by Markus Grassl. The best code able to protect a logical qubit against a single bit-flip, phase-flip, or joint bit flip and phase flip, is the Laflamme-Miquel-Paz-Zurek 5-qubit code. This code saturates the quantum Hamming bound and is optimal. We gave complete circuits for the encoding and decoding stages of the 5-qubit code. We also showed the codewords used for less efficient codes that were discovered before the 5-qubit code.

In our original formulation of the 5-qubit code, the encoded qubit had to be mapped back to the unencoded (logical) basis periodically in order for the error correction to be performed. This exposes the logical qubit to uncorrectable errors while it is back in the unencoded basis. We described how the stabilizer formalism can combat this by performing error correction while staying entirely within the encoded basis.

An obvious issue with quantum error correction is that the error-correcting circuitry may itself introduce more errors. For quantum error correction to be truly viable, we need to be able to use imperfect error correction to achieve perfect computation. Fortunately, through a combination of fault-tolerant circuit design principles, and the use of concatenated coding, we showed that coding schemes can be devised that, in principle, permit error-correctable quantum computations of arbitrary length. We showed that to achieve such concatenated coding schemes the error probability per qubit per gate operation needs to be below a critical threshold. This threshold is sensitive to the error model, architecture, and physical embodiment used. However, schemes now exist that suggest error rates as high as 3% might be tolerable.

Two relatively new directions for handling errors in quantum computing are the use of noise sources as an ally in quantum computation, and the use of decoherence-free subspaces and topological quantum effects to make quantum hardware that is immune to errors. We shall examine such topics in the next chapter in the context of alternative models of quantum computation.

14.12 Exercises

14.1 Prove that any 2×2 matrix can be written as a weighted sum of Pauli matrices according to:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \frac{a+d}{2}\mathbb{1} + \frac{b+c}{2}X + \frac{i(b-c)}{2}Y + \frac{a-d}{2}Z \quad (14.70)$$

See Sect. 2.4.1.1 for a definition of the Pauli matrices.

14.2 Write down the operators that describe the following errors afflicting a 5-qubit state.

- (a) A bit-flip on the first qubit.
- (b) A phase-flip and the fifth qubit.

- (c) A joint bit-flip and phase-flip and the third qubit.
- (d) A bit flip on the second qubit and phase-flip on the fourth qubit.

14.3 Consider a single logical qubit in a state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ that interacts with an environment describable using just two qubits—a simplification indeed. Equation (14.31) says that the joint state of the qubit and its environment evolve as follows:

$$\begin{aligned} U|\psi\rangle|E\rangle &= (\alpha|0\rangle + \beta|1\rangle) \otimes \frac{|E_{00}\rangle + |E_{11}\rangle}{2} \quad (\text{no error}) \\ &\quad + (\alpha|0\rangle - \beta|1\rangle) \otimes \frac{|E_{00}\rangle - |E_{11}\rangle}{2} \quad (\text{phase flip}) \\ &\quad + (\alpha|1\rangle + \beta|0\rangle) \otimes \frac{|E_{01}\rangle + |E_{10}\rangle}{2} \quad (\text{bit flip}) \\ &\quad + (\alpha|1\rangle - \beta|0\rangle) \otimes \frac{|E_{01}\rangle - |E_{10}\rangle}{2} \quad (\text{joint phase flip \& bit flip}) \end{aligned}$$

Assuming the states $|E_{00}\rangle$, $|E_{01}\rangle$, $|E_{10}\rangle$, and $|E_{11}\rangle$ are orthonormal:

- (a) Prove that the states of the environment $\frac{|E_{00}\rangle + |E_{11}\rangle}{2}$, $\frac{|E_{00}\rangle - |E_{11}\rangle}{2}$, $\frac{|E_{01}\rangle + |E_{10}\rangle}{2}$, and $\frac{|E_{01}\rangle - |E_{10}\rangle}{2}$ are orthonormal. What is the significance of this in terms of error detection?
- (b) Prove that the three qubit state $U|\psi\rangle|E\rangle$ is entangled? What is the significance of this in terms of error determination?

14.4 The quantum circuit that encodes a single logical qubit in state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ within a 5-qubit entangled state according to Braunstein and Smolin's version of the Laflamme-Miquel-Paz-Zurek code is show in Fig. 14.3.

- (a) Use the encoding circuit together with the fact that $L = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$ to compute the unitary matrix corresponding to the Laflamme-Miquel-Paz-Zurek encoding circuit.
- (b) Verify that the circuit acting on input $|\psi\rangle|0000\rangle$ produces the state $|\psi\rangle_L = \alpha|0\rangle_L + \beta|1\rangle_L$ where the quantum codewords $|0\rangle_L$ and $|1\rangle_L$ are as given by (14.40).
- (c) Write down the state that results from a joint bit-flip and phase-flip on the fourth qubit in the encoded state.
- (d) Compute the unitary matrix corresponding to the Laflamme-Miquel-Paz-Zurek decoding circuit. Note, this is the inverse of the encoding circuit.
- (e) Use the decoding matrix of part (d) to error-correct the error-afflicted state of part (c). What is the resulting state?

14.5 The quantum circuit that encodes a single logical qubit in state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ within a 5-qubit entangled state according to Braunstein and Smolin's version of the Laflamme-Miquel-Paz-Zurek code is show in Fig. 14.3. This circuit protects

against a single general error afflicting any of the five qubits in the encoded state. However, to know a single general error has occurred we have to measure the error syndrome. Prior to such measurements no error has yet occurred. Hence, so long as we are not looking, multiple single qubit errors can afflict our state, but which one is actually realized only becomes definite after we measure the error syndrome! This is one of the amazing facts of quantum mechanics. This exercise will help you appreciate this subtlety of quantum error correction:

- (a) Determine the unitary matrix for Braunstein and Smolin's version of the Laflamme-Miquel-Paz-Zurek code.
- (b) Use the unitary matrix of part (a) to compute the encoded form of the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, i.e., $|\psi\rangle_L$.
- (c) Define two representative single qubit error operators as follows: Let \mathcal{E}_{B1} be the operator representing a bit-flip on the first qubit in the encoded state, and let \mathcal{E}_{P3} be the operator representing a phase shift on the third qubit in the encoded state. Compute the state that results when *both* errors afflict the encoded qubit equally.
- (d) How many errors have afflicted the encoded qubit at this point? [Think carefully before you answer].
- (e) Now run the buggy state through the Laflamme-Miquel-Paz-Zurek decoding circuit. What are the possible values for the error syndrome you can obtain?
- (f) How does the measurement of the error syndrome affect the state of the unmeasured (top) qubit?
- (g) Is it fair to say that the Laflamme-Miquel-Paz-Zurek code can correct multiple errors? [Think carefully before you answer].

14.6 In the Laflamme-Miquel-Paz-Zurek 5-qubit code we allow joint bit flip and phase flip errors to afflict a given qubit. However, does it make a difference whether the bit flip or phase flip occurs first? To investigate this, answer the following questions:

- (a) Show that a bit flip followed by a phase flip yields a strictly different result from a phase flip followed by a bit flip.
- (b) Show, however, that the error syndrome corresponding to a bit flip followed by a phase flip on qubit i is the same as the error syndrome corresponding to a phase flip followed by a bit flip on qubit i .
- (c) Even though the error syndromes are the same, are the corrective actions needed to restore the buggy qubit to its original state, the same? If not, does this mean that we cannot really error correct the qubit because we cannot distinguish between a bit flip followed by a phase flip from a phase flip followed by a bit flip?
- (d) Is there any measurement you could do that would tell you whether you had restored a qubit to its original state $|\psi\rangle$ or whether you had restored it to $-|\psi\rangle$?

14.7 Prove that the asymptotic form for the quantum Hamming bound (14.61) is given by:

$$\frac{k}{n} \leq \left(1 - \frac{t}{n} \log_2 3 - H\left(\frac{t}{n}\right) \right) \quad (14.71)$$

where H is the entropy $H(x) = -x \log_2 x - (1-x) \log_2(1-x)$.

14.8 Prove that the asymptotic form for the quantum Gilbert-Varshamov bound (14.64) is given by:

$$\frac{k}{n} \geq \left(1 - \frac{2t}{n} \log_2 3 - H\left(\frac{2t}{n}\right) \right) \quad (14.72)$$

where H is the entropy $H(x) = -x \log_2 x - (1-x) \log_2(1-x)$.

14.9 Which of the following $\llbracket n, k, d \rrbracket$ codes are ruled out by the quantum Hamming, quantum Gilbert-Varshamov, or quantum Singleton bounds?

- (a) A $\llbracket 5, 2, 3 \rrbracket$ code
- (b) A $\llbracket 7, 2, 3 \rrbracket$ code
- (c) A $\llbracket 10, 1, 5 \rrbracket$ code
- (d) A $\llbracket 10, 5, 3 \rrbracket$ code
- (e) A $\llbracket 19, 1, 7 \rrbracket$ code
- (f) A $\llbracket 20, 1, 11 \rrbracket$ code

14.1 Problem What is the “minimum length”, i.e., minimum value of n , of a $k=1$ quantum error-correcting code that corrects $t = 1, 2, 3, 4, 5, 6, 7, 8, 9$ errors?

Chapter 15

Alternative Models of Quantum Computation

“New ideas pass through three periods: It can’t be done. It probably can be done, but it’s not worth doing. I knew it was a good idea all along!”

– Arthur C. Clarke

Just as there are alternative models of classical computers, so too are there alternative models of quantum computers. These models differ in the quantum phenomena and resources they harness to perform quantum computation, and the ease with which they can be implemented in different hardware schemes. Many equivalences between these models have now been established, in the sense that it has been proven that any quantum computation performed in one model can be performed in another model with at most a polynomial cost overhead. Consequently, from the perspectives of complexity theory and computability theory, it does not matter which model one uses. However, model equivalence is only a theoretical construct. In the real world different models may be more or less easy to implement in a given hardware scheme, even though they might be computationally equivalent. Thus, we end the book with a survey of some competing models of quantum computation, and ideas people have had for how they might be implemented. It is quite possible that the first scalable universal quantum computer will be built according to one of these “non-standard” models of quantum computation. In all cases though, they make use of entanglement in one way or another. So one could say legitimately that there’s more than one way to skin a Schrödinger cat!

15.1 Design Principles for a Quantum Computer

Throughout this book we have implicitly assumed that quantum computers will be designed in accordance with the quantum circuit model of quantum computation. However, quantum circuits can be realized in physical hardware in many different ways. As one implementation proposal after another began to emerge in the late 1990’s it became increasingly difficult to compare and contrast the various hardware schemes and to rank their relative merits. Worse, some proposals began to appear

that claimed to be sufficient for universal quantum computation, but which actually harbored design flaws, that prevented them from scaling to arbitrary numbers of qubits.

In an attempt to abstract away from hardware specifics to more general design principles for quantum computers, in 2000 David DiVincenzo published a landmark paper that identified a minimal set of requirements needed for any hardware scheme to be able to implement quantum computation [148]. These have since become known as the “DiVincenzo Criteria” and have been of tremendous value in sharpening thinking on the similarities and differences between different approaches to quantum computation. In brief, the DiVincenzo Criteria can be stated as follows:

DiVincenzo Criteria

1. A scalable physical system with well characterized qubits.
2. The ability to initialize the state of the qubits to a simple fiducial state, such as $|000\dots0\rangle$.
3. Long relevant decoherence times, e.g., in the circuit model the states must decohere much slower than the gate operation time.
4. A “universal” set of quantum gates, or other primitive operations from which arbitrary quantum computations can be built.
5. A qubit-specific measurement capability.

These criteria have proven to be a useful checklist for experimentalists devising novel hardware designs for quantum computation because they distill out the essential functionality a hardware proposal needs to possess in order to be a plausible contender as a basis for scalable universal quantum computation. As you read the alternative models of quantum computation in this chapter, keep these criteria in mind. Even though many of these models deviate significantly from the original quantum circuit model the DiVincenzo Criteria, or minor tweaks thereon, still remain surprisingly valid across radically different quantum computational models.

15.2 Distributed Quantum Computer

The first model we shall look at is a slight twist on the standard quantum circuit model. In the standard circuit model there is no restriction on the number of qubits allowed within a single quantum processor. However, this assumption might not be valid for certain types of quantum computer hardware. For example, in superconducting quantum processors there is a practical limit to the number of control lines that will fit into a dilution refrigerator, which limits the number of qubits, sitting deep within the refrigerator, that can be addressed directly. Such practical considerations led some people to devise distributed quantum computer architectures wherein, instead of a single monolithic quantum processor, the distributed quantum computer consisted of several simple quantum processors connected by communications channels. Different models of distributed quantum computation arise

according to the type of communications channel that is allowed and whether there is the assumption of shared prior entanglement between quantum processor nodes.

If the communications channels can carry arbitrary *qubits*, then the distributed machine is, in principle, equivalent to the standard quantum circuit model. To effect a gate operation on qubits residing on different quantum processor nodes one would simply shuttle the qubits to a common processor node, perform the gate operation and return them to their original processors.

Conversely, if the communications channels can carry only *classical* information, then the distributed machine is a much weakened quantum computer because entanglement can be created within the individual quantum processors but not between them. These are so-called Type-II quantum computers.

The most intriguing possibility is a hypothetical quantum computer that lies somewhere between these extremes: this is a distributed quantum computer wherein the individual quantum processors are assumed to be supplied with pairs of maximally entangled qubits each in state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. During operation, the only communication allowed between the quantum processors is the exchange of classical information. How powerful would such a distributed quantum computer architecture be?

To answer this is it sufficient to determine whether it is possible to effect a universal set of gates on this architecture. We could take our target universal set of gates to be CNOT and all 1-qubit gates. Clearly, the 1-qubit gates pose no problem because they are guaranteed to involve a single qubit, which has to be wholly within one quantum processor node. Likewise, a CNOT gate applied to a pair of qubits that are both within the same quantum processor node pose no problem either. The only difficulty is whether it is possible to perform a CNOT gate on a pair of qubits in which the control qubit is within one quantum processor node and the target qubit is within another. If such a gate operation is possible, then the distributed quantum computer imbued with shared prior entanglement must be equivalent to the standard circuit model.

Such a proof was given by J. Eisert, K. Jacobs, P. Papadopoulos and M. Plenio [165]. Their solution is shown in Fig. 15.1. The control qubit is on the top rail (inside processor A) and the target qubit is on the bottom rail (inside processor B). In addition, there is an EPR pair, $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, shared between processors A and B on the second and third rails in the figure. Such EPR pairs are assumed to be a shared prior resource in this computational model.¹ The first CNOT gate and measurement-outcome-controlled-X gate move the control qubit from processor A to processor B, and the CNOT is effected. The Hadamard gate and the measurement-outcome-controlled-Z gate return the control qubit to the top rail. Notice that there have been two communicative acts between processors A and B, but these only involved sending classical bits, which were then used to control 1-qubit gates. Thus, only classical information is exchanged during the quantum computation. This fact

¹One way to create such shared prior entanglement is to perform the quantum computation $(H \otimes \mathbb{1}) \cdot \text{CNOT}|00\rangle$ and physically transport one member of the pair to processor A and the other to processor B.

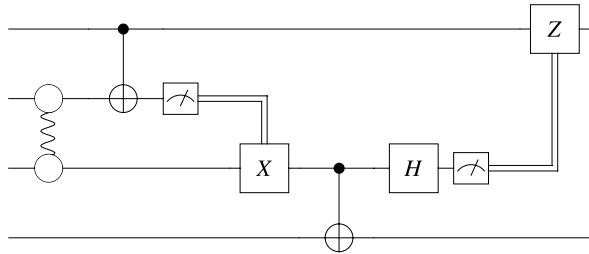


Fig. 15.1 A CNOT gate can be effected on two qubits that reside on different quantum processor nodes of a distributed quantum computer. In the figure, the *vertical squiggle* indicates that an EPR pair has been shared between processor A and B prior to any computation taking place. This shared prior entanglement is consumed during the computation of the distributed CNOT gate in order to move the control qubit into the same processor as the target qubit whereupon the CNOT can be effected. Thereafter the Hadamard gate and measurement-outcome-controlled- Z gate restores the control qubit to the top rail. Note that the joint state of the control and target qubits prior to the distributed CNOT can themselves be entangled due to prior computations and the distributed CNOT works just as well

is emphasized Fig. 15.1 by the use of the double lines. The EPR pair is established prior to the computation, i.e., “offline”.

Thus, a distributed quantum computer architecture consisting of several simple quantum processors, that *share fixed prior entanglement*, and are coupled by *classical* communication channels, *is* sufficient to achieve arbitrary quantum computation. These ideas have been extended to distributed QFT and distributed Shor’s algorithm by Anocha Yimsiriwattana and Samuel Lomonaco [552, 553].

15.3 Quantum Cellular Automata Model

Cellular automata (CA) are hypothetical computing devices that consist of a spatially infinite 1-, 2- or 3-dimensional lattice of “cells”, each of which can be in one of finitely many states, to which the same update rule, with inputs from the local neighborhood of each cell, is applied in parallel. They were first introduced by John von Neumann as a possible answer to the question “What kind of logical organization is sufficient for an automaton to reproduce itself?” [521]. They have since found extensive application in many scientific fields and are the foundation for Stephen Wolfram’s vision of a “New Kind of Science” [544].

In classical cellular automata, the value assigned to the cell at a particular index, x say, at time step $t + 1$ depends on the values held by its neighbors, \mathcal{N}_x , at time step t . For example, in a one-dimensional CA (say), to update the cell at index x one would read the contents of the cells at indices $x - 1$ and $x + 1$ and set cell x to have a value that is some (fixed) function of the measured values. Hence, to update the classical cellular automaton, it is necessary to maintain two registers: one that records the instantaneous values of the neighbors at time t and the other that records what the new cell values are about to become at time step $t + 1$ after all the updates

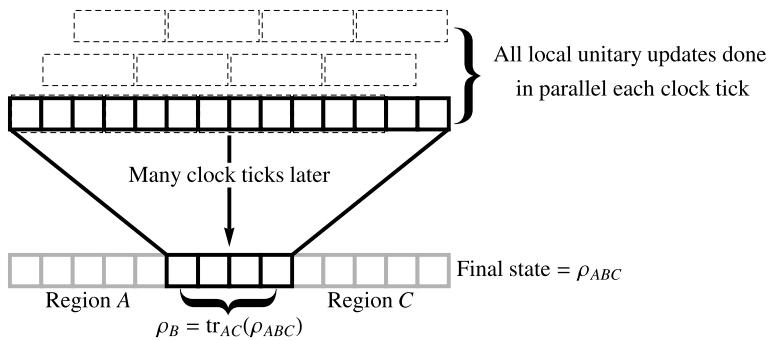


Fig. 15.2 A quantum cellular automaton (QCA). Each cell is initially assigned a starting value, i.e., quantum state. Thereafter, a local-unitary update rule is applied in parallel to contiguous blocks of nearest neighbor cells. In the figure a 1D QCA is shown together with the triplets of cells corresponding to the center cell and its two nearest neighbors. These triplets of cells overlap one another in the one-dimensional array of cells but, to visualize them clearly, we show them stacked in the figure. Unlike classical cellular automata no cells are read during the evolution. This allows the quantum state of the whole QCA to evolve into a non-separable (entangled) superposition. Nevertheless, such an entangled state can be expressed as a sum over separable states each corresponding to a different direct product of basis states in the cells. At the end of the evolution when the QCA is in state ρ_{ABC} the state of a particular subset of the cells is obtained by the partial trace over the complementary subset of cells, e.g., $\rho_B = \text{tr}_{AC}(\rho_{ABC})$. Although the QCA model is often defined over an infinite lattice, in practical implementations finite-sized hardware is used having periodic boundary conditions, or quiescent states, at the edges of the finite lattice

have been applied. Classically, this strategy can be implemented easily as reading and copying pose no problems whatsoever. In practical applications, we usually work with a finite lattice having either periodic boundary conditions (e.g., a torus (or doughnut shape)—which is periodic in two dimensions) or quiescent states, i.e., states that do not change under an update operation.

Quantum cellular automata (QCA) are quantum generalizations of classical cellular automata. Thus, we can regard them as hypothetical quantum computing devices that consist of a spatially infinite 1-, 2- or 3-dimensional lattice of “cells”, each of which can be in one of finitely many *basis* states, $\{|\psi^{(i)}\rangle\}$ say, to which the same update rule, with inputs from the local neighborhood of each cell, is applied in parallel (see Fig. 15.2). Unlike the classical case, where we actually read the values held by the neighbors, in a QCA we do not. Instead, the “read” operation is replaced by a fixed unitary operator, U say, applied uniformly across all cells. This unitary operator needs to have the property of commuting with all lattice translations of itself. That is, if U_x and U_y correspond to the same operation U centered on two different cell indices, x and y , then we require $[U_x, U_y] = 0$ for all x and y . This requirement ensures that the *order* in which the updates U_x and U_y are applied does not change the end result of the update of the QCA as a whole. Note that, even if each cell starts out in one of their allowed basis states, the dynamical evolution of the QCA will quickly entangle the states of the cells. Nevertheless, such an entangled state can be expressed as a sum over separable states each corresponding to different direct product of basis states for the cells. This means that the overall QCA

will become a non-separable (entangled) superposition of the form:

$$\sum_{i,j,k,\dots} \cdots |\psi_{x-1}^{(i)}\rangle|\psi_x^{(j)}\rangle|\psi_{x+1}^{(k)}\rangle\cdots \quad (15.1)$$

where $|\psi_x^{(j)}\rangle$ corresponds to the cell at index x being in basis state $|\psi^{(j)}\rangle$ etc. Only when a final measurement is made on some desired subset of cells is the final fate of the computation actually determined. Otherwise the state of a particular subset of cells would be given by the partial trace over the complementary subset of cells.

Although such a quantum generalization of cellular automata is intuitive, it turns out to be problematic in the sense that one can imagine reasonable-sounding quantum cellular automata, such as one that merely shifts the contents rightwards, that cannot be achieved within such a model. The crux of the problem is that the operations of reading and updating the QCA cannot be done as one move in a QCA as they can in a classical CA. To fix this, Carlos Perez-Delgado and Donny Cheung's have devised a local unitary quantum cellular automaton (LU-QCA) model [391], which subsumes previous QCA proposals [158, 159, 216, 440, 508, 523], and yet does not have this problem. The basic idea is as follows:

Quantum Cellular Automaton A QCA is a 5-tuple $(L, \Sigma, \mathcal{N}, U_0, V_0)$ consisting of:

1. a d -dimensional lattice of cells indexed by integers. $L = \mathbb{Z}^d$,
2. a finite set Σ of orthogonal basis states,
3. a finite neighborhood scheme $\mathcal{N} \subseteq \mathbb{Z}^d$,
4. a local read function $U_0 : (\mathcal{H}_\Sigma)^{\otimes \mathcal{N}} \rightarrow (\mathcal{H}_\Sigma)^{\otimes \mathcal{N}}$ such that any two lattice translations U_x and U_y must commute for all $x, y \in L$, and
5. a local update function $V_0 : \mathcal{H}_\Sigma \rightarrow \mathcal{H}_\Sigma$.

The key trick is to account for every operation the QCA needs to perform in a physically allowed way. So in the QCA Perez-Delgado and Cheung use two operations: one, U_x , to *compute* the updated value to cell x and one, V_x , to *perform* the actual cell update to cell x . Thus, the local update rule at cell index x is $R_x = V_x \cdot U_x$ and the global update rule is therefore:

$$R = V \cdot U = \left(\bigotimes_x V_x \right) \cdot \left(\bigotimes_x U_x \right) \quad (15.2)$$

Although real quantum cellular automata would be implemented directly in quantum hardware such as optical lattices [62, 84, 140] (see Fig. 15.3), we can best grasp idea of the local unitary QCA by looking at how it would be simulated (efficiently) in a traditional quantum circuit as shown in Fig. 15.4: The update rule computes what the new cell values will be in such a manner that the calculation for one cell does not prevent the neighboring cell from computing its update too. Then a final set of single qubit unitaries actually apply the updates.

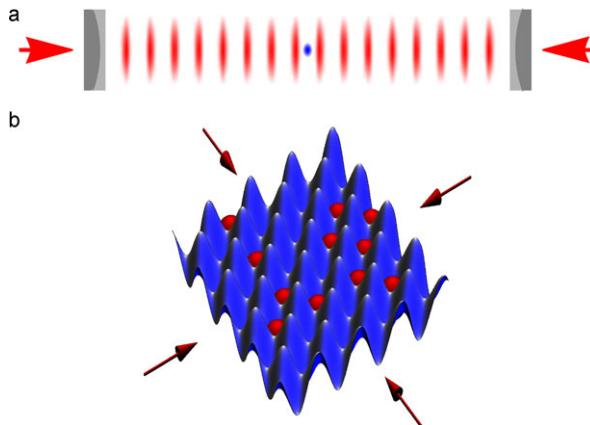


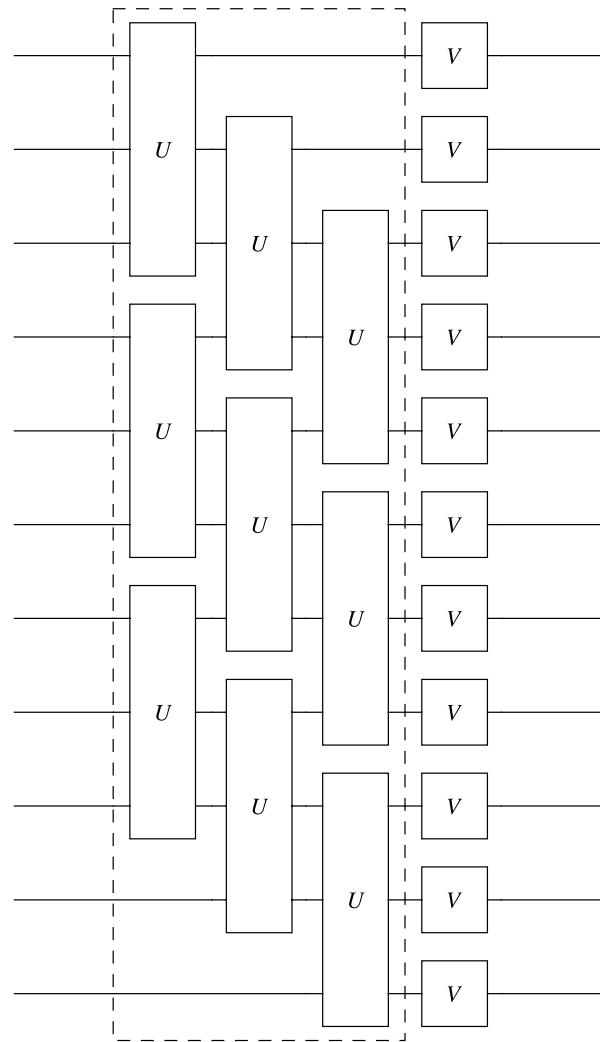
Fig. 15.3 (a) An optical standing wave is formed by superimposing two counter-propagating laser beams. The minima of the standing wave act as a perfectly periodic optical trap for cold atoms. This creates a structure mirroring that of a one-dimensional QCA. (b) If more counter-propagating laser beams are added in directions orthogonal to the first, we can create two-dimensional and three-dimensional optical lattices. Operations on nearest neighboring sites can then implement the fundamental update rule of a QCA

Perez-Delgado and Cheung show that their QCA is universal for quantum computation by proving it can implement any valid quantum circuit. Hence, if one wanted to build a quantum computer, one could in principle build it in a local unitary QCA architecture. Such a scheme has many advantages: it is very uniform, only requires nearest neighbor interactions, and maps in a very natural way onto certain types of quantum hardware such as optical lattices [62]. Moreover, the QCA model itself is very close to certain physical systems such as Ising spin systems and quantum lattice gases and so is likely to be ideally suited to simulating such systems. Thus the QCA is a very strong contender for a universal quantum computer architecture that could actually be realized.

15.4 Measurement I: Teleportation-Based Quantum Computer

The standard quantum circuit model assumes the ability to perform any quantum gate from a universal set of gates deterministically on demand. Various families of such gates were given in Table 2.14. You will notice, however, that every case each universal set of gates includes at least one 2-qubit gate. It turns out, however, that it can be very difficult to achieve such quantum gates deterministically on demand experimentally. In many schemes when we attempt a particular gate operation, especially a multi-qubit one, the operation may fail. If this were to happen in the midst of a long quantum computation, it could easily corrupt the quantum state being acted upon, and cause the computation to go awry. While quantum error correction can of

Fig. 15.4 Quantum circuit simulating a quantum cellular automaton (QCA). The dotted area corresponds to the “read” operation being applied to each cell and its immediate nearest neighbors. This read operation does not involve making any measurements at all, but instead involves applying the same unitary operator U across the whole array (in this case a 1-dimensional array). These U operators compute the updated value of the i -th cell and write it in an ancillary memory location (not shown). Then a subsequent V operation actually applies the update



course handle this, in principle, one wonders if there might not be some more fundamental way to change the computational model so that these kinds of gate failures can be avoided.

In most implementation schemes the 1-qubit gates can be achieved more reliably than the 2-qubit gate (or gates) needed in the particular universal gate family being employed. Thus, if only we could eliminate the need to perform the 2-qubit gates deterministically on demand during a quantum computation then perhaps quantum computation would be easier to achieve experimentally. This is exactly what Daniel Gottesman and Ike Chuang proposed in 1999 when they invented an alternative, measurement-based, model for quantum computation that requires only single qubit gates, offline Bell-basis measurements, and the ability to make and store many

copies of a special entangled state we call $|\Phi\rangle$ [210]. If we can achieve these operations as primitives then Gottesman and Chuang showed how to assemble them into arbitrary quantum computations. These scheme has the attractive feature that only single qubit operations and measurements are needed during the actual quantum computation. All the hard work is pushed offline to the creation and storage of these special entangled states $|\Phi\rangle$. Thus was born first measurement-based model of quantum computation. This treated quantum teleportation, and its associated Bell-basis measurement, as computational primitives from which arbitrary quantum computations could be built. This work inspired the revolutionary paper by Knill, Laflamme, and Milburn showing that it was possible to do universal quantum computation using only linear optical elements and photo-detectors. Prior to this, it was assumed that optical quantum computers had to involve nonlinear optical elements, such as a nonlinear Kerr medium, to achieve a CNOT gate [107]. However, the Gottesman and Chuang scheme essentially showed how to use the non-linearity implicit in the quantum measurement process to achieve a CNOT gate, deterministically on demand. This so-called “teleportation-based quantum computer” was further developed by Michael Nielsen [371] and Debbie Leung [310].

To understand how teleportation-based quantum computation works, we can proceed in stages. First realize that the original teleportation scheme we saw in Chap. 12 makes it possible to move a 1-qubit state from one place to another. In that case, you will recall the critical role played by the Bell basis measurement operation. This gives us a two-bit classical message that informs the receiver what operation they need to perform on their entangled qubit in order to re-incarnate the original quantum state being teleported. In the present context, it is most important to realize that, although we often *picture* a Bell basis measurement in terms of a quantum circuit, as illustrated in Figs. 15.5 and 15.6, as comprising a CNOT gate and Hadamard gate followed by a measurement in the computational basis, we do not have to think of it this way. Instead, a Bell basis measurement operation itself can be thought of as a primitive operation in quantum computation, without any CNOT gate being involved at all. Moreover, each Bell basis measurement can be regarded as a kind of measurement based quantum gate, which can be inserted into a quantum circuit in two possible orientations. The difference is apparent by comparing Figs. 15.5 to 15.6.

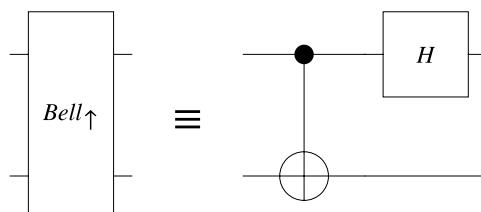


Fig. 15.5 A Bell-basis measurement gate. A Bell measurement gate can be described in terms of a CNOT gate, a Hadamard gate, and a measurement in the computational basis. However, one does not have to have CNOT gate available to implement a Bell basis measurement

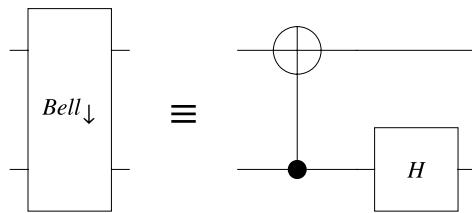


Fig. 15.6 A Bell measurement gate inserted “upside down”. A Bell-basis measurement gate can be described in terms of a CNOT gate, a Hadamard gate, and a measurement in the computational basis. However, one does not have to have CNOT gate available to implement a Bell-basis measurement

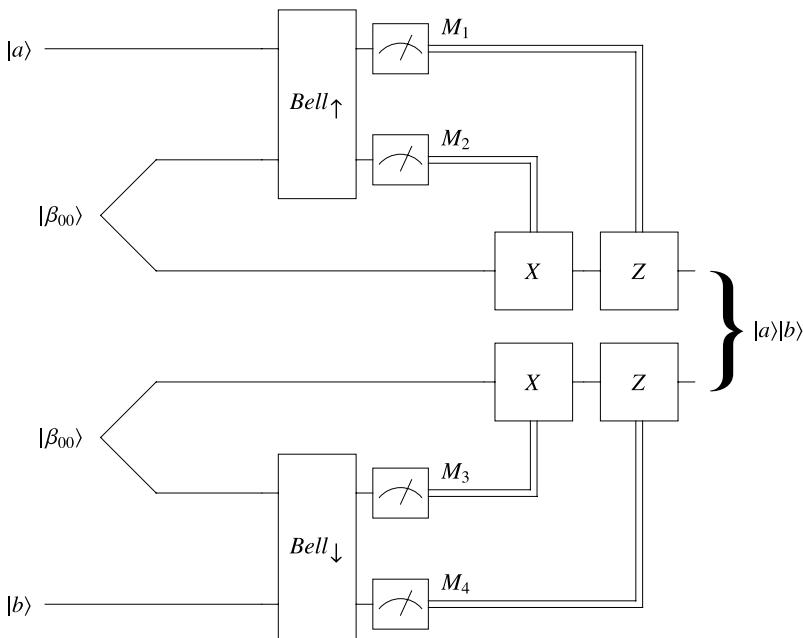


Fig. 15.7 Two mirror-image teleportation circuits succeed in teleporting a pair of qubits simultaneously. The input state is of the form $|a\rangle|\beta_{00}\rangle|\beta_{00}\rangle|b\rangle$ where $|a\rangle$ and $|b\rangle$ are arbitrary computational basis states and $|\beta_{00}\rangle|\beta_{00}\rangle$ is a pair of Bell states. In the middle of the circuit four Bell basis measurements are made yield (classical) bit values $\{M_1, M_2, M_3, M_4\}$. These values are used to control subsequent applications of *X* and *Z* gates. Regardless of the measurement outcomes $\{M_1, M_2, M_3, M_4\}$, the output on the middle pair of qubits (i.e., qubits 3 and 4) is always $|a\rangle|b\rangle$, i.e., the same as the input state on qubits 1 and 6. Although non-obvious, the same circuit can teleport an *arbitrary* (i.e., entangled and/or mixed) 2-qubit state split across qubits 1 and 6 too. The proof of this is given as a guided exercise at the end of this chapter

Next consider what happens if we employ two such teleporters simultaneously. In this case, we can move an arbitrary 2-qubit state from one place to another, as illustrated in Fig. 15.7. Notice, in the figure, that the two Bell basis measurements

are inserted into the double teleporter upside down with respect to one another. This makes the overall circuit more symmetric. The net result though is that we can now teleport the state of two qubits from one place to another.

To see this, we have shown, in Fig. 15.7, the case in which the input 2-qubit state is a product state fed into the circuit on qubits 1 and 6, i.e., $|\psi_{1,6}\rangle = |a\rangle|b\rangle$. However, as you will confirm later in the exercises, the same double teleportation circuit can successfully teleport any type of 2-qubit state, regardless of whether it is entangled, i.e., $|\psi_{1,6}\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$ such that $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$, or even mixed, i.e., $\rho_{1,6}$. A simple way to approach this proof is to start with $|\psi\rangle|\beta_{00}\rangle|\beta_{00}\rangle$ (on qubits 1 through 6 inclusive) and then permute the qubits using $\text{SWAP}_{1,6;6} \cdot \Pi_{2^6} \cdot |\psi\rangle|\beta_{00}\rangle|\beta_{00}\rangle$ to input $|\psi\rangle$ on qubits 1 and 6, $|\beta_{00}\rangle$ on qubits 2 and 3, and $|\beta_{00}\rangle$ on qubits 4 and 5. Then follow this input state through the circuit to show the circuit transforms it correctly. The working steps are left as an exercise.

Given the ability to teleport a 2-qubit state, we can then insert a CNOT gate immediately prior to the output from the double teleporter, and clearly be able to achieve a CNOT of the input state, i.e., $\text{CNOT}|a\rangle|b\rangle$, in the output, as shown in Fig. 15.8.

We can then insert a “no-op” operation in the form of two CNOTs back to back, as illustrated in Fig. 15.9.

Now we can imagine re-grouping the gates in the circuit so that we associate one of these new CNOT gates with the circuit to its left and the other new CNOT gate with the circuit to its right. The result is shown in Fig. 15.10

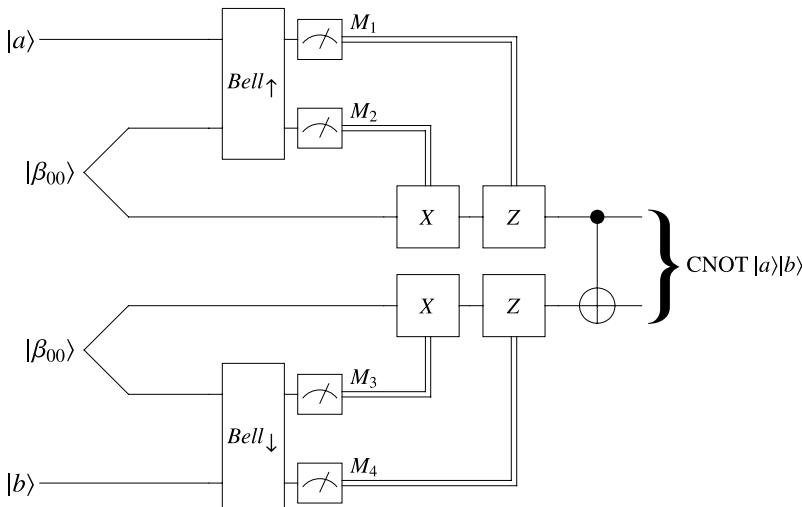


Fig. 15.8 Circuit for teleporting a 2-qubit state followed by a CNOT operation. Clearly, as the mirror image teleportation circuits merely teleport the joint state of qubits 1 and 6 to the joint state of qubits 3 and 4, applying a CNOT gate to this output causes it to become $\text{CNOT}|a\rangle|b\rangle$. Again, although shown for a separable input state, the same circuit also works for an arbitrary 2-qubit state input on qubits 1 and 6

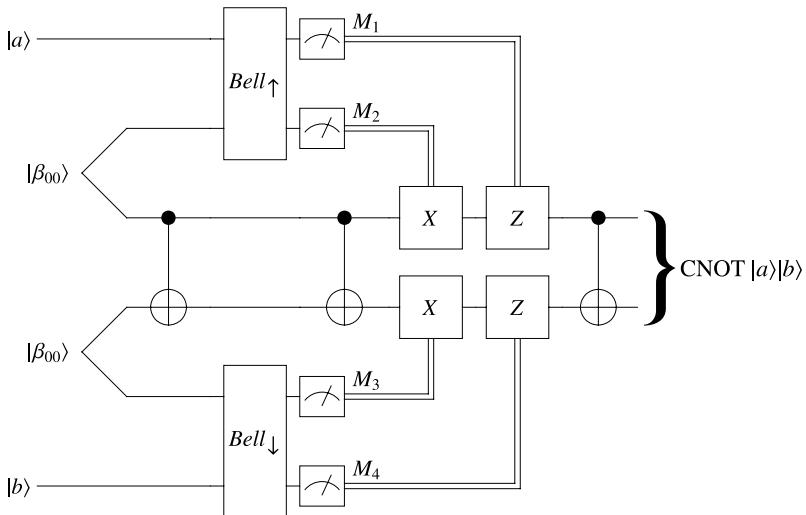


Fig. 15.9 The same circuit as Fig. 15.8 except for the introduction of the identity matrix written as a product of two back to back CNOT gates

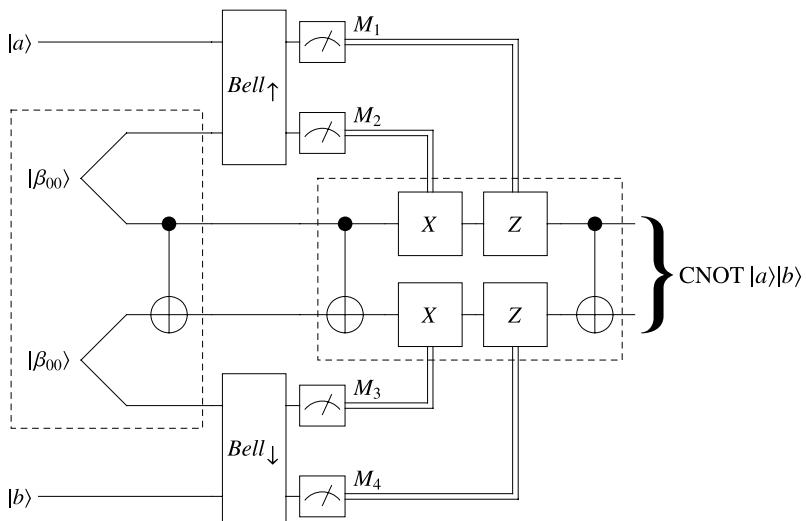


Fig. 15.10 Re-interpretation of Fig. 15.9 by re-grouping the gates as implied by the *dashed boxes*. The *left hand dashed box* is an operation that will produce some fixed entangled state independent of the joint state input on qubits 1 and 6. The *right hand dashed box* corresponds to a quantum circuit whose behavior is classically controlled by the bit values $\{M_1, M_2, M_3, M_4\}$. Note that, at this point, the right hand dashed box contains two CNOT gates

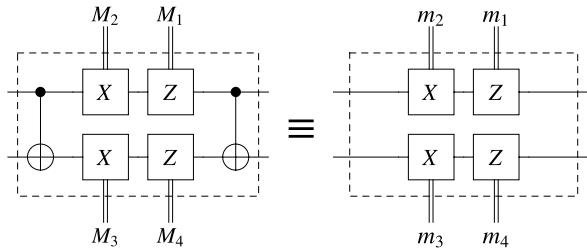


Fig. 15.11 Quantum circuit identity showing that the quantum circuit fragment within the right hand dashed box in Fig. 15.10 can be replaced with exclusively classically controlled gates provided the control logic is modified. Specifically, when the control values in the circuit involving two CNOT gates are $\{M_1, M_2, M_3, M_4\}$ the corresponding control values for the circuit devoid of CNOT gates are $\{m_1, m_2, m_3, m_4\}$. The corresponding sets of control values needed to realize the circuit fragment equivalence are shown in Table 15.1

The circuit inside the left hand dashed box in Fig. 15.10 generates the following fixed entangled state:

$$\begin{aligned} |\Phi\rangle &= (\mathbb{1} \otimes \text{CNOT} \otimes \mathbb{1}) \cdot |\beta_{00}\rangle|\beta_{00}\rangle \\ &= \frac{1}{2}|0000\rangle + \frac{1}{2}|0011\rangle + \frac{1}{2}|1101\rangle + \frac{1}{2}|1110\rangle \end{aligned} \quad (15.3)$$

Although the synthesis of $|\Phi\rangle$ requires a CNOT gate we imagine doing these computations offline, and storing many copies of $|\Phi\rangle$ in a quantum memory of some kind. Thereafter these pre-prepared entangled states are assumed to be available on demand for input to the middle ports of the circuit shown in Fig. 15.10 *after* the leftmost dashed box.

We are not quite done however, because it looks like the right hand dashed box in Fig. 15.10 contains two CNOT gates, and as recall the whole point of us doing this is to eliminate the need for any 2-qubit gates during active quantum computations. However, the two CNOTs in this case are illusory. To see this, consider Fig. 15.11. You are asked to verify these equivalences in Problem 15.4. This indicates we can rewrite the sub-circuit to a form devoid of CNOT gates merely by changing the classical control values. Specifically, when the control values in the circuit involving two CNOT gates are $\{M_1, M_2, M_3, M_4\}$ the corresponding control values for the circuit devoid of CNOT gates are $\{m_1, m_2, m_3, m_4\}$. The corresponding sets of control values needed to realize the circuit fragment equivalence are shown in Table 15.1.

Thus, the circuit identity shown in Fig. 15.11 proves the possibility of teleporting a CNOT gate provided we can achieve Bell basis measurements, classically controlled 1-qubit gates, and a supply of pre-prepared specially entangled states $|\Phi\rangle = (\mathbb{1} \otimes \text{CNOT} \otimes \mathbb{1}) \cdot |\beta_{00}\rangle|\beta_{00}\rangle$. In other words, provided we can create and store many instances of the entangled state $|\Phi\rangle$, no further entangling operation is needed in order to perform any feasible quantum computation. The appeal of this approach is that, in certain approaches to quantum computing hardware, this teleportation-based scheme may prove easier to implement than arbitrary 1-qubit gates and (deterministic) CNOT gates on demand. Indeed, it paved the way for the

Table 15.1 Corresponding pairs of control values for quantum circuits shown in Fig. 15.11. Specifically, when the set of control values for the circuit containing two CNOT gates are $\{M_1, M_2, M_3, M_4\}$ the corresponding control values for the circuit devoid of CNOT gates are $\{m_1, m_2, m_3, m_4\}$. You are asked to verify this in Problem 15.4

M_1	M_2	M_3	M_4	m_1	m_2	m_3	m_4
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	1	0	1	1
0	1	0	0	0	1	1	0
0	1	0	1	1	1	1	1
0	1	1	0	0	1	0	0
0	1	1	1	1	1	0	1
1	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	1	0	1	0
1	0	1	1	0	0	1	1
1	1	0	0	1	1	1	0
1	1	0	1	0	1	1	1
1	1	1	0	1	1	0	0
1	1	1	1	0	1	0	1

proof by Manny Knill, Raymond Laflamme and Gerald Milburn that universal quantum computation can be achieved using linear optical elements and photodetectors [293].

15.5 Measurement II: One-Way Quantum Computer

In 2001 R. Raussendorf and H.J. Briegel proposed an alternate measurement-based model of quantum computation [409]. Whereas in the teleportation-based scheme one must perform joint (i.e., complete Bell-basis) measurements, in the one-way scheme one need only use *single*-qubit measurements. This attribute of the one-way quantum computer model is believed by many to enable much simpler implementations of universal quantum computers.

The basic idea behind the one-way quantum computer is as follows: first one prepares the system in a highly entangled initial quantum state, called a “cluster-state”, which is independent of whatever quantum computation one is to perform. Next, one applies a sequence of single-qubit *measurements* on the qubits. The order in which these measurements are performed, and the choice of basis for each one, defines the quantum computation one performs. The measurements are “adaptive” in the sense that the measurement basis chosen for a given measurement can be computed (classically) based on the prior measurement outcomes. The scheme is called “one-way” because as a one-way quantum computation is performed the measurements introduce a time-asymmetry in the dynamics that prevents it from

being reversed unambiguously, so the computation can only be run forwards, i.e., “one-way”. Remarkably, it is possible to achieve universal quantum computation in this manner [410, 411].

The great appeal of the one-way quantum computer model is that it shifts the bulk of the technical challenge to the preparation of the initial (highly-entangled) cluster states. Thereafter, the single qubit measurements are assumed to be relatively easy. In practice, things are rarely so simple because the model does assume one can perform measurements on any qubit without it affecting nearby qubits. Depending on the architecture of the one-way quantum computer, such individual qubit addressing may or may not be so easy.

For example, the one-way model is especially popular with proponents of “optical lattice” quantum computers [82, 84, 140]. In such devices, once creates a standing wave potential using intersecting and counter-propagating laser beams. This creates a lattice of potential minima within which atoms can be cooled and trapped—rather like an egg-carton made from standing waves of light. Having confined the atoms spatially on a scale of the order of the wavelength of laser light used, one can then use the internal states of these atoms as qubits. Moreover, by fine tuning the applied laser beams one can engineer entanglement between nearest neighbor atoms and thereby build up the initial highly-entangled cluster states needed for the one-way computer. Unfortunately, the atoms in the optical lattice turn out to be are trapped so close together that it then becomes difficult to address them individually in order to perform the single-qubit measurements needed to actually effect the desired quantum computation. With effort this problem is expected to be solved. Nevertheless, it does illustrate that hardware implementations of quantum computation can be challenging even if the underlying physical model of quantum computation appears relatively straightforward. Ideas from the one-way model of quantum computation are now being combined with those of topological quantum computation to conceive of a one-way quantum computer architecture that is resilient to noise [412].

15.6 Topological Quantum Computer

“Topological quantum computation does not try to make the system noiseless, but instead deaf – that is, immune to the usual sources of decoherence.”

– Sankar Das Sarma, Michael Freedman, and Chetan Nayak

The usual approach to combatting errors in quantum computations, which we explained in Chap. 14, is to reduce the error rate in quantum gate operations to below a critical threshold such that fault tolerant concatenated quantum error correcting codes can then be used to achieve error-correctable quantum computations of arbitrary length. By contrast, the idea behind topological quantum computation (TQC), which was conceived of by Alexei Kitaev in 1997, is to encode quantum information in topological degrees of freedom that are inherently immune to errors [285]. Thus, rather than focussing on error *correction*, TQC focuses instead on error *avoidance*. The rationale being that it is better to avoid errors rather than to allow them to occur

and then have to correct them. For some good overviews of topological quantum computing see [27, 81, 114, 365].

15.6.1 Topological Quantum Effects

Topology is concerned with those properties of geometry that are not affected by stretching, shrinking, and twisting surfaces. Thus the common joke is that “A topologist cannot tell the difference between a donut and a coffee cup” [125] because one surface can be smoothly deformed into another without having to cut anything.

The trick to achieving such error-immune quantum computation is to exploit certain topological states of matter to encode qubits. That topology can affect quantum states is most commonly known from the example of the Aharonov-Bohm effect. Here a charged particle is made to encircle a line of magnetic flux. In so doing it accumulates a phase factor that depends only upon the number of times the charged particle winds around the magnetic flux line but not on the geometry of the path taken.

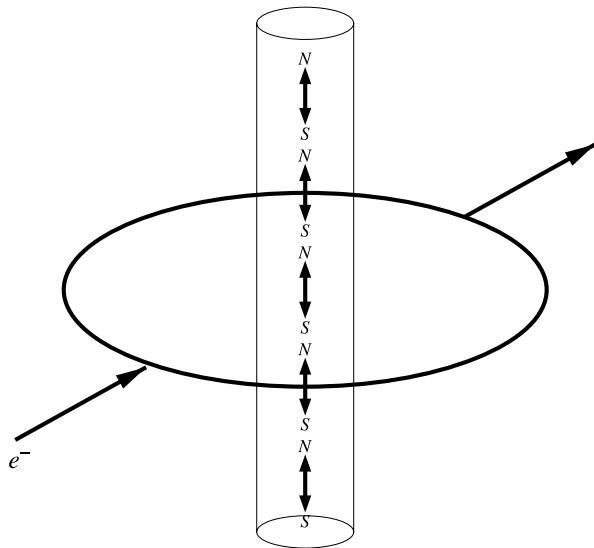
However, more exotic possibilities arise in the physics of particles confined to move in only two dimensions. This is because certain operations on identical particles that are indistinguishable from one another in three dimensions become distinguishable when the particles are confined to move in only two dimensions.

To give an intuition for how this is possible, imagine a pair of hypothetical quasi-particles, A and B say, that each carry charge and magnetic flux. If one such particle is made to encircle the other the situation is reminiscent of the Aharonov-Bohm effect in which a charge is made to encircle a line of magnetic flux (see Fig. 15.12). However, in three-dimensional space one can imagine looping the path of the moving quasi-particle, A , *over* the stationary quasi-particle, B , and shrinking the looping path down to a point co-located with the starting position of quasi-particle A . If we do this, the trajectory of quasi-particle A that encircles quasi-particle B is *topologically* equivalent to a path that did not encircle quasi-particle B ! So in three dimensions the geometrically distinct trajectories of quasi-particle A are topologically equivalent. However, if the quasi-particles A and B are confined to move only on a two-dimensional surface containing them such that quasi-particle A is made to encircle quasi-particle B , we won’t be able to deform this trajectory to one that does not encircle quasi-particle B because we are not allowed to move quasi-particle A in the direction normal to the surface. Hence, in two dimensions, there is no notion of passing one particle “over” another. Thus, certain geometric trajectories that are topologically equivalent in three dimensions are not equivalent in two dimensions. Fundamentally, this gives rise to new physical possibilities.

For example, let us imagine a quantum system described initially by the quantum state $|\psi\rangle = |\psi_A\rangle|\psi_B\rangle$, and let S be the operator that swaps their spatial locations. We can describe the effect of such an operator as:

$$S|\psi_A\rangle|\psi_B\rangle = e^{i\theta}|\psi_B\rangle|\psi_A\rangle \quad (15.4)$$

Fig. 15.12 When a charged particle encircles a line of magnetic flux it accumulates a phase that depends only upon the number of times the particle winds around flux but not the path taken. This is an example of a topological quantum effect



and so two consecutive swaps would give us:

$$S^2 |\psi_A\rangle |\psi_B\rangle = e^{2i\theta} |\psi_A\rangle |\psi_B\rangle \quad (15.5)$$

such that the particles end up back where they started. Hence, $S^2 = \mathbb{1}$, the identity operator. Consequently, $e^{2i\theta} = 1$ and so $e^{i\theta} = \pm 1$. Thus, if $\theta = 0$ the particles pick up a phase of $+1$ under particle exchange, and we call such particles bosons. However, if $\theta = \pi$ the particles pick up a phase of -1 under particle exchange, and we call such particles fermions. In three dimensions these are the two basic kinds of particles allowed.

15.6.2 Beyond Fermions and Bosons—Anyons

However, if the particles are confined to move on a two dimensional surface, another possibility arises. In particular, at extremely low temperatures and in the presence of very strong magnetic fields, aggregates of strongly interacting particles can form that possess unusual properties, such as fractional electronic charges [15, 134]. The aggregates behave collectively as if they are a new kind of particle, so they are given the name “quasi-particles”. In three dimensions such quasi-particles are invariably bosons or fermions. However, in two-dimensions these are no longer the only possibilities. It can happen that when such quasi-particles are made to encircle one another on a two-dimensional surface, their quantum state can be made to accumulate any phase between 0 and π . Consequently, such quasi-particles represent a fundamentally new type of object, distinct from bosons and fermions, called *anyons*, because they can pick up *any* phase.

Although anyons may seem like only a hypothetical possibility they do, in fact, exist in, e.g., the fractional quantum Hall effect [15, 113, 134, 286]. When one type of fractional quantum Hall state encircles another once it acquires a phase factor of $e^{2\pi i/3}$, and when it swaps positions with another (half an exchange) it acquires a phase factor of $e^{\pi i/3}$ independent of the geometric shape of the trajectory. Such anyonic behavior was seen experimentally by Daniel Tsui, Horst Störmer and Art Gossard in 1982 [497] and a theory accounting for how fractional charges can arise was provided by Robert Laughlin in 1983 [305].

15.6.3 Abelian Versus Non-Abelian Anyons

From our discussion of quantum gates in Chap. 2, we can view such a change in phase factor when one anyon encircles another as the application of a *phase gate* to the quantum state of the anyon. Interpreting such topological braiding operations as the application of quantum phase gates invites one to speculate on whether other non-phase gate operations might also be implementable via similar braiding operations? It turns out that they can [123, 285, 373, 398]. However, the utility of such braiding operations for quantum computation depends upon the type of anyon used.

If the quantum state of an anyon is changed according to a unitary matrix U when its worldline is braided around those of other anyons, then the usefulness of this for quantum computation depends upon whether this operation commutes with preceding or succeeding braiding operations. Specifically, if the unitary matrix describing the state change is a phase gate then the order in which a sequence of braiding operations is performed is immaterial because all the resulting phases gates commute with one another. However, if the state change of the anyon must instead be described by some gate other than a phase gate, then the possibility arises that the operations resulting from a sequence of braids do not commute. This is the key to achieving non-trivial quantum computations. If the quantum gates corresponding to braiding operations of the anyons do not commute they are called non-Abelian anyons. Not surprisingly, it is these non-Abelian anyons that we require for quantum computing.

15.6.4 Quantum Gates by Braiding Non-Abelian Anyons

In 1997 Alexei Kitaev was the first to realize how to achieve universal quantum computation by braiding non-Abelian anyons. These ideas were later extended by R. Walter Ogburn and John Preskill [373] and Mochon [357, 358]. In original Kitaev scheme measurements were needed to perform some quantum gates. However, Freedman, Larsen and Wang showed that if you use the right kind of anyons then all measurements can be postponed until the final readout at the end of the computation [190].

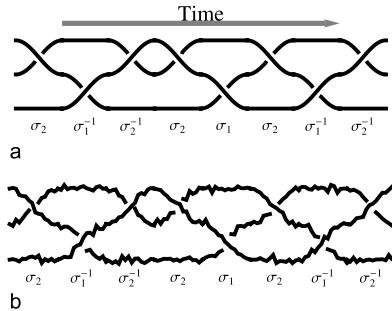


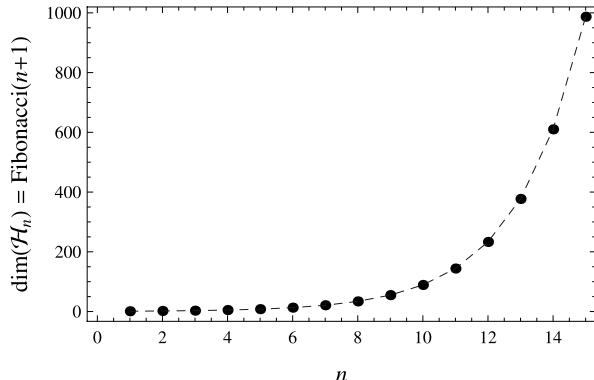
Fig. 15.13 By using braidings of the worldlines of non-Abelian anyons to achieve unitary quantum gates, the geometric trajectories can be fairly irregular without it affecting the overall unitary gate that is achieved. The non-Abelian anyons must nevertheless be kept sufficiently far apart during the braiding operations to suppress certain effects that would introduce errors

When non-Abelian anyons are exchanged the change in state is described as a unitary operation, i.e., as a quantum gate. Thus, certain unitary gates can be achieved by braiding one anyon around another in a carefully orchestrated pattern. This idea was developed in detail for Fibonacci anyons by Nicholas Bonesteel, Layla Hormozi, and Georgios Zikos in 2005 [65] and generalized to a wider class of non-Abelian anyons by Hormozi, Bonesteel, and Steve Simon in 2009 [238].

The basic idea behind topological quantum computing, as illustrated in Fig. 15.13, is that quantum information is stored in so-called quasi-particles (non-Abelian anyons) that are intrinsically immune to decoherence, and that quantum gates can be effected by dragging these non-Abelian anyons around one another on a two dimensional surface. We can visualize the trajectories the quasi-particles take by drawing their worldlines in a $2 + 1$ dimensional spacetime. The exact trajectories the quasi-particles take is unimportant provided they are not allowed to move too close to one another. Thus, one does not need heroic control over the particle trajectories in order to effect an accurate, and intrinsically error-immune, quantum gate.

Fibonacci anyons possess a new kind of spin quantum number called q-spin. Individual Fibonacci anyons have a “q-spin” of 1, but combinations of them can have a q-spin of 0 or 1. We indicate the total q-spin when combining q-spin objects with a subscript as in $(\bullet, \bullet)_a$. Just as there are rules for determining the net spin for composite particles built from ordinary fermions and bosons, so too are there rules—called fusion rules—for determining the overall q-spin of combinations of Fibonacci anyons. In particular, when a pair of Fibonacci anyons having q-spin 1 are combined the net q-spin can be either 0 or 1, but when a Fibonacci anyon with q-spin s is combined with one of q-spin 1, the net q-spin is always s . These fusion rules determine the dimensionality of the Hilbert space of n Fibonacci anyons. In particular, if there are n Fibonacci anyons, the dimension of the Hilbert space is the $(n + 1)$ -st Fibonacci number (hence their name). This dimension grows exponentially with the number of anyons as shown in Fig. 15.14, which allows plenty of room for embedding quantum information processing operations. Thus, the Hilbert

Fig. 15.14 The dimension, $\dim(\mathcal{H}_n)$, of the Hilbert space, \mathcal{H}_n , of n Fibonacci anyons grows as $(n + 1)$ -st Fibonacci number, i.e., exponentially in n . This creates a very large Hilbert space very quickly providing ample room for embedding quantum information processing operations



space of two Fibonacci anyons is two dimensional with basis states $|(\bullet, \bullet)_0\rangle$ and $|(\bullet, \bullet)_1\rangle$, and that of three Fibonacci anyons is three dimensional with basis states $|((\bullet, \bullet)_0, \bullet)_1\rangle$, $|((\bullet, \bullet)_1, \bullet)_1\rangle$, and $|((\bullet, \bullet)_1, \bullet)_0\rangle$. We take our logical qubits to be those triples of Fibonacci anyons having a total q-spin of 1, i.e.,

$$|0_L\rangle = |((\bullet, \bullet)_0, \bullet)_1\rangle \quad (15.6)$$

$$|1_L\rangle = |((\bullet, \bullet)_1, \bullet)_1\rangle \quad (15.7)$$

and we disregard $|((\bullet, \bullet)_1, \bullet)_0\rangle$ as an irrelevant “non-computational” state.

Quantum gate operations can be performed on these logical qubits’ by *braiding* their component anyons around one another. As Fibonacci anyons are non-Abelian, these braiding operations have the effect of applying unitary transformations more complicated than mere phase shifts to the state of the logical qubits. As three anyons make a single logical qubit, the two basic braiding operations involve either braiding particles 1 and 2, or braiding particles 2 and 3 as shown in Fig. 15.15. To interpret this figure, realize that the Fibonacci anyons are confined to move on a two-dimensional surface, and we track how one anyon is made to encircle another by plotting the braiding of their world lines in a 2 + 1 dimensional spacetime, with time flowing from left to right in the figures. In particular, the operation σ_1 describes a braid between particles 2 and 3 in the clockwise direction along the worldline. Similarly, σ_2 describes a braid between particles 1 and 2 in the clockwise direction along the worldline. The corresponding braids in the anti-clockwise direction are described by σ_1^{-1} and σ_2^{-1} respectively.

These elementary braiding operations have the effect of applying the following unitary transformations to the logical qubits [65]:

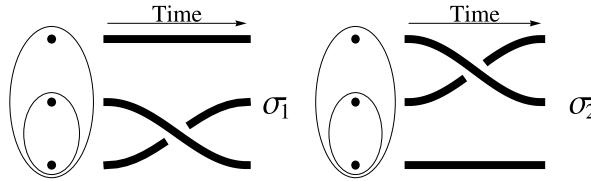


Fig. 15.15 The two basic braids amongst the worldlines of three Fibonacci (non-Abelian) anyons. σ_1 describes a braid between particles 2 and 3, and σ_2 describes a braid between particles 1 and 2, in the clockwise direction along the worldline. The corresponding braids in the anti-clockwise direction along the wordlines of the particles are described by σ_1^{-1} and σ_2^{-1} respectively. In these matrices, the constant $\tau = 1/\phi$ where $\phi = \frac{1}{2}(1 + \sqrt{5})$, i.e., the golden mean. This gives the irrational angle needed to create a universal gate set from a finite set of angles

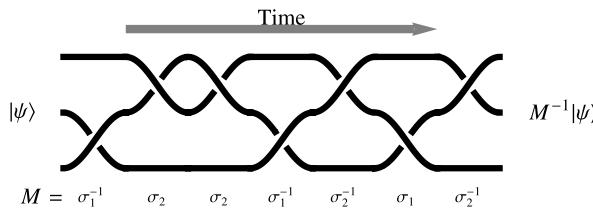


Fig. 15.16 A sequence of braiding operations is equivalent to a unitary transformation applied to the logical qubits. If the braiding occurs in the order $\sigma_1^{-1} \rightarrow \sigma_2 \rightarrow \sigma_2 \rightarrow \sigma_1^{-1} \rightarrow \sigma_2^{-1} \rightarrow \sigma_1 \rightarrow \sigma_2^{-1}$, the unitary operator achieved is $M^{-1} = \sigma_2^{-1} \cdot \sigma_1 \cdot \sigma_2^{-1} \cdot \sigma_1^{-1} \cdot \sigma_2 \cdot \sigma_2 \cdot \sigma_1^{-1}$

$$\begin{aligned} \sigma_1 &= \left(\begin{array}{cc|c} e^{-4\pi i/5} & 0 & 0 \\ 0 & -e^{-2\pi i/5} & 0 \\ \hline 0 & 0 & -e^{-2\pi i/5} \end{array} \right) \\ \sigma_2 &= \left(\begin{array}{cc|c} -\tau e^{-\pi i/5} & -i\sqrt{\tau} e^{-\pi i/10} & 0 \\ -i\sqrt{\tau} e^{-\pi i/10} & -\tau & 0 \\ \hline 0 & 0 & -e^{-2\pi i/5} \end{array} \right) \end{aligned} \quad (15.8)$$

Note that, in these matrices, the upper 2×2 blocks define the computational space (having total q-spin 1) and correspond to single qubit gates. The lower right element in each case applies a phase factor on the non-computational state in the Hilbert space, which we do not care about. By arranging for an irrational angle to appear in these matrices we can obtain a finite set of fixed angle gates that are, when augmented with a topological CNOT, universal for quantum computing.

To obtain other single qubit gates we can cascade the four elementary braiding operations, σ_1 , σ_1^{-1} , σ_2 , and σ_2^{-1} , as illustrated in Fig. 15.16. For example, the braiding sequence $\sigma_1^{-1} \rightarrow \sigma_2 \rightarrow \sigma_2 \rightarrow \sigma_1^{-1} \rightarrow \sigma_2^{-1} \rightarrow \sigma_1 \rightarrow \sigma_2^{-1}$ effects the unitary operator corresponding to the dot product of the elementary braiding operators in the reverse order. An explicit example is the topological identity gate. The braid sequence for the topological identity operation and its corresponding unitary trans-

formation is:

$$\begin{aligned} \sigma_2^3 &\rightarrow \sigma_1^{-2} \rightarrow \sigma_2^{-4} \rightarrow \sigma_1^2 \rightarrow \sigma_2^4 \rightarrow \sigma_1^2 \rightarrow \sigma_2^{-2} \rightarrow \sigma_1^{-2} \rightarrow \sigma_2^{-4} \rightarrow \sigma_1^{-4} \rightarrow \sigma_2^{-2} \\ &\rightarrow \sigma_1^4 \rightarrow \sigma_2^2 \rightarrow \sigma_1^{-2} \rightarrow \sigma_2^2 \rightarrow \sigma_1^2 \rightarrow \sigma_2^{-2} \rightarrow \sigma_1^3 \approx \left(\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right) \end{aligned} \quad (15.9)$$

This braid is illustrated in Fig. 15.17.

Similarly, the braid sequence for the NOT operation and its corresponding unitary transformation is given by:

$$\begin{aligned} \sigma_1^{-2} &\rightarrow \sigma_2^{-4} \rightarrow \sigma_1^4 \rightarrow \sigma_2^{-2} \rightarrow \sigma_1^2 \rightarrow \sigma_2^2 \rightarrow \sigma_1^{-2} \rightarrow \sigma_2^4 \rightarrow \sigma_1^{-2} \rightarrow \sigma_2^4 \rightarrow \sigma_1^2 \\ &\rightarrow \sigma_2^{-4} \rightarrow \sigma_1^2 \rightarrow \sigma_2^{-2} \rightarrow \sigma_1^2 \rightarrow \sigma_2^{-2} \rightarrow \sigma_1^{-2} \approx \left(\begin{array}{cc|c} 0 & i & 0 \\ i & 0 & 0 \\ 0 & 0 & 1 \end{array} \right) \end{aligned} \quad (15.10)$$

and its corresponding braid diagram is shown in Fig. 15.18.

To achieve a universal gate set, we obviously need to have some kind of conditional quantum gate too. These can be achieved topologically by weaving the quasi-particles from a control qubit through those of a target qubit as illustrated in Fig. 15.18. In the figure the top three Fibonacci anyons (which can move) form the “control” qubit and the bottom three (which are static) the “target” qubit. By physically dragging a pair of anyons from the control qubit around the three anyons in the target qubit in a specific way, and then returning them to the control qubit, we can achieve a CNOT operation between the pair of logical qubits. Such constructions can be made arbitrarily accurate by lengthening the braid sequences. It is thought that such topological quantum operations could provide a viable route to error-free quantum computation.

The final step in any quantum computation is to extract any answer, usually by reading some quantum memory register in the computational basis. How are such reading operations to be performed in the context of topological quantum computing? Well as we are working with logical qubits, our task is to distinguish between the $|0_L\rangle$ and $|1_L\rangle$ as defined in (15.7). You will notice that these two logical states differ in the total q-spin of the two leftmost quasi-particles in the triplet making up the logical qubit. So to perform a measurement to decide if we have $|0_L\rangle$ or $|1_L\rangle$ we

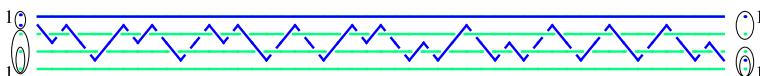


Fig. 15.17 Braiding scheme for a topological identity operation using Fibonacci anyons

Fig. 15.18 Braiding scheme for a topological NOT operation using Fibonacci anyons



attempt to “fuse” the two leftmost quasi-particles within the qubit. If the total q-spin of the leftmost pair is 0 they can fuse back to the vacuum state. However, if their total q-spin is 1 this is impossible. This will result in a measurable charge difference in the final state between $|0_L\rangle$ and $|1_L\rangle$. By reading each logical qubit in this manner we can essentially readout the answer from our topological quantum computer. Thus, the “Prepare-Evolve-Measure” cycle of conventional quantum computing becomes an “Initialize-Braid-Probe” cycle in topological quantum computing.

15.6.5 Do Non-Abelian Anyons Exist?

Experimental evidence for *Abelian* anyons has been seen in a Laughlin quasi-particle interferometer experiment conducted by Fernando Camino, Wei Zhou, and Vladimir J. Goldman [96]. In this experiment quasi-particles of the 1/3 fractional quantum Hall fluid were made to traverse a closed path around an island of the 2/5 fluid and thereby accumulate a phase, which was detected interferometrically by inducing oscillations in conductance. These results constituted the first direct experimental observation of fractional statistics of Laughlin quasi-particles.

Non-Abelian anyons are thought to exist in the fractional quantum Hall state [548], in rotating Bose condensates [116, 238], in quantum spin systems [191] and in superconductors. Bonesteel et al. have suggested that Fibonacci anyons may exist in an experimentally observed fractional quantum Hall state having filling fraction $\nu = 12/5$ [548]. However, finding compelling experimental evidence for the existence of *non-Abelian* anyons, including the Fibonacci and Read-Rezayi anyons, has been challenging. However, there are now several detailed experimental proposals for doing so [63, 64, 124, 481].

15.7 Adiabatic Quantum Computing

The models of quantum computing we have presented so far have leaned heavily on the ideas of quantum gates and circuits. An altogether different view of quantum computation—called Adiabatic Quantum Computing (AQC)—was devised by E. Farhi, J. Goldstone, S. Gutmann [177] and published in Science in 2001 [178]. Since that time concrete hardware designs have been proposed [265] and sophisticated multi-qubit implementations have been achieved [229]. There is now good evidence that these chips are harnessing genuine quantum mechanical phenomena to effect computation [229, 303].

Adiabatic Quantum Computing is based on the “Adiabatic Theorem” of quantum mechanics [434]. This states that if a quantum system is prepared in the ground state of a time-independent Hamiltonian \mathcal{H}_0 , and if we then cause the Hamiltonian to change from \mathcal{H}_0 to \mathcal{H}_1 in time T , e.g., by driving it linearly according to

$$\mathcal{H}(t) = \left(1 - \frac{t}{T}\right)\mathcal{H}_0 + \frac{t}{T}\mathcal{H}_1 \quad (15.11)$$

then the system will remain in the ground state of all the instantaneous Hamiltonians, $\mathcal{H}(t)$, passed through provided the change is made sufficiently slowly, i.e., *adiabatically*. Thus, if the final Hamiltonian, \mathcal{H}_1 , can be made to encode a computational problem such that the ground state of \mathcal{H}_1 corresponds to a *solution* to this problem, then the natural quantum mechanical evolution of the system under the slowly changing Hamiltonian $\mathcal{H}(t)$ would carry the initial state (the ground state of \mathcal{H}_0) into the final state (the ground state of \mathcal{H}_1), at which point a measurement would reveal the ground state and hence a solution to the computational problem encoded in \mathcal{H}_1 .

This is perhaps a little abstract so let's look at a concrete example. In particular, using an example developed by Richard Harris et al. [228] suppose we wish to compute the vector of values $\vec{s} = \{s_1, s_2, \dots, s_N\}$ that minimizes the objective function

$$E(\vec{s}) = - \sum_{i=1}^N h_i s_i + \sum_{i < j} K_{ij} s_i s_j \quad (15.12)$$

where $s_i = \pm 1$, N is the length of \vec{s} , and the values of h_i and K_{ij} are real numbers that vary depending on the particular problem instance we wish to solve. This is an example of a discrete combinatorial optimization problem, which is known to be NP-hard, and hence, cannot be solved in worst case polynomial time in all cases. Moreover, this type of problem is commonplace in many practical applications and finding a way to solve it that is superior to conventional methods would be very useful.

To pose this problem to an adiabatic quantum computer we need to pick a pair of Hamiltonians, \mathcal{H}_0 and \mathcal{H}_1 , such that the ground state of \mathcal{H}_0 is easy to obtain and unbiased and the ground state of \mathcal{H}_1 encodes a solution to (15.12).

Let us begin by defining a set of qubits, $\{|q_1\rangle, |q_2\rangle, \dots, |q_N\rangle\}$, to hold the answer to our computation. Without loss of generality we can assume the computational basis for these qubits is aligned with the z -axis. We will interpret $q_i = 0$ to mean $s_i = -1$ and $q_i = 1$ to mean $s_i = +1$ in the globally optimal solution to (15.12). Initially, we do not know what value each qubit will take on in a solution so it is reasonable to initialize each qubit in an equally weighted superposition of $|0\rangle$ and $|1\rangle$, e.g., $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Thus, the initial state of all N qubits, $|\psi(0)\rangle$, is given by:

$$\begin{aligned} |\psi(0)\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \cdots \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ &= \frac{1}{\sqrt{2^N}} \sum_{q_1=0}^1 \sum_{q_2=0}^1 \cdots \sum_{q_N=0}^1 |q_1\rangle |q_2\rangle \cdots |q_N\rangle \\ &= \frac{1}{\sqrt{2^N}} \sum_{i=0}^{2^N-1} |i\rangle \end{aligned} \quad (15.13)$$

in our standard notation. This state is to be the ground state of our initial Hamiltonian \mathcal{H}_0 . A suitable Hamiltonian having $|\psi(0)\rangle$ as a ground state is $\mathcal{H}_0 =$

$-\sum_{i=1}^N \sigma_x^{(i)}$ where $\sigma_x^{(i)}$ is the Pauli spin matrix

$$\sigma_x^{(i)} = \underbrace{\sigma_x \otimes \cdots \otimes \sigma_x}_{i-1 \text{ terms}} \otimes \mathbb{1} \otimes \underbrace{\sigma_x \otimes \cdots \otimes \sigma_x}_{N-i \text{ terms}} \quad (15.14)$$

Hence, let's take our initial Hamiltonian \mathcal{H}_0 to be:

$$\mathcal{H}_0 = - \sum_{i=1}^N \sigma_x^{(i)} \quad (15.15)$$

Next we have to construct a Hamiltonian \mathcal{H}_1 such that its ground state encodes a globally optimal solution to (15.12). Our particular problem instance can be mapped onto a Hamiltonian in a straightforward way. Simply define \mathcal{H}_1 by replacing each s_i in (15.12) with a corresponding Pauli spin matrix

$$\sigma_z^{(i)} = \underbrace{\sigma_z \otimes \cdots \otimes \sigma_z}_{i-1 \text{ terms}} \otimes \mathbb{1} \otimes \underbrace{\sigma_z \otimes \cdots \otimes \sigma_z}_{N-i \text{ terms}} \quad (15.16)$$

This gives us:

$$\mathcal{H}_1 = - \sum_{i=1}^N h_i \sigma_z^{(i)} + \sum_{i < j} K_{ij} \sigma_z^{(i)} \sigma_z^{(j)} \quad (15.17)$$

which is immediately recognizable as the Hamiltonian of an Ising spin system in which the spins have individual biases (given by h_i) and pairwise couplings given by K_{ij} . Thus, in the computational-basis, the ground state of \mathcal{H}_1 corresponds to a configuration of spins $|q_1 q_2 \cdots q_N\rangle$ that minimizes the net energy. Thus, if the ground state is found to be $|q_1 q_2 \cdots q_N\rangle$ then substituting $s_i = -1$ if $q_i = 0$ and $s_i = +1$ if $q_i = 1$ gives the vector of assignments $\vec{s} = \{s_1, s_2, \dots, s_N\}$ that minimizes the objective function of $E(\vec{s})$ shown in (15.12).

Thus, choosing the Hamiltonians \mathcal{H}_0 and \mathcal{H}_1 as:

$$\begin{aligned} \mathcal{H}_0 &= - \sum_{i=1}^N \sigma_x^{(i)} \\ \mathcal{H}_1 &= - \sum_{i=1}^N h_i \sigma_z^{(i)} + \sum_{i < j} K_{ij} \sigma_z^{(i)} \sigma_z^{(j)} \end{aligned} \quad (15.18)$$

and choosing the initial state to be $|\psi(0)\rangle = \frac{1}{\sqrt{2^N}} \sum_{i=0}^{2^N-1} |i\rangle$, then after a slow enough interpolation lasting time T the state will evolve into $|\psi(T)\rangle = |q_1 q_2 \cdots q_N\rangle$. Reading this state in the computational basis then reveals each bit value from which the corresponding values of the s_i can be obtained.

Specialized adiabatic quantum computer hardware has been developed that is able to solve discrete combinatorial optimization problems of type given

above [229]. This hardware has been used to solved image matching [366] and machine learning problems [367, 368] by posing the problems in terms of minimizing an objective function of the form given in (15.12).

A key question from a computational complexity perspective is how quickly one can drive the interpolation between the initial and final Hamiltonians while keeping the system in the ground state of the instantaneous Hamiltonians passed through. If the shortest feasible interpolation time scales polynomially with increasing problem size, the quantum adiabatic algorithm would be deemed “efficient”, otherwise it would be deemed “inefficient”. It is worth noting that even if the quantum adiabatic algorithm proves to be “inefficient” by this measure it may, nevertheless, still be faster than any known classical algorithm, e.g., by admitting a polynomial speedup. Indeed, it is known that an adiabatic quantum computer can solve unstructured search problems in square root of the time required classically [421]. This echoes Grover’s result but in the context of adiabatic quantum computing.

To estimate the maximum feasible interpolation rate we can ask under what conditions the passage from \mathcal{H}_0 and \mathcal{H}_1 can be performed adiabatically [434]. It is found that if the minimum eigenvalue gap between the ground state E_0 and first excited state E_1 of the instantaneous Hamiltonians $H(t)$ is given by g_{\min} , where

$$g_{\min} = \min_{0 \leq t \leq T} [E_1(t) - E_0(t)] \quad (15.19)$$

and the matrix element between the corresponding pair of eigenstates is

$$\left\langle \frac{d\mathcal{H}}{dt} \right\rangle_{1,0} = \langle E_1; t | \frac{d\mathcal{H}}{dt} | E_0; t \rangle, \quad (15.20)$$

then the Adiabatic Theorem asserts that the final state will be very close to the ground state of $\mathcal{H}_1(T)$, i.e.,

$$|\langle E_0; T | \psi(T) \rangle|^2 \geq 1 - \epsilon^2 \quad (15.21)$$

provided that

$$\frac{|\langle \frac{d\mathcal{H}}{dt} \rangle_{1,0}|}{g_{\min}^2} \leq \epsilon \quad (15.22)$$

where $\epsilon \ll 1$. If this criterion is met, we can be sure the system will evolve into the desired state, i.e., the ground state of \mathcal{H}_1 as desired. Thus the maximum rate at which we can drive the Adiabatic Algorithm is dependent upon the gap size between the ground state and first excited state of the instantaneous Hamiltonians passed through.

Less is known about the formal complexity of the quantum adiabatic approach than the quantum circuit approach applied to different computational problems. It is known that an adiabatic version of the unstructured search algorithm exists [421] and runs in $\mathcal{O}(\sqrt{N})$ time, where N is the number of items in the database. This matches the complexity of the standard Grover algorithm, but requires a non-uniform interpolation rate between the initial and final Hamiltonians. Moreover, by

nesting one adiabatic search within another, structured search problems may also be sped up by a polynomial factor [422]. However, exponential speedups are harder to come by [355, 427]. Initially, based on preliminary numerical evidence of the scaling of the adiabatic algorithm used to solve small instances of NP-Complete problems, the inventors of adiabatic quantum computing speculated that the running time might be polynomial. However, this was formally disproved for the case of a linear interpolation path by Wim van Dam, Michele Mosca and Umesh Vazirani [506]. They provided a counter-example that provably could not be solved in polynomial time using the linear interpolation path. Farhi et al. later countered this by showing an alternative interpolation path could work efficiently on cases when the linear interpolation path did not [177]. Nevertheless, today there remains much work to be done on developing a proper complexity theory for adiabatic quantum computing. However, it remains an interesting model of quantum computation given the relative ease of implementation compared to the circuit model, and the fact that it seems naturally well suited to discrete combinatorial optimization type problems. In these cases the “correctness” of the solution may not be the primary concern. Rather one is interested in understanding the tradeoff between running time of the algorithm and quality of solution obtained [417]. It is likely there are distinctions between the quantum and classical cases here, but this is not well understood at this time. Moreover, the theory of quantum error correction is much less well developed for adiabatic quantum computing than it is for the circuit model [260, 313].

15.8 Encoded Universality Using Only Spin-Spin Exchange Interactions

In quantum computing, the tradition has been to decompose any desired unitary transformation into a quantum circuit containing only 1-qubit and 2-qubit quantum logic gates. The most famous universal set of gates is the set of all 1-qubit gates together with controlled-NOT (CNOT). However, many other universal sets are possible. In fact, almost any 2-qubit gate (alone) is universal for quantum computation [147]. From a practical perspective, if we could find a particular universal 2-qubit gate that was easy to make, this would greatly simplify the fabrication of large scale quantum circuits.

15.8.1 The Exchange Interaction

Different physical implementations of quantum computation are able to achieve some gates more easily than others. Spintronics is thought by many to provide a viable path to scalable universal quantum computation [26].

In spintronic quantum computers, the simplest elementary operation is the 2-qubit exchange interaction [26]. The Hamiltonian for the exchange interaction has

the form:

$$\mathcal{H} = \sigma^{(j)} \cdot \sigma^{(k)} \quad (15.23)$$

where

$$\sigma^{(j)} = (\sigma_x^{(j)}, \sigma_y^{(j)}, \sigma_z^{(j)}) \quad (15.24)$$

and

$$\begin{aligned} \sigma_x^{(j)} &= \underbrace{\mathbb{1} \otimes \mathbb{1} \otimes \cdots \otimes}_{j-1} \sigma_x \underbrace{\otimes \cdots \otimes \mathbb{1} \otimes \mathbb{1}}_{n-j} \\ \sigma_y^{(j)} &= \underbrace{\mathbb{1} \otimes \mathbb{1} \otimes \cdots \otimes}_{i-1} \sigma_y \underbrace{\otimes \cdots \otimes \mathbb{1} \otimes \mathbb{1}}_{n-j} \\ \sigma_z^{(j)} &= \underbrace{\mathbb{1} \otimes \mathbb{1} \otimes \cdots \otimes}_{j-1} \sigma_z \underbrace{\otimes \cdots \otimes \mathbb{1} \otimes \mathbb{1}}_{n-j} \end{aligned} \quad (15.25)$$

Thus, the exchange interaction induces the unitary evolution:

$$U_{j,k}(\alpha) = e^{-i\alpha\mathcal{H}} = e^{-i\alpha(\sigma^{(j)} \cdot \sigma^{(k)})} \quad (15.26)$$

where $i = \sqrt{-1}$.

15.8.2 SWAP^α via the Exchange Interaction

What kinds of gates can we make easily if we have access to *only* a spin-spin exchange interaction? Not surprisingly, we can achieve the 2-qubit SWAP^α gate which was introduced in Sect. 2.7.1. This gate is essentially as powerful as CNOT since the set of all 1-qubit gates together with SWAP^α is universal for quantum computing.

$$\text{SWAP}^\alpha = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}(1 + e^{i\pi\alpha}) & \frac{1}{2}(1 - e^{i\pi\alpha}) & 0 \\ 0 & \frac{1}{2}(1 - e^{i\pi\alpha}) & \frac{1}{2}(1 + e^{i\pi\alpha}) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (15.27)$$

The proof is by way of the following matrix identity. A U gate between the j -th and k -th of n qubits is:

$$\begin{aligned} U_{j,k;n}(\alpha) &= e^{-i\alpha(\sigma^{(j)} \cdot \sigma^{(k)})} \\ &= \begin{pmatrix} e^{-i\alpha} & 0 & 0 & 0 \\ 0 & \frac{1}{2}e^{-i\alpha}(1 + e^{4i\alpha}) & -\frac{1}{2}e^{-i\alpha}(-1 + e^{4i\alpha}) & 0 \\ 0 & -\frac{1}{2}e^{-i\alpha}(-1 + e^{4i\alpha}) & \frac{1}{2}e^{-i\alpha}(1 + e^{4i\alpha}) & 0 \\ 0 & 0 & 0 & e^{-i\alpha} \end{pmatrix} \end{aligned} \quad (15.28)$$

$$= e^{-i\alpha} \text{SWAP}_{j,k;n}^{\frac{4\alpha}{\pi}}$$

Thus, if you can implement an exchange interaction between the j -th and k -th of n qubits you can implement a SWAP^α gate, up to an overall phase factor. Hence, in spintronics, obtaining a powerful 2-qubit gate, SWAP^α , appears to be relatively straightforward.

15.8.3 Problem: Although SWAP^α Is Easy 1-Qubits Gates Are Hard

“Spintronic” implementations of quantum computing harness the quantum mechanical spin of particles as the carriers of quantum information, and can make use of spin-spin exchange interactions to achieve a useful 2-qubit gate such as SWAP^α . However, ironically, in such schemes it can be surprisingly difficult to achieve arbitrary *one*-qubit gates [92, 326]!

It occurred to people to wonder whether or not it might be possible to work in a larger Hilbert space in which several spins were used to encode each qubit and which used only the spin-spin exchange interaction to implement both the 1-qubit and 2-qubit gates in this “encoded basis”. If this were possible it would mean that the exchange interaction alone could be used as the basis for implementing universal quantum computation.

15.8.4 Solution: Use an Encoded Basis

If one uses an encoded basis in which sets of three physical spins are used to encode the state of each logical qubit, Julia Kempe, Dave Bacon, Daniel Lidar and Birgitta Whaley showed that, in principle, universal quantum computation was possible [271, 272] and in principle a decoherence free subspace can be formed [28, 29, 314–318, 520]. However, their proof was not constructive. Although later work by Hsieh, Kempe, Mygren, and Whaley [240] demonstrated a constructive method to achieve an explicit *fixed* universal set of gates in the encoded basis, the chosen set gives very inefficient decompositions of arbitrary 1-qubit gates. Moreover, to date there has only been numerical evidence that when exchange interactions are run sequentially (i.e., so that at most one exchange interaction is active at any time), then *four* exchange gates appear sufficient to achieve an arbitrary 1-qubit gate.

In this section we improve upon these results. First we provide a *constructive* scheme for mapping *any* 1-qubit gate in the logical basis into an equivalent sequence of at most four spin-spin exchange interactions in the physical (encoded) basis. Second we provide a formal proof that four such exchange interactions are, indeed, sufficient (as DiVincenzo et al. has suspected). Our results make it very easy to map any quantum circuit consisting of standard 1-qubit gates and CNOT gates

into an encoded, so-called “decoherence-free” form. This will be of use to hardware experimentalists working on spintronic implementations of quantum logic.

Since the exchange interaction by itself is not a universal gate, we use the encoding invented by David DiVincenzo, Dave Bacon, Julia Kempe, Guido Burkard, and Birgitta Whaley to represent logical qubits [150]:

$$|0_L\rangle = \frac{1}{\sqrt{2}}(|010\rangle - |100\rangle), \quad (15.29)$$

$$|1_L\rangle = \sqrt{\frac{2}{3}}|001\rangle - \frac{1}{\sqrt{6}}|010\rangle - \frac{1}{\sqrt{6}}|100\rangle. \quad (15.30)$$

The vectors $|0_L\rangle$ and $|1_L\rangle$ generate a 2-dimensional subspace \mathcal{L} of the 8-dimensional Hilbert space $\mathbb{C}^{\otimes 8}$. This subspace can be characterized as the set of vectors $(0, \alpha, \beta, 0, \gamma, 0, 0, 0)$ such that $\alpha + \beta + \gamma = 0$.

15.8.5 $U_{\mathcal{L}}^{1,2}$, $U_{\mathcal{L}}^{2,3}$, and $U_{\mathcal{L}}^{1,3}$

The exchange interaction operations on the Hilbert space $\mathbb{C}^{\otimes 8}$ can be written explicitly as follows:

$$U^{1,2}(t) = \exp[-it(\sigma_x \otimes \sigma_x \otimes \mathbb{1}_2 + \sigma_y \otimes \sigma_y \otimes \mathbb{1}_2 + \sigma_z \otimes \sigma_z \otimes \mathbb{1}_2)] \quad (15.31)$$

$$U^{1,3}(t) = \exp[-it(\sigma_x \otimes \mathbb{1}_2 \otimes \sigma_x + \sigma_y \otimes \mathbb{1}_2 \otimes \sigma_y + \sigma_z \otimes \mathbb{1}_2 \otimes \sigma_z)] \quad (15.32)$$

$$U^{2,3}(t) = \exp[-it(\mathbb{1}_2 \otimes \sigma_x \otimes \sigma_x + \mathbb{1}_2 \otimes \sigma_y \otimes \sigma_y + \mathbb{1}_2 \otimes \sigma_z \otimes \sigma_z)] \quad (15.33)$$

15.1 Lemma *The subspace \mathcal{L} spanned by $|0_L\rangle$ and $|1_L\rangle$ is closed under each of the unitary operations $U^{1,2}(t)$, $U^{1,3}(t)$, and $U^{2,3}(t)$.*

We denote the unitary operations on the subspace \mathcal{L} induced by $U^{1,2}(t)$, $U^{1,3}(t)$, and $U^{2,3}(t)$ as $U_{\mathcal{L}}^{1,2}(t)$, $U_{\mathcal{L}}^{1,3}(t)$, and $U_{\mathcal{L}}^{2,3}(t)$, respectively. Then the matrix representations of these new unitary operations, with respect to the basis $\{|0_L\rangle, |1_L\rangle\}$, are as follows:

$$U_{\mathcal{L}}^{1,2}(t) = \begin{pmatrix} e^{3it} & 0 \\ 0 & e^{-it} \end{pmatrix} \quad (15.34)$$

$$U_{\mathcal{L}}^{1,3}(t) = \frac{1}{4}e^{-it} \begin{pmatrix} 3 + e^{4it} & \sqrt{3}(-1 + e^{4it}) \\ \sqrt{3}(-1 + e^{4it}) & 1 + 3e^{4it} \end{pmatrix} \quad (15.35)$$

$$U_{\mathcal{L}}^{2,3}(t) = \sigma_z \cdot U_{\mathcal{L}}^{1,3}(t) \cdot \sigma_z. \quad (15.36)$$

15.8.6 R_z Gates in Encoded Basis

Therefore, the $U_{\mathcal{L}}^{1,2}(t)$ operation is equivalent to an $R_z(\cdot)$ gate up to an overall phase factor. In particular, we have:

$$U_{\mathcal{L}}^{1,2}(t) = e^{i t} R_z(-4t) = e^{i t} R_z(4(\pi - t)) \quad (15.37)$$

15.8.7 R_x Gates in Encoded Basis

To obtain an $R_x(\cdot)$ gate we recall:

$$R_x(t) = H \cdot R_z(t) \cdot H \quad (15.38)$$

where H is the Walsh-Hadamard gate:

$$H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \quad (15.39)$$

However, the Walsh-Hadamard gate can be implemented using a sequence of three exchange interactions as:

$$\begin{aligned} U_{\mathcal{L}}^{1,2}(a) \cdot U_{\mathcal{L}}^{2,3}(b) \cdot U_{\mathcal{L}}^{1,2}(c) \\ = \begin{pmatrix} \frac{1}{4}e^{i(3a-b+3c)}(3 + e^{4ib}) & -\frac{1}{4}\sqrt{3}e^{i(3a-b-c)}(-1 + e^{4ib}) \\ -\frac{1}{4}\sqrt{3}e^{-i(a+b-3c)}(-1 + e^{4ib}) & \frac{1}{4}e^{-i(a+b+c)}(1 + 3e^{4ib}) \end{pmatrix} \end{aligned} \quad (15.40)$$

Equating matrices and solving for a , b , and c we have:

$$\begin{aligned} & \begin{pmatrix} \frac{1}{4}e^{i(3a-b+3c)}(3 + e^{4ib}) & -\frac{1}{4}\sqrt{3}e^{i(3a-b-c)}(-1 + e^{4ib}) \\ -\frac{1}{4}\sqrt{3}e^{-i(a+b-3c)}(-1 + e^{4ib}) & \frac{1}{4}e^{-i(a+b+c)}(1 + 3e^{4ib}) \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \end{aligned} \quad (15.41)$$

which implies (as you will verify in Problem 15.8:

$$\begin{aligned} a &= \frac{\pi}{2} + \frac{1}{8} \left(3\pi + 2 \cos^{-1} \left(\sqrt{\frac{2}{3}} \right) \right) \\ b &= \frac{1}{4} \cos^{-1} \left(-\frac{1}{3} \right) \end{aligned} \quad (15.42)$$

$$c = \frac{1}{8} \left(3\pi + 2 \cos^{-1} \left(\sqrt{\frac{2}{3}} \right) \right)$$

With these parameter values, direct substitution and simplification shows that the Walsh-Hadamard gate can be obtained from:

$$H = U_{\mathcal{L}}^{1,2}(a) \cdot U_{\mathcal{L}}^{2,3}(b) \cdot U_{\mathcal{L}}^{1,2}(c) \quad (15.43)$$

Thus, using the exchange-only implementations of H and $R_z(\cdot)$ we can achieve an arbitrary $R_x(\cdot)$ gate up to an overall phase factor too.

15.8.8 R_y Gates in Encoded Basis

Finally, an arbitrary $R_y(\cdot)$ can be obtained from:

$$R_y(t) = R_z\left(\frac{\pi}{2}\right) \cdot H \cdot R_z(t) \cdot H \cdot R_z\left(-\frac{\pi}{2}\right) \quad (15.44)$$

Thus we can use the exchange-only implementations of H and $R_z(\cdot)$ to achieve an arbitrary $R_y(\cdot)$ gate up to an overall phase factor too.

As any general 1-qubit gate can be written in the form $R_z(\alpha) \cdot R_y(\beta) \cdot R_z(\gamma) \cdot Ph(\delta)$ it is clear that the exchange interaction alone can be used to achieve an arbitrary 1-qubit gate up to an overall phase factor.

15.8.9 CNOT in Encoded Basis

To complete the proof that the exchange interaction is sufficient to implement any quantum computation up to an overall phase factor, we need to show how to implement the CNOT gate in the encoded basis. Luckily, this was done already by David DiVincenzo, Dave Bacon, Julia Kempe, Guido Burkard, and Birgitta Whaley and is shown in Fig. 15.19.

Hence, CNOT gates and arbitrary 1-qubit gates can be implemented using only the exchange interaction by working in an encoded basis wherein three physical qubits serve to represent the state of one logical qubit.

Universal quantum computation can be achieved in the encoded basis if we can implement any 1-qubit gate and the CNOT gate in the encoded basis. Our aforementioned results show how to implement any 1-qubit gate in the encoded basis. Thus by simply augmenting our constructive techniques for arbitrary 1-qubit gates with the encoded construction of the CNOT gate given in [240] we can therefore conclude that it is possible to perform universal quantum computation in the encoded basis. We also point out that our mapping from arbitrary 1-qubit gates to encoded gates is far more efficient than an earlier mapping by Hsieh et al. [240]. In that work

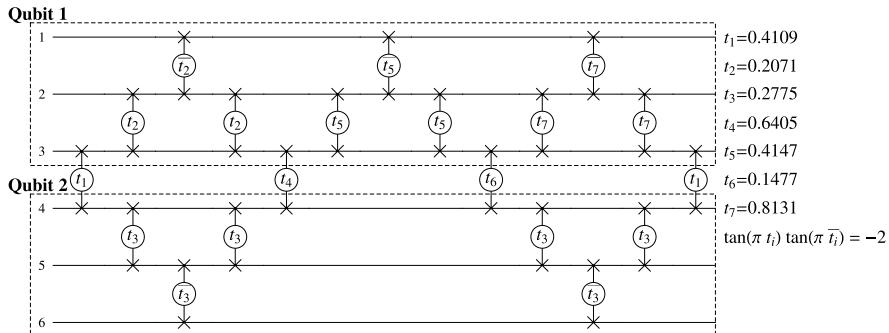


Fig. 15.19 CNOT in encoded basis as derived by David DiVincenzo, Dave Bacon, Julia Kempe, Guido Burkard, and Birgitta Whaley [150]. A sequence of exchange interactions can accomplish a gate equivalent to a CNOT gate, up to 1-qubit gates, in the encoded basis. This, coupled with exchange-only implementations of the 1-qubit gates, shows the exchange interaction is universal for quantum computation

the authors showed how to map from a universal, but *fixed*, set of gates, namely H , $\frac{\pi}{8}$, CNOT to an encoded basis. However, this fixed set of gates rarely gives efficient decompositions of 1-qubit gates. In contrast, we give explicit constructions for finding the exchange couplings in the physical basis needed to achieve any *arbitrary* 1-qubit gate. In addition, we give special cases of our result for the three standard types of 1-qubit gates R_x , R_y , or R_z . Thus, we have provided a fully constructive scheme for mapping any quantum computation into an encoded basis.

15.9 Equivalences Between Alternative Models of Quantum Computation

The alternative models of quantum computation described in this chapter, and elsewhere in the book, are noticeably different from one another. The Quantum Turing Machine Model, the Quantum Circuit Model, the Quantum Cellular Automata Model, the Teleportation-based Model, the One-Way Model, the Encoded Universality Model, the Topological Model, and the Adiabatic Model could not *appear* more different! It is therefore all the more surprising, therefore, that these disparate models of quantum computation are provably equivalent to one another! That is, we now know that the cost of mapping a quantum computation in any one of these models into a corresponding computation in any of the alternative models is at most polynomial in the size of the computation. The proofs appear in a handful of specialist papers scattered across the field.

The chain of equivalences was first begun by Andrew Yao in 1993 when he proved the Quantum Circuit Model can simulate the Quantum Turing Machine Model efficiently [550]. Dorit Aharonov, Wim van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, Oded Regev, Alexei Kitaev and Stewart Siu proved that the Quantum Adiabatic Model can simulate the Quantum Circuit Model efficiently [8, 273],

461]. Raussendorf, Browne, and Briegel proved the One-Way model is equivalent to standard Quantum Circuit Model [411]. Alexei Kitaev proved that non-Abelian anyons can simulate a quantum circuit efficiently [285]. That is, universal quantum computation is possible by braiding non-Abelian anyons. Freedman, Kitaev, and Wang later showed a system of anyons can be simulated efficiently by a quantum circuit [189]. Thus, topological quantum computing is provably no more and no less powerful than the quantum circuit model.

Such proofs, and similar ones for the other models, means that it really does not matter which model one picks as the basis for developing quantum computing hardware, as they are all roughly as good as one another. In practice, of course, different physical embodiments of quantum computational hardware may lend themselves more easily to one model or another. But the equivalence between the models allows the experimentalist to pick the model that best suits their scheme, without it affecting, in principle, the overall performance of their hardware in a significant way.

This equivalency between computational models is very reminiscent of what happened in classical computing when the Turing, Post, Church/Gödel models were all proven to be equivalent. There the surprising equivalency led to speculation that the essence of the computational process had been captured correctly, and provided the basis for the Strong Church-Turing thesis. That turned out to be incorrect. Should we feel more secure in the belief that the main features of quantum computing have been captured given the equivalence between our models of quantum computing?

15.10 Summary

This chapter surveyed many superficially different models of quantum computation, which turned out to be polynomially equivalent to one another. This equivalency frees experimentalists to choose whichever model of quantum computation best fits the quantum physical phenomena they have at their disposal and over which they can exert control. We do not yet know which model of quantum computing will lead to the first genuinely scalable universal quantum computer.

I hope you have enjoyed the book and found it useful, and that you are inspired to go forth and make your contributions to this exciting field. There is plenty left to discover!

15.11 Exercises

15.1 The circuit used to teleport a CNOT gate assumed that the input to the CNOT gate was of the form $|a\rangle|b\rangle$, i.e., an unentangled product of two single qubit computational basis states. Clearly this is not the most general input a CNOT gate can accept. Prove that the circuit used to teleport a CNOT gate also works correctly when the input to the CNOT gate is a direct product of two arbitrary 1-qubit

states, i.e., $|\psi_{ab}\rangle = |\varphi_a\rangle|\varphi_b\rangle$ where $|\varphi_a\rangle = \alpha|0\rangle + |\beta\rangle$ such that $|\alpha|^2 + |\beta|^2 = 1$ and $|\varphi_b\rangle = \gamma|0\rangle + |\delta\rangle$ such that $|\gamma|^2 + |\delta|^2 = 1$.

15.2 The circuit used to teleport a CNOT gate assumed that the input to the CNOT gate was of the form $|a\rangle|b\rangle$, i.e., an unentangled product of two single qubit computational basis states. Clearly this is not the most general input a CNOT gate can accept. Prove that the circuit used to teleport a CNOT gate also works correctly when the input to the CNOT gate is an arbitrary *entangled* state $|\psi_{ab}\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$ such that $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$. In this case, you need to think carefully about how to specify the 6-qubit input state to the double teleportation circuit when the input to the CNOT gate is on the first and sixth qubits. (Hint: Imagine starting with a 6-qubit state of the form $|\psi\rangle|\beta_{00}\rangle|\beta_{00}\rangle$ and think how to permute the qubits so that the qubit ordering $|j_1 j_2 j_3 j_4 j_5 j_6\rangle$ is mapped into the qubit ordering $|j_1 j_3 j_4 j_5 j_6 j_2\rangle$.)

15.3 The circuit used to teleport a CNOT gate assumed that the input to the CNOT gate was of the form $|a\rangle|b\rangle$, i.e., an unentangled product of two single qubit computational basis states. Clearly this is not the most general input a CNOT gate can accept. Prove that the circuit used to teleport a CNOT gate also works correctly when the input to the CNOT gate is an arbitrary 2-qubit *mixed* state?

15.4 A critical step in designing a quantum circuit capable of teleporting a CNOT gate is to realize that both CNOT gates can be eliminated from the quantum circuit shown in the left hand side of Fig. 15.11 merely by modifying control logic on the remaining gates. In the text, we claimed that if the classical control values in the circuit containing two CNOT gates were M_1, M_2, M_3, M_4 that the corresponding control values in the circuit devoid of CNOT gates would be m_1, m_2, m_3, m_4 as summarized in Table 15.1. Verify the corresponding sets of control values claimed in Table 15.1 are indeed correct.

15.5 Given the exchange interaction $U^{1,2}(t) = \exp[-i t (\sigma_x \otimes \sigma_x \otimes \mathbb{1}_2 + \sigma_y \otimes \sigma_y \otimes \mathbb{1}_2 + \sigma_z \otimes \sigma_z \otimes \mathbb{1}_2)]$ prove:

$$U_{\mathcal{L}}^{1,2}(t) = \begin{pmatrix} e^{3it} & 0 \\ 0 & e^{-it} \end{pmatrix}$$

15.6 Given the exchange interaction $U^{1,3}(t) = \exp[-i t (\sigma_x \otimes \mathbb{1}_2 \otimes \sigma_x + \sigma_y \otimes \mathbb{1}_2 \otimes \sigma_y + \sigma_z \otimes \mathbb{1}_2 \otimes \sigma_z)]$ prove:

$$U_{\mathcal{L}}^{1,3}(t) = \frac{1}{4} e^{-it} \begin{pmatrix} 3 + e^{4it} & \sqrt{3}(-1 + e^{4it}) \\ \sqrt{3}(-1 + e^{4it}) & 1 + 3e^{4it} \end{pmatrix}$$

15.7 Given the exchange interaction $U^{2,3}(t) = \exp[-it(\mathbb{1}_2 \otimes \sigma_x \otimes \sigma_x + \mathbb{1}_2 \otimes \sigma_y \otimes \sigma_y + \mathbb{1}_2 \otimes \sigma_z \otimes \sigma_z)]$ prove:

$$U_{\mathcal{L}}^{2,3}(t) = \sigma_z U_{\mathcal{L}}^{1,3}(t) \sigma_z$$

15.8 In the text we claimed a particular sequence of three exchange interactions were sufficient to generate a Walsh-Hadamard gate in the logical basis. Verify this claim as follows. First prove that:

$$U_{\mathcal{L}}^{1,2}(a) \cdot U_{\mathcal{L}}^{2,3}(b) \cdot U_{\mathcal{L}}^{1,2}(c) \\ = \begin{pmatrix} \frac{1}{4}e^{i(3a-b+3c)}(3 + e^{4ib}) & -\frac{1}{4}\sqrt{3}e^{i(3a-b-c)}(-1 + e^{4ib}) \\ -\frac{1}{4}\sqrt{3}e^{-i(a+b-3c)}(-1 + e^{4ib}) & \frac{1}{4}e^{-i(a+b+c)}(1 + 3e^{4ib}) \end{pmatrix}$$

then substitute in the following values for a , b , and c into $U_{\mathcal{L}}^{1,2}(a) \cdot U_{\mathcal{L}}^{2,3}(b) \cdot U_{\mathcal{L}}^{1,2}(c)$ and simplify:

$$a = \frac{\pi}{2} + \frac{1}{8} \left(3\pi + 2\cos^{-1} \left(\sqrt{\frac{2}{3}} \right) \right) \\ b = \frac{1}{4} \cos^{-1} \left(-\frac{1}{3} \right) \\ c = \frac{1}{8} \left(3\pi + 2\cos^{-1} \left(\sqrt{\frac{2}{3}} \right) \right)$$

Hence, verify:

$$H = U_{\mathcal{L}}^{1,2}(a) \cdot U_{\mathcal{L}}^{2,3}(b) \cdot U_{\mathcal{L}}^{1,2}(c) \quad (15.45)$$

References

1. D. S. Abrams and S. Lloyd, "Simulation of Many-Body Fermi Systems on a Universal Quantum Computer," *Phys. Rev. Lett.*, Volume **79**, Issue 13 (1997) pp. 2586–2589.
2. D. Abrams and S. Lloyd, "Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors," *Phys. Rev. Lett.*, Volume **83**, Issue 24 (1999) pp. 5162–5165.
3. D. S. Abrams and C. P. Williams, "Fast Quantum Algorithms for Numerical Integrals and Stochastic Processes," [arXiv:quant-ph/9908083](https://arxiv.org/abs/quant-ph/9908083) (1999).
4. D. Achlioptas, A. Naor, and Y. Peres, "Rigorous Location of Phase Transitions in Hard Optimization Problems," *Nature*, Volume **435** (2005) pp. 759–764.
5. L. Adleman and M. Huang, "Recognizing Primes in Random Polynomial Time," in *Proceedings of ACM STOC'87* (1987) pp. 462–470.
6. M. Agrawal, N. Kayal, and N. Saxena, "PRIMES Is in P," *Annals of Mathematics*, Volume **160** (2004) pp. 781–793.
7. D. Aharonov and M. Ben-Or, "Fault-Tolerant Quantum Computation with Constant Error," in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, (1997) pp. 176–188.
8. D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, "Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation," in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04)*, IEEE Computer Society, Washington (2004) pp. 42–51.
9. N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transform," *IEEE Trans. Computer*, Volume **C-23**, Issue 1 (1974) pp. 90–93.
10. G. M. Akselrod, J. B. Altepeter, E. R. Jeffrey, and P. G. Kwiat, "Phase-compensated Ultra-bright Source of Entangled Photons: Erratum," *Opt. Express*, Volume **15** (2007) pp. 5260–5261.
11. P. Aliferis, "Level Reduction and the Quantum Threshold Theorem," Ph.D. Thesis, California Institute of Technology, Pasadena, CA, [arXiv:quant-ph/0703230v1](https://arxiv.org/abs/quant-ph/0703230v1) (2007).
12. L. Allen, M. W. Beijersbergen, R. J. C. Spreeuw, and J. P. Woerdman, "Orbital Angular Momentum of Light and the Transformation of Laguerre-Gaussian Laser Modes," *Phys. Rev. A*, Volume **45**, Issue 11 (1992) pp. 8185–8189.
13. J. B. Altepeter, E. Jeffrey, and P. G. Kwiat, "Phase-compensated Ultra-bright Source of Entangled Photons," *Opt. Express*, Volume **13** (2005) p. 8951.
14. "A Mathematical Theory of Communication," *The Bell System Technical Journal*, Volume **27**, pp. 379–423, 623–656 (1948).
15. P. W. Anderson, "When the Electron Falls Apart," *Physics Today*, Volume **50**, Issue 10 (1997) pp. 42–47.

16. "Announcing the Advanced Encryption Standard (AES)," United States Federal Information Processing Standards Publication 197 (FIPS 197), 26th November (2001). Available at <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.
17. M. Arndt, M. Aspelmeyer, H. J. Bernstein, R. Bertlmann, C. Brukner, J. P. Dowling, J. Eisert, A. Ekert, C. A. Fuchs, D. M. Greenberger, M. A. Horne, T. Jennewein, P. G. Kwiat, N. D. Mermin, J.-W. Pan, E. M. Rasel, H. Rauch, T. G. Rudolph, C. Salomon, A. V. Sergienko, J. Schmiedmayer, C. Simon, V. Vedral, P. Walther, G. Weihs, P. Zoller, and M. Zukowski, "Quantum Physics from A to Z," [arXiv:quant-ph/0505187v4](https://arxiv.org/abs/quant-ph/0505187v4) (2005) p. 2.
18. M. Arndt, L. Hackermüller, K. Hornberger, and A. Zeilinger, "Coherence and Decoherence Experiments with Fullerenes," in *Decoherence, Entanglement and Information Protection in Complex Quantum Systems*, Springer, Berlin (2005).
19. M. C. Arnesen, S. Bose, and V. Vedral, "Natural Thermal and Magnetic Entanglement in 1D Heisenberg Model," *Phys. Rev. Lett.*, Volume **87** (2001) 017901.
20. S. Arora, "Nearly Linear Time Approximation Schemes for Euclidean TSP and other Geometric Problems," in *Proceedings of the International Workshop on Randomization and Approximation Techniques in Computer Science*, Lecture Notes in Computer Science, Volume **1269**, Springer, Berlin (1997) pp. 55–56.
21. A. Aspect, "Bell's Inequality Test: More Ideal than Ever," *Nature*, Volume **398** (1999) pp. 189–190.
22. A. Aspect, P. Grangier, and G. Roger, "Experimental Tests of Realistic Local Theories via Bell's Theorem," *Phys. Rev. Lett.*, Volume **47**, Issue 7 (1981) pp. 460–463.
23. A. Aspect, J. Dalibard, and G. Roger, "Experimental Test of Bell's Inequalities Using Time-Varying Analyzers," *Phys. Rev. Lett.*, Volume **49**, Issue 25 (1982) pp. 1804–1807.
24. A. Aspect, P. Grangier, and G. Roger, "Experimental Realization of Einstein-Podolsky-Rosen-Bohm Gedankenexperiment: A New Violation of Bell's Inequalities," *Phys. Rev. Lett.*, Volume **49**, Issue 2 (1982) pp. 91–94.
25. A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, "Simulated Quantum Computation of Molecular Energies," *Science*, Volume **309** (2005) pp. 1704–1707.
26. D. Awschalom, D. Loss, and N. Samarth (eds.), *Semiconductor Spintronics and Quantum Computation*, Springer, Berlin (2002) ISBN 3-540-42176-9.
27. D. Bacon, *The Race to Build a Quantum Computer*, Physics World (2009) pp. 26–31.
28. D. Bacon, D. A. Lidar, and K. B. Whaley, "Robustness of Decoherence-Free Subspaces for Quantum Computation," *Phys. Rev. A*, Volume **60** (1999) p. 1944.
29. D. Bacon, J. Kempe, D. A. Lidar, and K. B. Whaley, "Universal Fault-Tolerant Quantum Computation on Decoherence-Free Subspaces," *Phys. Rev. Lett.*, Volume **85** (2000) p. 1758.
30. D. Bacon, A. Childs, and W. van Dam, "From Optimal Measurement to Efficient Quantum Algorithms for the Hidden Subgroup Problem over Semidirect Product Groups," in *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science* (2005) pp. 469–478.
31. S. Balensiefer, L. Kregor-Stickles, and M. Oskin, "An Evaluation Framework and Instruction Set Architecture for Ion-Trap Based Quantum Micro-Architectures," *ACM SIGARCH Computer Architecture News*, Volume **33**, Issue 2 (2005) pp. 186–196.
32. A. Barenco, "A Universal Two-Bit Gate for Quantum Computation," *Proc. R. Soc. Lond. A*, Volume **449**, Issue 1937 (1995) pp. 679–683.
33. A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, "Elementary Gates for Quantum Computation," *Phys. Rev. A*, Volume **52** (1995) pp. 3457–3467.
34. A. Barenco, A. Ekert, K.-A. Suominen, and P. Törmä, "Approximate Quantum Fourier Transform and Decoherence," *Phys. Rev. A*, Volume **54**, Issue 1 (1996) pp. 139–146.
35. A. Barenco, A. Berthiaume, D. Deutsch, A. Ekert, R. Jozsa, and C. Macchiavello, "Stabilisation of Quantum Computations by Symmetrisation," *SIAM Journal on Computing* (1997) pp. 1541–1557.
36. H. Barnum, H. J. Bernstein, and L. Spector, "Quantum Circuits for OR and AND of ORs," *Journal of Physics A: Mathematical and General*, Volume **33**, Issue 45 (2000) pp. 8047–8057.

37. J. T. Barreiro, T. C. Wei, and P. G. Kwiat, "Beating the Channel Capacity Limit for Linear Photonic Superdense Coding," *Nature Physics*, Volume **4** (2008) pp. 282–286.
38. M. D. Barrett, J. Chiaverini, T. Schaetz, J. Britton, W. M. Itano, J. D. Jost, E. Knill, C. Langer, D. Leibfried, R. Ozeri, and D. J. Wineland, "Deterministic Quantum Teleportation of Atomic Qubits," *Nature*, Volume **429** (2004) pp. 737–739.
39. A. Beige, "Quantum Computing Using Dissipation," *Inst. Phys. Conf. Ser.*, Volume **173** (2003) p. 35.
40. A. Beige, "Dissipation-Assisted Quantum Gates with Cold Trapped Ions," *Phys. Rev. A*, Volume **67** (2003) 020301(R).
41. A. Beige, H. Cable, and P. L. Knight, "Dissipation-Assisted Quantum Computation in Atom-Cavity Systems," in *Proceedings of SPIE*, Volume **5111** (2003) p. 370.
42. A. Ben-Kish, B. DeMarco, V. Meyer, M. Rowe, J. Britton, W. M. Itano, B. M. Jelenković, C. Langer, D. Leibfried, T. Rosenband, and D. J. Wineland, "Experimental Demonstration of a Technique to Generate Arbitrary Quantum Superposition States of a Harmonically Bound Spin-1/2 Particle," *Phys. Rev. Lett.*, Volume **90** (2003) 037902.
43. P. Benioff, "The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines," *Journal of Statistical Physics*, Volume **22** (1980) pp. 563–591.
44. C. Bennett, "Logical Reversibility of Computation," *IBM Journal of Research and Development*, Volume **17** (1973) pp. 525–532.
45. C. Bennett, "Time-Space Tradeoff for Reversible Computation," *SIAM Journal on Computing*, Volume **18** (1989) pp. 766–776.
46. C. H. Bennett, "Quantum Cryptography Using Any Two Nonorthogonal States," *Phys. Rev. Lett.*, Volume **68** (1992) pp. 3121–3124.
47. C. H. Bennett and G. Brassard, "Quantum Cryptography: Public Key Distribution and Coin Tossing," in *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing*, Bangalore, December (1984) pp. 175–179. A scanned PDF of this paper is available at <http://www.research.ibm.com/people/b/bennetc/bennetc198469790513.pdf>.
48. C. H. Bennett and G. Brassard, "The Dawn of a New Era for Quantum Cryptography: The Experimental Prototype is Working!" *SIGACT News*, Volume **20** (1989) pp. 78–82.
49. C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an Unknown Quantum State via Dual Classical and Einstein-Podolsky-Rosen Channels," *Phys. Rev. Lett.*, Volume **70** (1993) pp. 1895–1899.
50. C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer, "Generalized Privacy Amplification," *IEEE Transactions on Information Theory*, Volume **41**, Issue 6 (1995) pp. 1915–1923.
51. C. H. Bennett, H. J. Bernstein, S. Popescu, and B. Schumacher, "Concentrating Partial Entanglement by Local Operations," *Phys. Rev. A*, Volume **53**, Issue 4 (1996) pp. 2046–2052.
52. C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. Smolin, and W. K. Wootters, "Purification of Noisy Entanglement and Faithful Teleportation via Noisy Channels," *Phys. Rev. Lett.*, Volume **76**, Issue 5 (1996) pp. 722–725.
53. C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, "Purification of Noisy Entanglement and Faithful Teleportation via Noisy Channels," *Phys. Rev. Lett.*, Volume **76**, Issue 5 (1996) pp. 722–725.
54. C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, "Mixed-state Entanglement and Quantum Error Correction," *Phys. Rev. A*, Volume **54**, Issue 5 (1996) pp. 3824–3851.
55. C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, "Strengths and Weaknesses of Quantum Computing," *SIAM Journal on Computing*, Volume **26**, Issue 5 (1997) pp. 1510–1523.
56. L. Berman and J. Hartmanis, "On Isomorphisms and Density of NP and Other Complete Sets," *SIAM Journal on Computing*, Volume **6**, Issue 2 (1977) pp. 305–322.
57. E. Bernstein and U. Vazirani, "Quantum Complexity Theory," in *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing* (1993) pp. 11–20.
58. E. Bernstein and U. Vazirani, "Quantum Complexity Theory," *SIAM Journal on Computing*, Volume **26** (1997) pp. 11–20.

59. A. Berzina, A. Dubrovsky, R. Freivalds, L. Lace, and O. Scegulnaja, *Quantum Query Complexity for Some Graph Problems*, Lecture Notes in Computer Science, Volume **2932**, Springer, Berlin (2004) pp. 1–11.
60. H. A. Bethe, “Zur Theorie der Metalle: I. Eigenwerte und Eigenfunktionen der linearen Atom Kette,” *Zeitschrift für Physik*, Volume **71** (1931) pp. 205–226.
61. D. Biron, O. Biham, E. Biham, M. Grassl, and D. A. Lidar, *Generalized Grover Search Algorithm for Arbitrary Initial Amplitude Distribution*, Lecture Notes in Computer Science, Volume **1509**, Springer, Berlin (1999) pp. 140–147.
62. I. Bloch, “Quantum Coherence and Entanglement with Ultracold Atoms in Optical Lattices,” *Nature*, Volume **453** (2008) pp. 1016–1022.
63. P. Bonderson, A. Kitaev, and K. Shtengel, “Detecting Non-Abelian Statistics in the $v = 5/2$ Fractional Quantum Hall State,” *Phys. Rev. Lett.*, Volume **96** (2006) 016803.
64. P. Bonderson, K. Shtengel, and J. K. Slingerland, “Probing Non-Abelian Statistics with Quasiparticle Interferometry,” *Phys. Rev. Lett.*, Volume **97** (2006) 016401.
65. N. E. Bonesteel, L. Hormozi, G. Zikos, and S. H. Simon, “Braid Topologies for Quantum Computation,” *Phys. Rev. Lett.*, Volume **95** (2005) 140503.
66. D. Boschi, S. Branca, F. De Martini, L. Hardy, and S. Popescu, “Experimental Realization of Teleporting an Unknown Pure Quantum State via Dual Classical and Einstein-Podolsky-Rosen Channels,” *Phys. Rev. Lett.*, Volume **80**, Issue 6 (1998) pp. 1121–1125.
67. D. Boschi, S. Branca, F. De Martini, L. Hardy, and S. Popescu, “Experimental Realization of Teleporting an Unknown Pure Quantum State via Dual Classical and Einstein-Podolsky-Rosen Channels,” *Phys. Rev. Lett.*, Volume **80**, Issue 6 (1998) pp. 1121–1125.
68. D. Bouwmeester, J.-W. Pan, K. Mattle, M. Eibl, H. Weinfurter, and A. Zeilinger, “Experimental Quantum Teleportation,” *Nature*, Volume **390** (1997) pp. 575–579.
69. D. Bouwmeester, J. W. Pan, K. Mattle, M. Eibl, H. Weinfurter, and A. Zeilinger, “Experimental Quantum Teleportation,” *Nature*, Volume **390** (1997) pp. 575–579.
70. M. Boyer, G. Brassard, P. Høyer, and A. Tapp, “Tight Bounds on Quantum Searching,” *Fortsch. Phys.*, Volume **46** (1998) pp. 493–506; Also in *Proceedings of the Fourth Workshop on Physics and Computation (PhysComp'96)*, eds. T. Toffoli, M. Biafore, and J. Leao, New England Complex Systems Institute, Boston (1996) pp. 36–43.
71. G. Brassard, “Searching a Quantum Phone Book,” *Science*, Volume **275** (1997) pp. 627–628.
72. G. Brassard and P. Høyer, “An Exact Polynomial-Time Algorithm for Simon’s Problem,” in *Proceedings of Fifth Israeli Symposium on Theory of Computing and Systems, ISTCS*, IEEE Computer Society, Los Alamitos (1997) pp. 12–33.
73. G. Brassard and L. Salvail, “Secret Key Reconciliation by Public Discussion,” *Advances in Cryptology, Proceedings of Eurocrypt ’93* (1994) pp. 410–423.
74. G. Brassard, P. Høyer, and Alain Tapp, “Quantum Algorithm for the Collision Problem,” *ACM SIGACT News*, Volume **28** (1997) pp. 14–19.
75. G. Brassard, S. L. Braunstein, and R. Cleve, “Teleportation as a Quantum Computation,” *Physica D*, Volume **120**, Issues 1–2 (1998) pp. 43–47.
76. G. Brassard, P. Høyer, and A. Tapp, “Quantum Counting,” [arXiv:quant-ph/9805082v1](https://arxiv.org/abs/quant-ph/9805082v1) (1998).
77. G. Brassard, R. Cleve, and A. Tapp, “Cost of Exactly Simulating Quantum Entanglement with Classical Communication,” *Phys. Rev. Lett.*, Volume **83** (1999) pp. 1874–1877.
78. G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, “Quantum Amplitude Amplification and Estimation,” [arXiv:quant-ph/0005055v1](https://arxiv.org/abs/quant-ph/0005055v1) (2000).
79. S. L. Braunstein and J. A. Smolin, “Perfect Quantum Error Correction Coding in 24 Laser Pulses,” *Phys. Rev. A*, Volume **55** (1997) pp. 945–950.
80. M. J. Bremner, C. M. Dawson, J. L. Dodd, A. Gilchrist, A. W. Harrow, D. Mortimer, M. A. Nielsen, and T. J. Osborne, “Practical Scheme for Quantum Computation with Any Two-Qubit Entangling Gate,” *Phys. Rev. Lett.*, Volume **89** (2002) 247902.
81. G. K. Brennen and J. K. Pachos, “Why Should Anyone Care about Computing with Anyons?”, *Proc. R. Soc. A*, Volume **464** (2008) pp. 1–24.
82. G. K. Brennen, C. M. Caves, P. S. Jessen, and I. H. Deutsch, “Quantum Logic Gates in Optical Lattices,” *Phys. Rev. Lett.*, Volume **82** (1999) pp. 1060–1063.

83. H. J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, “Quantum Repeaters: The Role of Imperfect Local Operations in Quantum Communication,” Phys. Rev. Lett., Volume **81** (1998) pp. 5932–5935.
84. H. J. Briegel, T. Calarco, D. Jaksch, J. I. Cirac, and P. Zoller, “Quantum Computing with Neutral Atoms,” Journal of Modern Optics, Volume **47** (2000) pp. 415–451.
85. C. Brukner, V. Vedral, and A. Zeilinger, “A Crucial Role of Entanglement in Bulk Properties of Solids,” Phys. Rev. A, Volume **73** (2006) 012110.
86. D. Bruß, “Optimal Eavesdropping in Quantum Cryptography with Six States,” Phys. Rev. Lett., Volume **81**, Issue 14 (1998) pp. 3018–3021.
87. H. Buhrman and R. Špalek, “Quantum Verification of Matrix Products,” in *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms* (2006) pp. 880–889.
88. H. Buhrman, J. Tromp, and P. Vitányi, “Time and Space Bounds for Reversible Simulation,” Journal of Physics A: Mathematical & General, Volume **34**, Issue 35 (2001) pp. 6821–6830.
89. D. Bulger, “Quantum Basin Hopping with Gradient-based Local Optimisation,” [arXiv: quant-ph/0507193](https://arxiv.org/abs/quant-ph/0507193) (2005).
90. S. Bullock and I. Markov, “Arbitrary Two-qubit Computation in 23 Elementary Gates,” Phys. Rev. A, Volume **68** (2003) 012318.
91. A. Bundy, *The Computer Modelling of Mathematical Reasoning*, Academic Press, London (1983) ISBN 0-12-141352-0.
92. G. Burkard, D. Loss, and D. P. Di Vincenzo, “Coupled Quantum Dots as Quantum Gates,” Phys. Rev. B, Volume **59** (1999) pp. 2070–2078.
93. V. Bužek and M. Hillery, “Quantum Copying: Beyond the No-cloning Theorem,” Phys. Rev. A, Volume **54**, Issue 3 (1996) pp. 1844–1852.
94. Z. Cai, K. Sendl, and J. Reimers, “Failure of Density-functional Theory and Time-dependent Density-functional Theory for Large Extended Pi Systems,” Journal of Chemical Physics, Volume **117**, Issue 12 (2002) pp. 5543–5549.
95. A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, “Quantum Error Correction via Codes over GF(4),” IEEE Trans. Inform. Theory, Volume **44**, Issue 4 (1998) pp. 1369–1387.
96. F. E. Camino, W. Zhou, and V. J. Goldman, “Realization of a Laughlin Quasiparticle Interferometer: Observation of Fractional Statistics,” Phys. Rev. B, Volume **72** (2005) 075342.
97. A. Carlini and A. Hosoya, “Quantum Probabilistic Subroutines and Problems in Number Theory,” Phys. Rev. A, Volume **62** (2000) 032312.
98. S. Cass, “Listening In,” IEEE Spectrum, April (2003) pp. 33–37.
99. N. Cerf, “Optical Quantum Cloning—a Review,” ed. E. Wolf, *Progress in Optics*, Volume **49**, Elsevier, Amsterdam (2006) p. 455.
100. N. J. Cerf, L. K. Grover, and C. P. Williams, “Nested Quantum Search and Structured Problems,” Phys. Rev. A, Volume **61** (2000) 032303.
101. P. Cheeseman, B. Kanefsky, and W. M. Taylor, “Where the Really Hard Problems Are,” in *Proc. of International Joint Conference on Artificial Intelligence (IJCAI’91)*, Sydney, Morgan Kauffman, San Mateo (1991) pp. 331–337.
102. D. Cheung, D. Maslov, J. Mathew, and D. K. Pradhan, *On the Design and Optimization of a Quantum Polynomial-Time Attack on Elliptic Curve Cryptography*, Lecture Notes in Computer Science, Volume **5106**, Springer, Berlin (2008).
103. L. Childress, J. M. Taylor, A. S. Sørensen, and M. D. Lukin, “Fault-tolerant Quantum Repeaters with Minimal Physical Resources and Implementations based on Single-photon Emitters,” Phys. Rev. A, Volume **72** (2005) 052330.
104. A. Childs and W. van Dam, “Quantum Algorithms for Algebraic Problems,” Reviews of Modern Physics, Volume **82**, Issue 1 (2010) pp. 1–52.
105. A. W. Chin, A. Datta, F. Caruso, S. F. Huelga, and M. B. Plenio, “Noise-Assisted Energy Transfer in Quantum Networks and Light-Harvesting Complexes,” [arXiv:0910.4153v1](https://arxiv.org/abs/0910.4153v1) (2009).
106. C. W. Chou, J. Laurat, H. Deng, K. S. Choi, H. de Riedmatten, D. Felinto, and H. J. Kimble, “Functional,” Quantum Nodes for Entanglement Distribution over Scalable Quantum Networks,” Volume **316**, Issue 5829 (2007) pp. 1316–1320.

107. I. L. Chuang and Y. Yamamoto, "A Simple Quantum Computer," *Phys. Rev. A*, Volume **52** (1995) pp. 3489–3496.
108. J. I. Cirac, P. Zoller, H. J. Kimble, and H. Mabuchi, "Quantum State Transfer and Entanglement Distribution Among Distant Nodes in a Quantum Network," *Phys. Rev. Lett.*, Volume **78** (1997) p. 3221.
109. M. Clarke, A.Chefles, S. M. Barnett, and E. Riis, "Experimental Demonstration of Optimal Unambiguous State Discrimination," *Phys. Rev. A*, Volume **63** (2000) 040305.
110. R. Cleve, "Methodologies for Designing Block Ciphers and Cryptographic Protocols," Ph. D. Thesis, University of Toronto (1990).
111. R. Cleve, "Complexity Theoretic Issues Concerning Block Ciphers Related to DES," in *Proceedings of Advances in Cryptology (CRYPTO'90)*, Springer, Berlin (1991) pp. 530–544.
112. R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, "Quantum Algorithms Revisited," *Proc. R. Soc. Lond. A*, Volume **454**, Issue 1969 (1998) pp. 339–354.
113. G. P. Collins, "Fractionally Charged Quasiparticles Signal Their Presence with Noise," *Physics Today*, Volume **50**, Issue 11 (1997) pp. 17–19.
114. G. P. Collins, *Computing with Quantum Knots*, Scientific American, New York (2006) pp. 57–63.
115. S. A. Cook, "The Complexity of Theorem-Proving Procedures," in *Proceedings of ACM STOC'71* (1971) pp. 151–158.
116. N. R. Cooper, N. K. Wilkin, and J. M. F. Gunn, "Quantum Phases of Vortices in Rotating Bose-Einstein Condensates," *Phys. Rev. Lett.*, Volume **87** (2001) 120405.
117. T. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley, New York (1991) ISBN 0-471-06259-6.
118. R. Crandall, *Topics in Advanced Scientific Computation*, TELOS, Springer, New York (1996) pp. 125–126.
119. P. Crescenzi and C. H. Papadimitriou, "Reversible Simulation of Space-bounded Computations," *Theoretical Computer Science*, Volume **143**, Issue 1 (1995) pp. 159–165.
120. J. A. Csirik, "Cost of Exactly Simulating a Bell Pair Using Classical Communication," *Phys. Rev. A*, Volume **66**, Issue 1 (2002) 014302.
121. D. Curtis and D. A. Meyer, "Towards Quantum Template Matching," in *Quantum Communications and Quantum Imaging*, eds. R. Meyers and Y. Shih, Proceedings of SPIE, Volume **5161** (2004) p. 134.
122. G. Cybenko, *Reducing Quantum Computations to Elementary Unitary Operations*, Computing in Science and Engineering, IEEE Computer Society, Los Alamitos (2001) pp. 27–32.
123. S. Das Sarma, M. Freedman, and C. Nayak, "Topologically Protected Qubits from a Possible Non-Abelian Fractional Quantum Hall State," *Phys. Rev. Lett.*, Volume **94** (2005) 166802.
124. S. Das Sarma, M. Freedman, and C. Nayak, "Topologically Protected Qubits from a Possible Non-Abelian Fractional Quantum Hall State," *Phys. Rev. Lett.*, Volume **94** (2005) 166802.
125. S. Das Sarma, M. Freedman, and C. Nayak, "Topological Quantum Computation," *Physics Today*, Volume **7** (2006) pp. 32–38.
126. I. Daubechies, "Discrete Sets of Coherent States and Their Use in Signal Analysis," in *Differential Equations and Mathematical Physics*, Volume **1285**, eds. I. W. Knowles and Y. Saito, Springer, Berlin (1987) pp. 73–82.
127. I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets," *Comm. Pure & Appl. Math.*, Volume **41**, Issue 7 (1988) pp. 909–996.
128. I. Daubechies, "Time-frequency Localization Operators: a Geometric Phase Space Approach," *IEEE Trans. Inf. Theory*, Volume **34**, Issue 4 (1988) pp. 605–612.
129. I. Daubechies, "Orthonormal Bases of Wavelets with Finite Support—Connection with Discrete Filters," in *Proceedings of the 1987 International Workshop on Wavelets and Applications*, Marseille, France, eds. J. M. Combes, A. Grossmann, and Ph. Tchamitchian, Springer, Berlin (1989).
130. I. Daubechies, "The Wavelet Transform, Time-frequency Localization and Signal Analysis," *IEEE Trans. Inf. Theory*, Volume **36**, Issue 5 (1990) pp. 961–1005.

131. I. Daubechies and T. Paul, "Wavelets—Some Applications," in *Proceedings of the International Conference on Mathematical Physics*, eds. M. Mebkhout and R. Seneor, World Scientific, Marseille (1987) pp. 675–686.
132. R. M. Davis, "The Data Encryption Standard in Perspective," IEEE Comm. Soc. Magazine, Volume **16** (1978) pp. 5–9.
133. M. Davis, G. Logemann, and D. Loveland, "A Machine Program for Theorem Proving," Comm. ACM, Volume **5** (1962) p. 394.
134. B. Daviss, "Splitting the Electron," New Scientist (1998) p. 36.
135. J. N. de Beaudrap, R. Cleve, and J. Watrous, "Sharp Quantum versus Classical Query Complexity Separations," Algorithmica, Volume **34**, Issue 4 (2002) pp. 449–461.
136. D. Deutsch, "Quantum Theory, the Church-Turing Principle, and the Universal Quantum Computer," Proc. R. Soc. Lond. A, Volume **400** (1985) pp. 97–117.
137. D. Deutsch, "Quantum Computational Networks," Proc. R. Soc. Lond. A, Volume **425**, Issue 1868 (1989) pp. 73–90.
138. D. Deutsch and R. Jozsa, "Rapid Solution of Problems by Quantum Computation," Proc. R. Soc. Lond. A, Volume **439**, Issue 1907 (1992) pp. 553–558.
139. D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera, "Quantum Privacy Amplification and the Security of Quantum Cryptography over Noisy Channels," Phys. Rev. Lett., Volume **77**, Issue 2818 (1996) pp. 2818–2821.
140. I. H. Deutsch, G. K. Brennen, and P. S. Jessen, "Quantum Computing with Neutral Atoms in an Optical Lattice," Fortschritte der Physik, Volume **48** (2000) pp. 925–943.
141. S. J. Devitt, K. Nemoto, and W. J. Munro, "The Idiots Guide to Quantum Error Correction," [arXiv:0905.2794v2](https://arxiv.org/abs/0905.2794v2) (2009).
142. D. Dieks, "Communication by EPR Devices," Phys. Lett. A, Volume **92**, Issue 6 (1982) pp. 271–271. Available at http://igitur-archive.library.uu.nl/phys/2006-1214-212615/dieks_82_communication.pdf.
143. W. Diffie and M. E. Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, Volume **22** (1976) pp. 644–654.
144. W. Diffie and M. Hellman, "New Directions in Cryptography," IEEE Trans. Info. Theory, Volume **IT-22** (1976) pp. 644–654.
145. P. Dirac, in *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, Volume **123**, Issue 792, 6th April (1929).
146. D. P. DiVincenzo, "Quantum Computation," Science, Volume **270** (1995) pp. 255–261.
147. D. P. DiVincenzo, "Two-Bit Gates are Universal for Quantum Computation," Phys. Rev. A, Volume **51** (1995) pp. 1015–1022.
148. D. P. DiVincenzo, "The Physical Implementation of Quantum Computation," Fortschritte der Physik, Volume **48**, Issue 9–11 (2000) pp. 771–783.
149. D. P. DiVincenzo and J. Smolin, "Results on Two-Bit Gate Design for Quantum Computers," in *Proceedings of the Workshop on Physics and Computation*, IEEE Computer Society Press, Dallas (1994) pp. 14–23.
150. D. P. DiVincenzo, D. Bacon, J. Kempe, G. Burkard, and K. B. Whaley, "Universal Quantum Computation with the Exchange Interaction," Nature (London), Volume **408** (2000) pp. 339–342.
151. A. R. Dixon, Z. L. Yuan, J. F. Dynes, A. W. Sharpe, and A. J. Shields, "Gigahertz Decoy Quantum Key Distribution with 1 Mbit/s Secure Key Rate," Opt. Express, Volume **16**, Issue 23 (2008) pp. 18790–18979.
152. L. M. Duan and G. C. Guo, "Probabilistic Cloning and Identification of Linearly Independent Quantum States," Phys. Rev. Lett., Volume **80** (1998) pp. 4999–5002.
153. L. M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, "Long-distance Quantum Communication with Atomic Ensembles and Linear Optics," Nature, Volume **414** (2001) pp. 413–418.
154. S. V. Dudiy and A. Zunger, "Searching for Alloy Configurations with Extreme Physical Properties: Genetic Algorithm Inverse Band Structure of Ga(P, N)," Phys. Rev. Lett., Volume **97** (2006) 046401.

155. Duncan Campbell in STOA (Scientific and Technological Options Assessment), 1999, Part 2/5, with reference to Baltimore Sun, "America's Fortress of Spies," by Scott Shane and Tom Bowman, 3rd December 1995, and Washington Post, "Recent U.S. Coups in New Espionage," by William Drozdiak.
156. Duncan Campbell in STOA (Scientific and Technological Options Assessment), 1999, Part 2/5, with reference to New York Times, "How Washington Inc. Makes a Sale," by David Sanger, 19th February 1995.
157. C. Dürr and P. Hoyer, "A Quantum Algorithm for Finding the Minimum," [arXiv:quant-ph/9607014v2](https://arxiv.org/abs/quant-ph/9607014v2) (1996).
158. C. Dürr and M. Santha, "A Decision Procedure for Unitary Linear Quantum Cellular Automata," SIAM Journal on Computing, Volume **31** (2002) pp. 1076–1089.
159. C. Dürr, H. LêThanh, and M. Santha, "A Decision Procedure for Well-formed Linear Quantum Cellular Automata," Random Struct. Algorithms, Volume **11** (1997) pp. 381–394.
160. W. Dür, H. J. Briegel, J. I. Cirac, and P. Zoller, "Quantum Repeaters Based on Entanglement Purification," Phys. Rev. A, Volume **59**, Issue 1 (1999) pp. 169–181.
161. C. Dürr, M. Heiligman, P. Hoyer, and M. Mhalla, "Quantum Query Complexity of Some Graph Problems," SIAM Journal on Computing, Volume **35**, Issue 6 (2006) pp. 1310–1328.
162. M. Dyer, A. Frieze, and R. Kannan, "A Random Polynomial Time Algorithm for Approximating the Volume of Convex Bodies," Journal of the Association of Computing Machinery, Volume **38**, Issue 1 (1991) pp. 1–17.
163. P. Echternach, C. P. Williams, S. C. Dultz, S. Braunstein, and J. P. Dowling, "Universal Quantum Gates for Single Cooper Pair Box Based Quantum Computing," Quantum Information and Computation, Volume **1** (2001) pp. 143–150.
164. J. Eisert, "Entanglement in Quantum Information Theory," Ph.D. thesis, University of Potsdam, February (2001). Available as [arXiv:quant-ph/0610253](https://arxiv.org/abs/quant-ph/0610253).
165. J. Eisert, K. Jacobs, P. Papadopoulos, and M. B. Plenio, "Optimal Local Implementation of Nonlocal Quantum Gates," Phys. Rev. A, Volume **62** (2000) 052317.
166. J. Eisert, F. G. S. L. Brandão, and K. Audenaert, "Quantitative Entanglement Witnesses," New J. Phys., Volume **9** (2007) p. 46.
167. A. K. Ekert, "Quantum Cryptography Based on Bell's Theorem," Phys. Rev. Lett., Volume **67** (1991) pp. 661–663.
168. A. Ekert and C. Macchiavello, "Quantum Error Correction for Communication," Phys. Rev. Lett., Volume **77**, Issue 12 (1996) pp. 2585–2588.
169. C. Elektra quote <http://www.nationalledger.com/cgi-bin/artman/exec/view.cgi?archive=5&num=9028>.
170. T. ElGamal, "A Public-key Cryptosystem and a Signature Scheme based on Discrete Logarithms," IEEE Transactions on Information Theory, Volume **31** (1985) pp. 469–472.
171. C. Elliot, "The DARPA Quantum Network," in *Quantum Communications and Cryptography*, ed. Alexander Sergienko, CRC Press/Taylor & Francis, Boca Raton/London (2005) ISBN 9780849336843, pp. 83–102.
172. G. S. Engel, T. R. Calhoun, E. L. Read, T.-K. Ahn, T. Mančal, Y.-C. Cheng, R. E. Blankenship, and G. R. Fleming, "Evidence for Wavelike Energy Transfer through Quantum Coherence in Photosynthetic Systems," Nature, Volume **446** (2007) pp. 782–786.
173. D. Englund, A. Faraon, B. Zhang, Y. Yamamoto, and J. Vuckovic, "Generation and Transfer of Single Photons on a Photonic Crystal Chip," Opt. Express, Volume **15** (2007) p. 5550.
174. "EU Investigates Mystery Buggings," <http://news.bbc.co.uk/2/hi/europe/2864063.stm>.
175. B. Everett, "Tapping into Fibre Optic Cables," Network Security, Volume **2007**, Issue 5 (2007) pp. 13–16.
176. J. Fan, M. D. Eisaman, and A. Migdall, "Bright Phase-stable Broadband Fiber-based Source of Polarization-entangled Photon Pairs," Phys. Rev. A, Volume **76** (2007) 043836.
177. E. Farhi, J. Goldstone, and S. Gutmann, "Quantum Adiabatic Evolution Algorithms with Different Paths," [arXiv:quant-ph/0208135v1](https://arxiv.org/abs/quant-ph/0208135v1) (2002).
178. E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, "A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem," Science, Volume **292** (2002) pp. 472–475.

179. S. Fasel, O. Alibart, S. Tanzilli, P. Baldi, A. Beveratos, N. Gisin, and H. Zbinden, "High-quality Asynchronous Heralded Single-photon Source at Telecom Wavelength," *New J. Phys.*, Volume **6** (2004) p. 163.
180. A. M. Ferrenberg, D. P. Landau, and Y. J. Wong, "Monte Carlo Simulations: Hidden Errors from "Good" Random Number Generators," *Phys. Rev. Lett.*, Volume **69**, Issue 23 (1992) pp. 3382–3384.
181. R. Feynman, "Simulating Physics with Computers," *International Journal of Theoretical Physics*, Volume **21**, Issue 6–7 (1982) pp. 467–488.
182. R. P. Feynman, "There's Plenty of Room at the Bottom," transcript available at http://media.wiley.com/product_data/excerpt/53/07803108/0780310853.pdf.
183. A. Fijany and C. P. Williams, *Quantum Wavelet Transforms: Fast Algorithms and Complete Circuits*, Lecture Notes in Computer Science, Volume **1509**, Springer, Berlin (1999) pp. 10–33.
184. G. Fishman, *Monte Carlo: Concepts, Algorithms, and Application*, Springer, Berlin (1996), p. 7.
185. Focus Issue "Focus on Single Photons on Demand," *New J. Phys.* **6** (2004).
186. A. Franceschetti and A. Zunger, "The Inverse Band Structure Problem: Find the Atomic Configuration with Given Electronic Properties," *Nature*, Volume **402** (1999) pp. 60–63.
187. E. Fredkin and T. Toffoli, "Conservative Logic," *International Journal of Theoretical Physics*, Volume **21** (1982) pp. 219–253.
188. S. J. Freedman and J. F. Clauser, "Experimental Test of Local Hidden-Variable Theories," *Phys. Rev. Lett.*, Volume **28**, Issue 14 (1972) pp. 938–941.
189. M. H. Freedman, A. Kitaev, and Z. Wang, "Simulation of Topological Field Theories by Quantum Computers," *Commun. Math. Phys.*, Volume **227** (2002) pp. 587–603.
190. M. H. Freedman, M. Larsen, and Z. Wang, "A Modular Functor Which is Universal for Quantum Computation," *Communications in Mathematical Physics*, Volume **227**, Issue 3 (2002) pp. 605–622.
191. M. Freedman, C. Nayak, K. Shtengel, K. Walker, and Z. Wang, "A Class of P,T-Invariant Topological Phases of Interacting Electrons," *Ann. Phys.*, Volume **310**, Issue 2 (2004) pp. 428–492.
192. K. Friedl, G. Ivanyos, F. Magniez, M. Santha, and P. Sen, "Hidden Translation and Orbit Coset in Quantum Computing," in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing* (2003) pp. 1–9.
193. P. Fulde, *Electron Correlations in Molecules and Solids*, Third Edition, Springer, Berlin (1995) ISBN 3540593640, p. 50.
194. A. Furusawa, J. L. Sørensen, S. L. Braunstein, C. A. Fuchs, H. J. Kimble, and E. S. Polzik, "Unconditional Quantum Teleportation," *Science*, Volume **282**, Issue 5389 (1998) pp. 706–709.
195. R. Gallant, R. Lambert, and S. Vanstone, "Improving the Parallelized Pollard Lambda Search on Anomalous Binary Curves," *Mathematics of Computation*, Volume **69** (2000) pp. 1699–1705.
196. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York (1979).
197. A. Gepp and P. Stocks, "A Review of Procedures to Evolve Quantum Algorithms," *Genetic Programming and Evolvable Machines*, Volume **10**, Issue 2 (2009) pp. 181–228.
198. J. Gill, "Computational Complexity of Probabilistic Turing Machines," *SIAM Journal on Computing*, Volume **6**, Issue 4 (1977) pp. 675–695.
199. R. Gingrich and C. P. Williams, "Non-Unitary Probabilistic Quantum Computing," in *Proceedings of the Winter International Symposium on Information and Communication Technologies*, Hyatt Regency Cancun, Mexico, 5th–8th January (2004).
200. R. M. Gingrich, C. P. Williams, and N. J. Cerf, "Generalized Quantum Search with Parallelism," *Phys. Rev. A*, Volume **61** (2000) 052313.
201. N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum Cryptography," *Rev. Mod. Phys.*, Volume **74** (2002) p. 146.

202. D. Giulini, E. Joos, C. Kiefer, J. Kupsch, I.-O. Stamatescu, and H. D. Zeh, *Decoherence and the Appearance of a Classical World in Quantum Theory*, Springer, Berlin (1996) ISBN 3-540-61394-3.
203. E. A. Goldschmidt, M. D. Eisaman, J. Fan, S. V. Polyakov, and A. Migdall, “Spectrally Bright and Broad Fiber-based Heralded Single-photon Source,” Phys. Rev. A, Volume **78** (2008) 013844.
204. G. Golub and C. van Loan, *Matrix Computations*, Third Edition, Johns Hopkins University Press, Baltimore (1996).
205. C. Gomes and B. Selman, “Satisfied with Physics,” Science, Volume **297** (2002) pp. 784–785.
206. C. Gomes and B. Selman, “Can get Satisfaction,” Nature, Volume **435** (2005) pp. 751–752.
207. D. Gottesman, “Class of Quantum Error-Correcting Codes Saturating the Quantum Hamming Bound,” Phys. Rev. A, Volume **54**, Issue 3 (1996) pp. 1862–1868.
208. D. Gottesman, “Stabilizer Codes and Quantum Error Correction,” Ph.D. thesis, California Institute of Technology, [arXiv:quant-ph/9705052v1](https://arxiv.org/abs/quant-ph/9705052v1) (1997).
209. D. Gottesman, “An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation,” [arXiv:0904.2557v1](https://arxiv.org/abs/0904.2557v1) (2009).
210. D. Gottesman and I. L. Chuang, “Demonstrating the Viability of Universal Quantum Computation using Teleportation and Single-qubit Operations,” Nature, Volume **402** (1999) pp. 390–393.
211. M. Grassl, “Bounds on the Minimum Distance of Qubit Block Codes for Given Length and Dimension,” online at <http://www.codetables.de/> (2009).
212. M. Grassl and T. Beth, “A Note on Non-additive Quantum Codes,” [arXiv:quant-ph/9703016](https://arxiv.org/abs/quant-ph/9703016) (1997).
213. R. B. Griffiths and C. Niu, “Semiclassical Fourier Transform for Quantum Computation,” Phys. Rev. Lett., Volume **76** (1996) pp. 3228–3231.
214. F. Grosshans and P. Grangier, “Continuous Variable Quantum Cryptography Using Coherent States,” Phys. Rev. Lett., Volume **88** (2002) 057902.
215. F. Grosshans, G. Van Assche, J. Wenger, R. Brouri, N. J. Cerf, and P. Grangier, “Quantum Key Distribution using Gaussian-modulated Coherent States,” Nature, Volume **421** (2003) pp. 238–241.
216. G. Grössing and A. Zeilinger, “Quantum Cellular Automata,” Complex Syst., Volume **2** (1988) pp. 197–208.
217. L. K. Grover, “A Fast Quantum Mechanical Algorithm for Database Search,” in *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, ACM Press, New York, (1996) pp. 212–219.
218. L. K. Grover, “A Fast Quantum Mechanical Algorithm for Estimating the Median,” [arXiv:quant-ph/9607024v1](https://arxiv.org/abs/quant-ph/9607024v1) (1996).
219. L. K. Grover, “Quantum Mechanics Helps in Searching for a Needle in a Haystack,” Phys. Rev. Lett., Volume **79**, Issue 2 (1997) pp. 325–328.
220. L. Grover, “A Framework for Fast Quantum Mechanical Algorithms,” in *Proceedings of 30th Annual ACM Symposium on Theory of Computing (STOC)*, May (1998), pp. 53–62.
221. L. K. Grover, “Quantum Computers can Search Rapidly by Using Almost any Transformation,” Phys. Rev. Lett., Volume **80**, Issue 19 (1998) pp. 4329–4332.
222. L. K. Grover, “Synthesis of Quantum Superpositions by Quantum Computation,” Phys. Rev. Lett., Volume **85**, Issue 6 (2000) pp. 1334–1337.
223. L. K. Grover and T. Rudolph, “Creating Superpositions that Correspond to Efficiently Integrable Probability Distributions,” [arXiv:quant-ph/0208112v1](https://arxiv.org/abs/quant-ph/0208112v1) (2002).
224. L. Hackermüller, S. Uttenthaler, K. Hornberger, E. Reiger, B. Brezger, A. Zeilinger, and M. Arndt, “Wave Nature of Biomolecules and Fluorofullerenes,” Phys. Rev. Lett., Volume **91** (2003) 090408.
225. S. Hallgren, “Polynomial Time Quantum Algorithms for Pell’s Equation and the Principal Ideal Problem,” in *Proceedings of ACM STOC’02* (2002).

226. S. Hallgren, "Fast Quantum Algorithms for Computing the Unit Group, and Class Group of a Number Field," in *Proceedings of the 37th ACM Symposium on Theory of Computing* (2005).
227. S. Haroche and J.-M. Raimond, "Quantum Computing: Dream of Nightmare?" *Phys. Today* (1996) pp. 51–52.
228. R. Harris, A. J. Berkley, J. Johansson, M. W. Johnson, T. Lanting, P. Bunyk, E. Tolkacheva, E. Ladizinsky, B. Bumble, A. Fung, A. Kaul, A. Kleinsasser, and S. Han, "Implementation of a Quantum Annealing Algorithm Using a Superconducting Circuit," [arXiv:0903.3906](https://arxiv.org/abs/0903.3906) (2009).
229. R. Harris, M. W. Johnson, T. Lanting, A. J. Berkley, J. Johansson, P. Bunyk, E. Tolkacheva, E. Ladizinsky, N. Ladizinsky, T. Oh, F. Cioata, I. Perminov, P. Spear, C. Enderud, C. Rich, S. Uchaikin, M. C. Thom, E. M. Chapple, J. Wang, B. Wilson, M. H. S. Amin, N. Dickson, K. Karimi, B. Macready, C. J. S. Truncik, and G. Rose, "Experimental Investigation of an Eight Qubit Unit Cell in a Superconducting Optimization Processor," *Phys. Rev. B*, Volume **82**, (2010) 024511.
230. N. Hatano and M. Suzuki, "Finding Exponential Product Formulas of Higher Orders," in *Quantum Annealing and Other Optimization Methods*, Lecture Notes in Physics, Volume **679**, Springer, Berlin (2005) pp. 37–68. ISBN 978-3-540-27987-7.
231. M. Heiligman, "Quantum Algorithms for Lowest Weight Paths and Spanning Trees in Complete Graphs," [arXiv:quant-ph/0303131](https://arxiv.org/abs/quant-ph/0303131) (2003).
232. S. Hill and W. Wootters, "Entanglement of a Pair of Quantum Bits," *Phys. Rev. Lett.*, Volume **78**, Issue 26 (1997) pp. 5022–5025.
233. P. A. Hiskett, D. Rosenberg, C. G. Peterson, R. J. Hughes, S. Nam, A. E. Lita, A. J. Miller, and J. E. Nordholt, "Long-distance Quantum Key Distribution in Optical Fibre," *New J. Phys.*, Volume **8** (2006) p. 193.
234. T. Hogg, B. A. Huberman, and C. P. Williams, "Phase Transitions and the Search Problem," *Artificial Intelligence*, Volume **81**, Issue 1–2 (1996) pp. 1–15.
235. L. Hollenberg, "Fast Quantum Search Algorithms in Protein Sequence Comparisons: Quantum Bioinformatics," *Phys. Rev. E*, Volume **62**, Issue 5 (2000) pp. 7532–7535.
236. T. Honjo, S. W. Nam, H. Takesue, Q. Zhang, H. Kamada, Y. Nishida, O. Tadanaga, M. Asobe, B. Baek, R. Hadfield, S. Miki, M. Fujiwara, M. Sasaki, Z. Wang, K. Inoue, and Y. Yamamoto, "Long-distance Entanglement-based Quantum Key Distribution over Optical Fiber," *Optics Express*, Volume **16**, Issue 23 (2008) pp. 19118–19126.
237. J. Hopcroft, *Turing Machines*, Scientific American, New York (1984) pp. 86–98.
238. L. Hormozni, N. E. Bonesteel, and S. H. Simon, "Topological Quantum Computing with Read-Rezayi States," *Phys. Rev. Lett.*, Volume **103** (2009) 160501.
239. M. Horodecki, P. Horodecki, and R. Horodecki, "Separability of Mixed States: Necessary and Sufficient Conditions," *Phys. Lett. A*, Volume **223**, Issues 1–2 (1996) pp. 1–8.
240. M. Hsieh, J. Kempe, S. Myrgren, and K. B. Whaley, "An Explicit Universal Gate-Set for Exchange-Only Quantum Computation," *Quantum Information Processing*, Volume **2**, Issue 4 (2001) pp. 289–307.
241. <http://cnn.com/2000/TECH/computing/03/17/brazil.nasa.hackers/index.html>.
242. <http://cnn.com/2000/TECH/space/06/30/nasa.hacker/index.html>.
243. <http://cnn.com/2000/TECH/space/07/03/nasa.hacker.02/>.
244. <http://cnn.com/TECH/computing/9905/24/nasa.idg/>.
245. http://www.secoqc.net/downloads/pressrelease/SECOQC_english.pdf.
246. http://www.secoqc.net/downloads/pressrelease/SECOQC_PRESS%20RELEASE_english.pdf.
247. R. J. Hughes, J. E. Nordholt, D. Derkacs, and C. G. Peterson, "Practical Free-Space Quantum Key Distribution Over 10 km in Daylight and at Night," *New J. Phys.*, Volume **4** (2002) p. 43.
248. B. Huttner, J. D. Gautier, A. Muller, H. Zbinden, and N. Gisin, "Unambiguous Quantum Measurement of Nonorthogonal States," *Phys. Rev. A*, Volume **54** (1996) pp. 3783–3789.
249. id Quantique (Switzerland), <http://www.idquantique.com>.
250. idQuantique, "idQuantique on QKD Security," see <http://www.idquantique.com/network-encryption/qkd-security.html>

251. idQuantique, “Vulnerability in Commercial Quantum Cryptography Tackled by International Collaboration,” joint press release between idQuantique, Norwegian University of Science and Technology, University of Erlangen-Nürnberg, and the Max Planck Institute for the Science of Light (2010) <http://www.idquantique.com/images/stories/PDF/press-releases/pr-internationalcollaboration.pdf>.
252. K. Inoue and Y. Iwai, “Differential Quadrature Phase Shift Quantum Key Distribution,” Phys. Rev. A, Volume **79** (2009) 022319.
253. S. Ishizaka and M. B. Plenio, “Multiparticle Entanglement under Asymptotic Positive-Partial-Transpose-Preserving Operations,” Phys. Rev. A, Volume **72** (2005) 042325.
254. I. D. Ivanovic, “How to Differentiate Between Nonorthogonal States,” Phys. Lett. A, Volume **123** (1987) pp. 257–259.
255. M. Jerrum, A. Sinclair, and E. Vigoda, “A Polynomial-Time Approximation Algorithm for the Permanent of a Matrix with Non-Negative Entries,” University of Edinburgh preprint, 27th June (2003). Available at <http://www.dcs.ed.ac.uk/home/mrj/PermanentRev.pdf>.
256. S. Johnsen and K. J. Lohmann, “Magnetoreception in Animals,” Physics Today (2008) pp. 29–35.
257. E. Joos and H. Zeh, “The Emergence of Classical Properties through Interaction with the Environment,” Zeitschrift fur Physik B, Volume **59** (1985) pp. 223–243.
258. S. P. Jordan, “Fast Quantum Algorithm for Numerical Gradient Estimation,” Phys. Rev. Lett., Volume **95** (2005) 050501.
259. S. P. Jordan, “Fast Quantum Algorithms for Approximating the Irreducible Representations of Groups,” [arXiv:0811.0562](https://arxiv.org/abs/0811.0562) (2008).
260. S. P. Jordan, E. Farhi, and P. W. Shor, “Error Correcting Codes For Adiabatic Quantum Computation,” Phys. Rev. A, Volume **74**, (2006) 052322.
261. R. Jozsa, “Characterizing Classes of Functions Computable by Quantum Parallelism,” Proc. R. Soc. Lond. A, Volume **435** (1991) pp. 563–574.
262. R. Jozsa, *Quantum Factoring, Discrete Logarithms and the Hidden Subgroup Problem*, Computing in Science and Engineering, IEEE Computer Society, Los Alamitos (2001) pp. 34–43.
263. R. Jozsa, “Quantum Computation in Algebraic Number Theory: Hallgren’s Efficient Quantum Algorithm for Solving Pell’s Equation,” Ann. Phys., Volume **306**, Issue 2 (2003) pp. 241–279.
264. B. Kahr, J. Freudenthal, S. Phillips, and W. Kaminsky, “Herapathite,” Science, Volume **324**, Issue 5933 (2009) p. 1407.
265. W. M. Kaminsky and S. Lloyd, “Scalable Architecture for Adiabatic Quantum Computing of NP-Hard Problems,” in *Quantum Computing and Quantum Bits in Mesoscopic Systems*, eds. A. J. Leggett, B. Ruggiero, and P. Silvestrini, Springer, Berlin, (2003) pp. 229–236.
266. I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, and A. Aspuru-Guzik, “Polynomial-Time Quantum Algorithm for the Simulation of Chemical Dynamics,” Proceedings of the National Academy of Sciences, Volume **105** (2008) pp. 18681–18686.
267. P. Kaye and M. Mosca, “Quantum Networks for Generating Arbitrary Quantum States,” in *Proceedings of the International Conference on Quantum Information (ICQI)*, Rochester, New York, USA (2001).
268. P. Kaye and M. Mosca, “Quantum Networks for Concentrating Entanglement,” J. Phys. A: Math. Gen., Volume **34** (2001) pp. 6939–6948.
269. P. Kaye and C. Zalka, “Optimized Quantum Implementation of Elliptic Curve Arithmetic over Binary Fields,” Quantum Information and Computation, Volume **5**, Issue 6 (2005) pp. 474–491.
270. J. Kempe, “Approaches to Quantum Error Correction,” Séminaire Poincaré, Volume **2** (2005) pp. 1–29.
271. J. Kempe, D. Bacon, D. P. Di Vincenzo, and K. B. Whaley, “Encoded Universality from a Single Physical Interaction,” Quantum Information and Computation, Volume **1** (2001) pp. 33–55 (Special Issue).
272. J. Kempe, D. Bacon, D. A. Lidar, and K. B. Whaley, “Theory of Decoherence-Free Fault-tolerant Universal Quantum Computation,” Phys. Rev. A, Volume **63** (2001) 042307.

273. J. Kempe, A. Kitaev, and O. Regev, "The Complexity of the Local Hamiltonian Problem," *SIAM Journal of Computing*, Volume **35**, Issue 5 (2006) pp. 1070–1097.
274. V. Kendon, "A Random Walk Approach to Quantum Algorithms," *Phil. Trans. R. Soc. A*, Volume **364** (2006) pp. 3407–3422.
275. V. Kendon and O. Maloyer, "Optimal Computation with Non-unitary Quantum Walks," *Theoretical Computer Science*, Volume **394**, Issue 3 (2008) pp. 187–196.
276. L. G. Khachiyan, "A Polynomial Algorithm for Linear Programming," *Soviet Math Doklady*, Volume **20** (1979) pp. 191–194.
277. N. Khaneja, R. Brockett, and S. J. Glaser, "Time Optimal Control in Spin Systems," *Phys. Rev. A*, Volume **63** (2001) 032308.
278. Y. Kim, S. P. Kulik, and Y. Shih, "Quantum Teleportation of a Polarization State with a Complete Bell State Measurement," *Phys. Rev. Lett.*, Volume **86**, Issue 7 (2001) pp. 1370–1373.
279. Y.-H. Kim, S. P. Kulik, and Y. Shih, "Quantum Teleportation of a Polarization State with a Complete Bell State Measurement," *Phys. Rev. Lett.*, Volume **86**, Issue 7 (2001) pp. 1370–1373.
280. S. Kirkpatrick and B. Selman, "Critical Behavior in the Satisfiability of Random Boolean Expressions," *Science*, Volume **264** (1994) pp. 1297–1301.
281. S. Kirkpatrick and B. Selman, "Insights from Statistical Physics into Computational Complexity," in *More is Different—Fifty Years of Condensed Matter Physics*, eds. N. Ong and R. Bhatt, Princeton Series in Physics (2001) pp. 331–339.
282. A. Kitaev, "Quantum Measurements and the Abelian Stabilizer Problem," [arXiv:quant-ph/9511026](https://arxiv.org/abs/quant-ph/9511026) (1995).
283. A. Y. Kitaev, "Quantum Computations: Algorithms and Error Correction," *Russian Math. Surveys*, Volume **52**, Issue 6 (1997) pp. 1191–1249.
284. A. Y. Kitaev, "Quantum Computations: Algorithms and Error Correction," *Russ. Math. Surv.*, Volume **52**, Issue 6 (1997) pp. 1191–1249.
285. A. Yu. Kitaev, "Fault-Tolerant Quantum Computation by Anyons," *Ann. Phys.*, Volume **303**, Issue 1 (2003) pp. 2–30. Based on preprint [arXiv:quant-ph/9707021v1](https://arxiv.org/abs/quant-ph/9707021v1) (1997).
286. S. Kivelson, D. H. Lee, and S. C. Zhang, *Electrons in Flatland*, Scientific American, New York (1996) pp. 86–91.
287. A. Klappenecker and M. Roetteler, "Discrete Cosine Transforms on Quantum Computers," *IEEE ISPA01*, Pula, Croatia (2001).
288. A. Klappenecker and M. Roetteler, "On the Irresistible Efficiency of Signal Processing Methods in Quantum Computing," [arXiv:quant-ph/0111039v1](https://arxiv.org/abs/quant-ph/0111039v1) (2001).
289. A. Klappenecker and M. Rötteler, "Quantum Software Reusability," *International Journal of Foundations of Computer Science*, Volume **14**, Issue 5 (2003) pp. 777–796.
290. A. Klappenecker and M. Rötteler, "Engineering Functional Quantum Algorithms," *Phys. Rev. A*, Volume **67** (2003) 010302.
291. E. Knill, "Quantum Computing with Realistically Noisy Devices," *Nature*, Volume **434** (2005) pp. 39–44.
292. E. Knill and R. Laflamme, "A Theory of Quantum Error-correcting Codes," *Phys. Rev. A*, Volume **55** (1997) pp. 900–911.
293. E. Knill, R. Laflamme, and G. J. Milburn, "A Scheme for Efficient Quantum Computation with Linear Optics," *Nature*, Volume **409** (2001) pp. 46–52.
294. N. Koblitz, "Elliptic Curve Cryptosystems," *Mathematics of Computation*, Volume **48** (1987) pp. 203–209.
295. P. Kok, C. P. Williams, and J. P. Dowling, "Construction of a Quantum Repeater with Linear Optics," *Phys. Rev. A*, Volume **68** (2003) 022301.
296. B. Kraus and J. I. Cirac, "Optimal Creation of Entanglement Using a Two-Qubit Gate," *Phys. Rev. A*, Volume **63** (2001) 062309.
297. R. Laflamme, C. Miquel, P. Paz, and W. Zurek, "Perfect Quantum Error Correcting Code," *Phys. Rev. Lett.*, Volume **77** (1996) pp. 198–201.

298. A. M. Lance, T. Symul, V. Sharma, C. Weedbrook, T. C. Ralph, and P. K. Lam, “No-Switching Quantum Key Distribution using Broadband Modulated Coherent Light,” *Phys. Rev. Lett.*, Volume **95** (2005) 180503.
299. R. Landauer, “Irreversibility and Heat Generation in the Computing Process,” *IBM Journal of Research and Development*, Volume **5** (1961) pp. 183–191.
300. R. Landauer, “Is Quantum Mechanics Useful?” *Philosophical Transactions of the Royal Society, London, Series A*, Volume **353** (1995) pp. 367–376.
301. R. Landauer, “The Physical Nature of Information,” *Phys. Lett. A*, Volume **217** (1996) p. 188.
302. K. J. Lang, P. McKenzie, and A. Tapp, “Reversible Space Equals Deterministic Space,” *Journal of Computers and System Science*, Volume **60** (2000) pp. 354–367.
303. T. Lanting, R. Harris, J. Johansson, M. H. S. Amin, A. J. Berkley, S. Gildert, M. W. Johnson, P. Bunyk, E. Tolkacheva, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, E. M. Chapple, C. Enderud, C. Rich, B. Wilson, M. C. Thom, S. Uchaikin, and G. Rose, “Cotunneling in Pairs of Coupled Flux Qubits,” *Phys. Rev. B*, Volume **82**, (2010) 060512(R).
304. B. P. Lanyon, J. D. Whitfield, G. G. Gillet, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, A. Aspuru-Guzik, and A. G. White, “Towards Quantum Chemistry on a Quantum Computer,” *Nature Chemistry*, Volume **2** (2009) pp. 106–111.
305. R. B. Laughlin, “Anomalous Quantum Hall Effect: An Incompressible Quantum Fluid with Fractionally Charged Excitations,” *Phys. Rev. Lett.*, Volume **50**, Issue 18 (1982) pp. 1395–1398.
306. H. Lee, Y.-C. Cheng, and G. R. Fleming, “Coherence Dynamics in Photosynthesis: Protein Protection of Excitonic Coherence,” *Science*, Volume **316** (2007) pp. 1462–1465.
307. H. W. Lenstra Jr., “Solving the Pell Equation,” *Notices of the American Mathematical Society*, Volume **49**, Issue 2 (2002) pp. 182–192.
308. A. K. Lenstra and H. W. Lenstra Jr., *The Development of the Number Field Sieve*, Lecture Notes in Mathematics, Volume **1554**, Springer, Berlin (1993).
309. A. K. Lenstra, H. W. Lenstra Jr., M. Manasse, and J. Pollard, “The Number Field Sieve,” in *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, Baltimore (1990) pp. 564–572.
310. D. W. Leung, “Quantum Computation by Measurements,” *Int. J. Quantum Inform.*, Volume **2** (2004) pp. 33–43.
311. R. Y. Levine and A. T. Sherman, “A Note on Bennett’s Time-space Tradeoff for Reversible Computation,” *SIAM Journal on Computing*, Volume **19** (1990) pp. 673–677.
312. M. Li, J. Tromp, and P. Vitányi, “Reversible Simulation of Irreversible Computation,” *Physica D*, Volume **120** (1998) pp. 168–176.
313. D. A. Lidar, “Towards Fault Tolerant Adiabatic Quantum Computation,” *Phys. Rev. Lett.*, Volume **100**, (2008) 160506.
314. D. A. Lidar, I. L. Chuang, and K. B. Whaley, “Decoherence-Free Subspaces for Quantum Computation,” *Phys. Rev. Lett.*, Volume **81** (1998) p. 2594.
315. D. A. Lidar, D. Bacon, and K. B. Whaley, “Concatenating Decoherence-Free Subspaces with Quantum Error Correcting Codes,” *Phys. Rev. Lett.*, Volume **82** (1999) p. 4556.
316. D. A. Lidar, D. Bacon, J. Kempe, and K. B. Whaley, “Protecting Quantum Information Encoded in Decoherence-Free States against Exchange Errors,” *Phys. Rev. A*, Volume **61** (2000) 052307.
317. D. A. Lidar, D. Bacon, J. Kempe, and K. B. Whaley, “Decoherence-Free Subspaces for Multiple-Qubit Errors: II. Universal, Fault-Tolerant Quantum Computation,” *Phys. Rev. A*, Volume **63** (2001) 022307.
318. D. A. Lidar, D. Bacon, J. Kempe, and K. B. Whaley, “Decoherence-Free Subspaces for Multiple-Qubit Errors: I. Characterization,” *Phys. Rev. A*, Volume **63** (2001) 022306.
319. E. H. Lieb and F. Y. Wu, “The One-Dimensional Hubbard Model: A Reminiscence,” [arXiv:cond-mat/0207529v2](https://arxiv.org/abs/cond-mat/0207529v2) (2002).
320. S. Lloyd, “Quantum Mechanical Computers and Uncomputability,” *Phys. Rev. Lett.*, Volume **71** (1993) pp. 943–946.

321. S. Lloyd, "Universal Quantum Simulators," *Science*, Volume **273** (1996) pp. 1073–1078.
322. H. K. Lo, X. Ma, and K. Chen, "Decoy State Quantum Key Distribution," *Phys. Rev. Lett.*, Volume **94** (2005) 230504.
323. C. Lomont, "Quantum Convolution and Quantum Correlation Algorithms are Physically Impossible," [arXiv:quant-ph/0309070v2](https://arxiv.org/abs/quant-ph/0309070v2) (2003).
324. C. Lomont, "The Hidden Subgroup Problem—Review and Open Problems," [arXiv:quant-ph/0411037v1](https://arxiv.org/abs/quant-ph/0411037v1) (2004).
325. S. Lorenz, N. Korolkova, and G. Leuchs, "Continuous Variable Quantum Key Distribution using Polarization Encoding and Post Selection," *Appl. Phys. B*, Volume **79**, Issue 3 (2004) pp. 273–277.
326. D. Loss and D. P. Di Vincenzo, "Quantum Computation with Quantum Dots," *Phys. Rev. A*, Volume **57** (1998) pp. 120–126.
327. M. Lukac and M. Perkowski, "Evolving Quantum Circuits Using Genetic Algorithm," in *Proceedings of the 2002 NASA/DoD Conference on Evolvable Hardware (EH'02)*, 15th–18th July (2002) p. 177.
328. M. Lukac and M. Perkowski, "Evolutionary Approach to Quantum Symbolic Logic Synthesis," in *Proceedings of the 2008 IEEE Congress on Evolutionary Computation (CEC2008)*, June (2008) pp. 3374–3380.
329. L. Lydersen and J. Skaar, "Security of Quantum Key Distribution with Bit and Basis Dependent Detector Flaws," *Quantum Information and Computation*, Volume **10** (2010) 0060.
330. L. Lydersen, C. Wiechers, C. Wittmann, D. Elser, J. Skaar, and C. Makarov, "Thermal Blinding of Gated Detectors in Quantum Cryptography," [arXiv:1009.2663](https://arxiv.org/abs/1009.2663) [quant-ph] (2010).
331. L. Lydersen, C. Wiechers, C. Wittmann, D. Elser, J. Skaar, and C. Makarov, "Hacking Commercial Quantum Cryptography Systems by Tailored Bright Illumination," *Nature Photonics*, Volume **4** (2010) p. 686.
332. I. Lynce and J. Marques-Silva, "Building State-of-the-Art SAT Solvers," in *Proc. of the European Conference on Artificial Intelligence*, IOS Press, Amsterdam (2002).
333. MagiQ Technologies (U.S.A.), <http://www.magiqtech.com/>.
334. A. Mair, A. Vaziri, G. Weihs, and A. Zeilinger, "Entanglement of the Orbital Angular Momentum States of Photons," *Nature*, Volume **412** (2001) pp. 313–316.
335. V. Makarov, "Exploiting the Saturation Mode of Passively Quenched Avalanche Photodiodes to Attack Quantum Cryptosystems," in *Proceedings of the Optical Society of Korea Annual Meeting '08* (2008) pp. 417–418.
336. V. Makarov and J. Skaar, "Faked States Attack using Detector Efficiency Mismatch on SARG04, Phase-Time, DPSK, and Ekert Protocols," *Quantum Information and Computation*, Volume **8** (2008) 0622.
337. V. Makarov, A. Anisimov, and S. Sauge, "Quantum Hacking: Adding a Commercial Actively-Quenched Module to the List of Single-Photon Detectors Controllable by Eve," [arXiv:0809.3408](https://arxiv.org/abs/0809.3408) [quant-ph] (2008).
338. V. Makarov, A. Anisimov, and J. Skaar, "Effects of Detector Efficiency Mismatch on Security of Quantum Cryptosystems," *Phys. Rev. A*, Volume **74** (2006) 022313. Erratum in Volume **78** (2008) 019905.
339. S. Mancini, S. Lloyd, S. L. Braunstein, and S. Pirandola, "Continuous-variable Quantum Cryptography using Two-way Quantum Communication," *Nature Physics*, Volume **4** (2008) pp. 726–730.
340. I. Marcikic, H. de Riedmatten, W. Tittel, H. Zbinden, and N. Gisin, "Long-Distance Teleportation of Qubits at Telecommunication Wavelengths," *Nature*, Volume **421** (2003) pp. 509–513.
341. Ø. Marøy, L. Lydersen, and J. Skaar, "Security of Quantum Key Distribution with Arbitrary Individual Imperfections," *Phys. Rev. A*, Volume **82** (2010) 032337.
342. C. Marr, A. Beige, and G. Rempe, "Entangled State Preparation via Dissipation-Assisted Adiabatic Passages," *Phys. Rev. A*, Volume **68** (2003) 033817.
343. D. Maslov and G. W. Dueck, "Reversible Cascades with Minimal Garbage," *IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems*, Volume **23**, Issue 11 (2004) pp. 1497–1509.

344. D. Maslov, J. Mathew, D. Cheung, and D. K. Pradhan, “An $\mathcal{O}(m^2)$ -depth Quantum Algorithm for the Elliptic Curve Discrete Logarithm Problem over $GF(2^m)$,” *Quantum Information and Computation*, Volume **9**, Issue 7–8 (2009) pp. 0610–0621.
345. P. Massey, J. A. Clark, and S. A. Stepney, “Human-Competitive Evolution of Quantum Computing Artefacts by Genetic Programming,” *Evolutionary Computation*, Volume **14**, Issue 1 (2006) pp. 21–40.
346. D. N. Matsukevich et al., “Bell Inequality Violation with Two Remote Atomic Qubits,” *Phys. Rev. Lett.*, Volume **100** (2008) 150404.
347. D. McCullagh and A. Broache, “NSA Eavesdropping: How it Might Work,” CNET News.com, February 7 (2006).
348. D. McGloin, N. B. Simpson, and M. J. Padgett, “The Transfer of Orbital Angular Momentum from a Stressed Fibre-optic Waveguide to a Light Beam,” *Appl. Opt.*, Volume **37** (1998) pp. 469–472.
349. R. Merkle, “Secure Communication over an Insecure Channel,” *Commun. Ass. Comp. Mach.*, Volume **21** (1978) pp. 294–299.
350. T. Metodiev, A. Cross, D. Thaker, K. Brown, D. Copsey, F. T. Chong, and I. Chuang, “Preliminary Results on Simulating a Scalable Fault Tolerant Ion-Trap System for Quantum Computation,” in *Third Workshop on Non-Silicon Computing (NSC-3)*, June (2004).
351. M. Mezard, G. Parisi, and R. Zecchina, “Analytic and Algorithmic Solution of Random Satisfiability Problems,” *Science*, Volume **297** (2002) pp. 812–815.
352. A. L. Migdall, D. Branning, and S. Castelletto, “Tailoring Single-photon and Multiphoton Probabilities of a Single-photon On-demand Source,” *Phys. Rev. A*, Volume **66** (2002) 053805.
353. V. Miller, “Uses of Elliptic Curves in Cryptography,” in *Advances in Cryptology CRYPTO’85*, Lecture Notes in Computer Science, Volume **218**, Springer, Berlin (1986) pp. 417–426.
354. C. Miquel, J. P. Paz, M. Saraceno, E. Knill, R. Laflamme, and C. Negrevergne, “Interpretation of Tomography and Spectroscopy as Dual forms of Quantum Computation,” *Nature*, Volume **418** (2002) pp. 59–62.
355. D. R. Mitchell, C. Adami, W. Lue, and C. P. Williams, “Random Matrix Model of Adiabatic Quantum Computing,” *Phys. Rev. A*, Volume **71** (2005) 052324.
356. Y. Mitsumori, J. A. Vaccaro, S. M. Barnett, E. Andersson, A. Hasegawa, M. Takeoka, and M. Sasaki, “Experimental Demonstration of Quantum Source Coding,” *Phys. Rev. Lett.*, Volume **91** (2003).
357. C. Mochon, “Anyons from Nonsolvable Finite Groups are Sufficient for Universal Quantum Computation,” *Phys. Rev. A*, Volume **67** (2003) 022315.
358. C. Mochon, “Anyon Computers with Smaller Groups,” *Phys. Rev. A*, Volume **69** (2004) 032306.
359. M. Mohseni, P. Rebentrost, S. Lloyd, and A. Aspuru-Guzik, “Environment-Assisted Quantum Walks in Photosynthetic Energy Transfer,” *Journal of Chemical Physics*, Volume **129** (2008) 174106.
360. G. Molina-Terriza, J. P. Torres, and L. Torner, “Management of the Angular Momentum of Light: Preparation of Photons in Multidimensional Vector States of Angular Momentum,” *Phys. Rev. Lett.*, Volume **88** (2001) 013601.
361. R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky, “Determining the Computational Complexity from Characteristic ‘Phase Transitions’,” *Nature*, Volume **400** (1999) pp. 133–137.
362. M. Mosca and A. Ekert, “The Hidden Subgroup Problem and Eigenvalue Estimation on a Quantum Computer,” *Lecture Notes in Computer Science*, Volume **1509**, Springer, Berlin (1999) pp. 174–188.
363. M. Möttönen, J. Vartiainen, V. Bergholm, and M. Salomaa, “Quantum Circuits for General Multiqubit Gates,” *Phys. Rev. Lett.*, Volume **93** (2004) 130502.
364. R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, Cambridge (1997) ISBN 0-521-47465-5.

365. C. Nayak, S. H. Simon, A. Stern, M. Freedman, and S. Das Sarma, “Non-Abelian Anyons and Topological Quantum Computation,” Rev. Mod. Phys., Volume **80**, Issue 3 (2008) p. 1083.
366. H. Neven, G. Rose, and W. G. Macready, “Image Recognition with an Adiabatic Quantum Computer I. Mapping to Quadratic Unconstrained Binary Optimization,” [arXiv:0804.4457](https://arxiv.org/abs/0804.4457) (2008).
367. H. Neven, V. S. Denchev, G. Rose, and W. G. Macready, “Training a Binary Classifier with the Quantum Adiabatic Algorithm,” [arXiv:0811.0416](https://arxiv.org/abs/0811.0416) (2008).
368. H. Neven, V. S. Denchev, G. Rose, and W. G. Macready, “Training a Large Scale Classifier with the Quantum Adiabatic Algorithm,” [arXiv:0912.0779](https://arxiv.org/abs/0912.0779) (2009).
369. I. Newton, in *Isaac Newton: Philosophical Writings*, ed. A. Janiak, Cambridge University Press, Cambridge (2004).
370. New Yorker article http://www.newyorker.com/fact/content/?040802fa_fact.
371. M. A. Nielsen, “Quantum Computation by Measurement and Quantum Memory,” Phys. Lett. A, Volume **308** (2003) pp. 96–100.
372. “NSA Suite B Cryptography.” See http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml, 15th January (2009).
373. R. W. Ogburn and J. Preskill, *Topological Quantum Computation*, Lecture Notes in Computer Science, Volume **1509**, Springer, Berlin (1999) pp. 341–356.
374. S. Olmschenk, D. N. Matsukevich, P. Maunz, D. Hayes, L.-M. Duan, and C. Monroe, “Quantum Teleportation Between Distant Matter Qubits,” Science, Volume **323**, Issue 5913 (2009) pp. 486–489.
375. G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme, “Quantum Algorithms for Fermionic Simulations,” Phys. Rev. A, Volume **64** (2001) 022319.
376. G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme, “Erratum: Quantum Algorithms for Fermionic Simulations [Phys. Rev. A, Volume **64** (2001) 022319],” Phys. Rev. A, Volume **65** (2002) 029902.
377. C. H. Papadimitriou, “The Euclidean Travelling Salesman Problem is NP-Complete,” Theor. Comput. Sci., Volume **4**, Issue 3 (1977) pp. 237–244.
378. C. Paterson, “Atmospheric Turbulence and Orbital Angular Momentum of Single Photons for Optical Communications,” Phys. Rev. Lett., Volume **94** (2005) 153901.
379. R. Paturi, P. Pudlák, M. E. Saks, and F. Zane, “An Improved Exponential Time Algorithm for k -SAT,” in *Proc. IEEE 39th Symp. Foundations of Computer Science*, IEEE Computer Society Press, Los Alamitos (1998) pp. 628–637.
380. J. P. Paz and A. Roncaglia, “A Quantum Gate Array can be Programmed to Evaluate the Expectation Value of any Operator,” Phys. Rev. A, Volume **68** (2003) 052316.
381. M. Pelton, C. Santori, J. Vuckovic, B. Zhang, G. S. Solomon, J. Plant, and Y. Yamamoto, “Efficient Source of Single Photons: A Single Quantum Dot in a Micropost Microcavity,” Phys. Rev. Lett., Volume **89** (2002) 233602.
382. C. Z. Peng, J. Zhang, D. Yang, W. B. Gao, H. X. Ma, H. Yin, H. P. Zeng, T. Yang, X. B. Wang, and J. W. Pan, “Experimental Long-Distance Decoy-State Quantum Key Distribution Based on Polarization Encoding,” Phys. Rev. Lett., Volume **98** (2007) 010505.
383. J. M. Perdigues Armengol, B. Furch, C. J. de Matos, O. Minster, L. Cacciapuoti, M. Pfennigbauer, M. Aspelmeyer, T. Jennewein, R. Ursin, T. Schmitt-Manderbach, G. Baister, J. Rarity, W. Leeb, C. Barbieri, H. Weinfurter, and A. Zeilinger, “Quantum Communications at ESA: Towards a Space Experiment on the ISS,” in *Proceedings of the 58th International Astronautical Congress*, Hyderabad, India, 24th–28th September, IAF/IAA (2007).
384. A. Perdomo, C. Truncik, I. Tubert-Brohman, G. Rose, and A. Aspuru-Guzik, “Construction of Model Hamiltonians for Adiabatic Quantum Computation and its Application to Finding Low-Energy Conformations of Lattice Protein Models,” Phys. Rev. A, Volume **78** (2008) 012320.
385. A. Peres, “Einstein, Gödel, Bohr,” Foundations of Physics, Volume **15** (1985) pp. 201–205.
386. A. Peres, “How to Differentiate Between Two Nonorthogonal States,” Phys. Lett. A, Volume **128** (1988) p. 19.
387. A. Peres, “Separability Criterion for Density Matrices,” Phys. Rev. Lett., Volume **77** (1996) pp. 1413–1415.

388. A. Peres, "Error Symmetrization in Quantum Computers," *Int. J. Theor. Phys.*, Volume **38** (1999) pp. 799–805.
389. A. Peres, "How the No-Cloning Theorem Got its Name," [arXiv:quant-ph/0205076v1](https://arxiv.org/abs/quant-ph/0205076v1) (2002).
390. A. Peres and W. Zurek, "Is Quantum Theory Universally Valid?", *American Journal of Physics*, Volume **50** (1982) pp. 807–810.
391. C. A. Perez-Delgado and D. Cheung, "Local Unitary Quantum Cellular Automata," *Phys. Rev. A*, Volume **76** (2007) 032320.
392. N. A. Peters, K. J. Arnold, A. P. VanDevender, E. R. Jeffrey, R. Rangarajan, O. Hosten, J. T. Barreiro, J. B. Altepeter, and P. G. Kwiat, "Towards a Quasi-deterministic Single-photon Source," *Proc. SPIE*, Volume **6305** (2006) 630507.
393. T. B. Pittman, B. C. Jacobs, and J. D. Franson, "Single Photons on Pseudodemand from Stored Parametric Down-conversion," *Phys. Rev. A*, Volume **66** (2002) 042303.
394. M. B. Plenio and S. F. Huelga, "Dephasing Assisted Transport: Quantum Networks and Biomolecules," *New J. Phys.*, Volume **10** (2008) 113019.
395. J. Pollard, "Monte Carlo Methods for Index Computation Mod p ," *Mathematics of Computation*, Volume **32** (1978) pp. 918–924.
396. C. Pomerance, "A Tale of Two Sieves," *Notices of the American Mathematical Society* (1996) pp. 1473–1485.
397. E. Post, "Finite Combinatory Processes—Formulation I," *J. Symb. Logic*, Volume **1** (1936) pp. 103–105.
398. J. Preskill, "Fault-Tolerant Quantum Computation," in *Introduction to Quantum Computation and Information*, eds. H. K. Lo, S. Popescu, and T. Spiller, World Scientific, Singapore (1998) ISBN 981023399X, pp. 213–269.
399. J. Preskill, "Reliable Quantum Computers," *Proc. R. Soc. Lond. A*, Volume **454** (1998) pp. 385–410.
400. J. Preskill, "Fault-Tolerant Quantum Computation," in *Introduction to Quantum Computation and Information*, eds. H.-K. Lo, S. Popescu, and T. P. Spiller, World Scientific, Singapore (1998) pp. 213–269.
401. Press release "QuintessenceLabs Announces Partnership with Lockheed Martin Corporation," <http://www.quintessencelabs.com/global/docs/PRESS-090622-QuintessenceLabs-LM-Alliance.pdf>, 22nd June (2009).
402. Press release "SmartQuantum Beef's up its Development in North America," http://www.smartquantum.com/IMG/pdf/CPSMQ_PnP-UK-4.pdf, 9th February (2009).
403. J. Proos and C. Zalka, "Shor's Discrete Logarithm Quantum Algorithm for Elliptic Curves," *Quantum Information and Computation*, Volume **3**, Issue 4 (2003) pp. 317–344.
404. Public domain U.S. government report, "Critical Infrastructure Protection: Commercial Satellite Security Should Be More Fully Addressed," GAO-02-781, August (2002). Available at <http://www.gao.gov/new.items/d02781.pdf>.
405. "Quantis Quantum Random Number Generators," Sold by idQuantique (www.idquantique.com). See <http://www.idquantique.com/products/quantis.htm>.
406. Quintessence Laboratories (Australia), <http://www.quintessencelabs.com/>.
407. E. M. Rains, R. H. Hardin, P. W. Shor, and N. J. A. Sloane, "A Nonadditive Quantum Code," *Phys. Rev. Lett.*, Volume **79** (1997) pp. 953–954.
408. T. C. Ralph, "Continuous Variable Quantum Cryptography," *Phys. Rev. A*, Volume **61** (1999) 010303(R).
409. R. Raussendorf and H. J. Briegel, "A One-Way Quantum Computer," *Phys. Rev. Lett.*, Volume **86** (2001) pp. 5188–5191.
410. R. Raussendorf and H. J. Briegel, "Computational Model Underlying the One-Way Quantum Computer," *Quantum Information and Computation*, Volume **2** (2002) pp. 443–486.
411. R. Raussendorf, D. E. Browne, and H. J. Briegel, "Measurement-Based Quantum Computation on Cluster States," *Phys. Rev. A*, Volume **68** (2003) 022312.
412. R. Raussendorf, J. Harrington, and K. Goyal, "Topological Fault-Tolerance in Cluster State Quantum Computation," *New J. Phys.*, Volume **9** (2007) p. 199.
413. P. Rebentrost, M. Mohseni, and A. Aspuru-Guzik, "Role of Quantum Coherence in Chremophoric Energy Transport," *Journal of Physical Chemistry B*, Volume **113** (2009) p. 9942.

414. P. Rebentrost, M. Mohseni, I. Kassal, S. Lloyd, and A. Aspuru-Guzik, "Environment-Assisted Quantum Transport," *New J. Phys.*, Volume **11** (2009) 033003.
415. M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, "Experimental Realization of any Discrete Unitary Operator," *Phys. Rev. Lett.*, Volume **73**, Issue 1 (1994) pp. 58–61.
416. O. Regev, "Quantum Computation and Lattice Problems," *SIAM Journal on Computing*, Volume **33**, Issue 3 (2004) pp. 738–760.
417. A. T. Rezakhani, A. K. Pimachev, and D. A. Lidar, "Accuracy vs Run Time in Adiabatic Quantum Search," [arXiv:1008.0863](https://arxiv.org/abs/1008.0863) (2010).
418. M. Riebe, H. Häffner, C. F. Roos, W. Hänsel, J. Benhelm, G. P. T. Lancaster, T. W. Körber, C. Becher, F. Schmidt-Kaler, D. F. V. James, and R. Blatt, "Deterministic Quantum Teleportation with Atoms," *Nature*, Volume **429** (2004) pp. 734–737.
419. R. Rivest, A. Shamir, and L. Adleman, "On Digital Signatures and Public Key Cryptosystems," *Commun. Ass. Comp. Mach.*, Volume **21** (1978) pp. 120–126.
420. T. Robb, "Animating Schrödinger's Equation in Two Dimensions," available at <http://library.wolfram.com/infocenter/MathSource/453/> (1993). Link verified January 2010.
421. J. Roland and N. J. Cerf, "Quantum Search by Local Adiabatic Evolution," *Phys. Rev. A*, Volume **65**, (2002) 042308.
422. J. Roland and N. J. Cerf, "Adiabatic Quantum Search Algorithm for Structured Problems," *Phys. Rev. A*, Volume **68** (2003) 062312.
423. G. Rose and W. G. Macready, "An Introduction to Quantum Annealing," DWave Technical Document 0712, http://dwave.files.wordpress.com/2007/08/20070810_d-wave_quantum_annealing.pdf.
424. D. Rosenberg, J. W. Harrington, P. R. Rice, P. A. Hiskett, C. G. Peterson, R. J. Hughes, A. E. Lita, S. W. Nam, and J. E. Nordholt, "Long-Distance Decoy-State Quantum Key Distribution in Optical Fiber," *Phys. Rev. Lett.*, Volume **98** (2007) 010503.
425. M. Rosing, *Implementing Elliptic Curve Cryptography*, Manning Publications, Greenwich (1999) ISBN 1-884777-69-4.
426. M. A. Rowe et al., "Experimental Violation of a Bell's Inequality with Efficient Detection," *Nature*, Volume **409** (2001) pp. 791–794.
427. M. B. Ruskai, "Comments on Adiabatic Quantum Algorithms," *Contemporary Mathematics*, Volume **307** (2002) pp. 265–274.
428. N. Sangouard, R. Dubessy, and C. Simon, "Quantum Repeaters based on Single Trapped Ions," *Phys. Rev. A*, Volume **79** (2009) 042340.
429. M. Santha, "Introduction to Quantum Computing," available at http://www.iri.fr/~santha/Cours/quantum_intro.pdf.
430. C. Santori, M. Pelton, G. Solomon, Y. Dale, and Y. Yamamoto, "Triggered Single Photons from a Quantum Dot," *Phys. Rev. Lett.*, Volume **86** (2001) p. 1502.
431. M. Sasaki, A. Carlini, and R. Jozsa, "Quantum Template Matching," *Phys. Rev. A*, Volume **64**, Issue 2 (2001) 022317.
432. S. Sauge, V. Makarov, and A. Anisimov, "Quantum Hacking: How Eve can Exploit Component Imperfections to Control yet another of Bob's Single-Photon Qubit Detectors," presented at CLEO/Europe-EQEC 2009, Munich, Germany, June 14th–19th (2009).
433. V. Scarani, A. Acín, J. G. Ribordy, and N. Gisin, "Quantum Cryptography Protocols Robust against Photon Number Splitting Attacks for Weak Laser Pulse Implementations," *Phys. Rev. Lett.*, Volume **92**, Issue 5 (2004) 057901.
434. L. Schiff, *Quantum Mechanics*, McGraw–Hill, New York (1955).
435. E. Schillinger, "Money Changes Hands in Key Bank Transaction," *Nature*, Volume **428** (2004) p. 883.
436. G. Schmid, "Report on the Existence of a Global System for the Interception of Private and Commercial Communications (ECHELON interception system) (2001/2098(INI))," A5-0264/2001 PAR1, Temporary Committee on the ECHELON Interception System, July 11 (2001). Available at http://www.fas.org/irp/program/process/rapport_echelon_en.pdf.
437. A. Schmidt and U. Vollmer, "Polynomial Time Quantum Algorithm for the Computation of the Unit Group of a Number Field," in *Proceedings of the 37th Symposium on the Theory of Computing* (2005) pp. 475–480.

438. T. Schmitt-Manderbach, H. Weier, M. Fürst, R. Ursin, F. Tiefenbacher, T. Scheidl, J. Perdigues, Z. Sodnik, C. Kurtsiefer, J. G. Rarity, A. Zeilinger, and H. Weinfurter, "Experimental Demonstration of Free-Space Decoy-State Quantum Key Distribution over 144 km," *Phys. Rev. Lett.*, Volume **98** (2007) 010504.
439. U. Schöning, "A Probabilistic Algorithm for k -SAT and Constraint Satisfaction Problems," in *40th Annual Symposium on Foundations of Computer Science*, IEEE Press, New York (1999) pp. 17–19.
440. B. Schumacher and R. F. Werner, "Reversible Quantum Cellular Automata," [arXiv: quant-ph/0405174](https://arxiv.org/abs/quant-ph/0405174), May (2004).
441. "Secrets, Lies, and Atomic Spies," NOVA PBS television series, aired 5th February (2002).
442. See "National Information Assurance Policy for Space Systems used to Support National Security Missions," Committee on National Security Systems, CNSS Policy No. 12, 20th March (2007) available at <http://www.cnss.gov/Assets/pdf/CNSSP-12.pdf>.
443. See PowerPoint presentation on "Vulnerability of Fiber Optic Infrastructure to Intrusion," <http://www.certconf.org/presentations/2003/Tues/TG2.pdf>.
444. See "RSA-200 is Factored!" <http://www.rsa.com/rsalabs/node.asp?id=2879>.
445. See "RSA-576 is Factored!" <http://www.rsa.com/rsalabs/node.asp?id=2096>.
446. B. Selman, H. J. Levesque, and D. G. Mitchell, "A New Method for Solving Hard Satisfiability Problems," in *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, AAAI Press/MIT Press, Menlo Park/Cambridge (1992) pp. 440–446.
447. M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, and H. T. Cory, "The British Nationality Act as a Logic Program," *Communications of the ACM*, Volume **29**, Issue 5 (1986) pp. 370–386.
448. A. Shamir, *Factoring Large Numbers with the TWINKLE Device*, Lecture Notes in Computer Science, Volume **1717**, Springer, Berlin (1999) ISBN 978-3-540-66646-2.
449. A. Shamir and E. Tromer, "Factoring Large Numbers with the TWIRL Device," in *Proc. Crypto 2003*, Lecture Note in Computer Science, Volume **2729**, Springer, Berlin (2003) pp. 1–26.
450. S. Shapiro (ed.), *Church's Thesis*, Encyclopedia of Artificial Intelligence, Wiley, New York (1990) pp. 99–100.
451. V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, "Synthesis of Reversible Logic Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems*, Volume **22**, Issue 6 (2003) pp. 710–722.
452. V. V. Shende, I. L. Markov, and S. S. Bullock, "Minimal Universal Two-qubit Controlled-NOT-based Circuits," *Phys. Rev. A*, Volume **69** (2004) 062321.
453. V. V. Shende, S. S. Bullock, and I. L. Markov, "Synthesis of Quantum Logic Circuits," *IEEE Trans. on Computer-Aided Design*, Volume **25**, Issue 6 (2006) pp. 1000–1010.
454. J. Sherson, H. Krauter, R. Olsson, B. Julsgaard, K. Hammerer, I. Cirac, and E. Polzik, "Quantum Teleportation between Light and Matter," *Nature*, Volume **443** (2006) pp. 557–560.
455. P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," in *Proc. of the 35th Annual Symposium on Foundations of Computer Science*, ed. S. Goldwasser, IEEE Computer Society, New York (1994) pp. 124–134.
456. P. Shor, "Scheme for Reducing Decoherence in Quantum Computer Memory," *Phys. Rev. A*, Volume **52** (1995) pp. R2493–R2496.
457. P. Shor, "Fault-tolerant Quantum Computation," in *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society, Los Alamitos (1996) pp. 56–65.
458. P. Shor, "Polynomial-time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Journal on Computing*, Volume **26**, Issue 5 (1997) pp. 1484–1509.
459. Ch. Silberhorn, T. C. Ralph, N. Lütkenhaus, and G. Leuchs, "Continuous Variable Quantum Cryptography: Beating the 3 dB Loss Limit," *Phys. Rev. Lett.*, Volume **89** (2002) 167901.
460. R. Silverman, "The Multiple Polynomial Quadratic Sieve," *Mathematical Computing*, Volume **48** (1987) pp. 329–339.

461. M. S. Siu, "From Quantum Circuits to Adiabatic Algorithms," Phys. Rev. A, Volume **71** (2005) 062314.
462. Smart Quantum (France), <http://www.smartquantum.com>.
463. D. Solenov and L. Fedichkin, "Nonunitary Quantum Walks on Hyper-Cycles," Phys. Rev. A, Volume **73** (2006) 012308.
464. R. Somma, G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme, "Simulating Physical Phenomena by Quantum Networks," Phys. Rev. A, Volume **65** (2002) 042323.
465. L. Song and C. P. Williams, "Computational Synthesis of Any n -Qubit Pure or Mixed State," in *Proceedings of SPIE*, Volume **5105**, Aerosense, 21st–22nd April 2003, Orlando, Florida, April (2003), pp. 195–203.
466. A. T. Sornborger and E. D. Stewart, "Higher-Order Methods for Quantum Simulations," <arXiv:quant-ph/9809009v1> (1998).
467. A. T. Sornborger and E. D. Stewart, "Higher-Order Methods for Simulations on Quantum Computers," Phys. Rev. A, Volume **60**, Issue 3 (1999) pp. 1956–1965.
468. Spec sheet "Cerberis: the best of classical and quantum worlds. Symmetric encryption and quantum key distribution," <http://www.idquantique.com/products/files/Cerberis-specs.pdf>.
469. Spec sheet "MagiQ QPN 8505 Security Gateway: Uncompromising VPN Security," http://www.magiqtech.com/MagiQ/Products_files/8505_Data_Sheet.pdf.
470. L. Spector and H. J. Bernstein, "Communication Capacities of Some Quantum Gates, Discovered in Part through Genetic Programming," in *Proceedings of the Sixth International Conference on Quantum Communication, Measurement, and Computing (QCMC)*, eds. J. H. Shapiro and O. Hirota, Rinton Press, Princeton (2003) pp. 500–503.
471. L. Spector and J. Klein, "Machine Invention of Quantum Computing Circuits by Means of Genetic Programming," AI-EDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Volume **22**, Issue 3 (2008) pp. 275–283.
472. L. Spector, H. Barnum, and H. J. Bernstein, "Genetic Programming for Quantum Computers," in *Genetic Programming 1998: Proceedings of the Third Annual Conference*, eds. J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, and R. L. Riolo, Morgan Kaufmann, San Francisco (1998) pp. 365–374.
473. L. Spector, H. Barnum, and H. J. Bernstein, "Quantum Computing Applications of Genetic Programming," in *Advances in Genetic Programming*, Volume **3**, eds. L. Spector, U. O'Reilly, W. Langdon, and P. Angeline, MIT Press, Cambridge (1999) pp. 135–160.
474. L. Spector, H. Barnum, H. J. Bernstein, and N. Swamy, "Finding a Better-than-Classical Quantum AND/OR Algorithm using Genetic Programming," in *Proceedings of the 1999 Congress on Evolutionary Computation*, Piscataway, NJ, IEEE Press, New York (1999) pp. 2239–2246.
475. L. Spector, H. Barnum, H. J. Bernstein, and N. Swamy, *Quantum Computing Applications of Genetic Programming*, Complex Adaptive Systems Series, Advances in Genetic Programming, Volume **3**, MIT Press, Cambridge (1999) pp. 135–160.
476. F. M. Spedalieri, "Quantum Key Distribution Without Reference Frame Alignment: Exploiting Photon Orbital Angular Momentum," Optics Communications, Volume **260**, Issue 1 (2006) pp. 340–346.
477. A. M. Steane, "Multiple Particle Interference and Quantum Error Correction," Proc. R. Soc. Lond. A, Volume **452** (1996) pp. 2551–2577.
478. A. M. Steane, "Error Correcting Codes in Quantum Theory," Phys. Rev. Lett., Volume **77** (1996) pp. 793–797.
479. A. M. Stephens, A. G. Fowler, and L. C. L. Hollenberg, "Universal Fault-Tolerant Quantum Computation on Bilinear Nearest Neighbor Arrays," Quantum Information and Computation, Volume **8**, Issue 3 & 4 (2008) pp. 330–344.
480. S. Stepney and J. A. Clark, "Searching for Quantum Programs and Quantum Protocols: a Review," J. Comput. Theor. Nanosci., Volume **5** (2008) pp. 942–969.
481. A. Stern and B. I. Halperin, "Proposed Experiments to Probe the Non-Abelian $v = 5/2$ Quantum Hall State," Phys. Rev. Lett., Volume **96** (2006) 016802.

482. Z.-K. Su, F.-Q. Wang, R.-B. Jin, R.-S. Liang, and S.-H. Liu, “A Simple Scheme for Quantum Networks Based on Orbital Angular Momentum States of Photons,” *Optics Communications*, Volume **281**, Issue 19 (2008) pp. 5063–5066.
483. T. Sugimoto and K. Yamazaki, “A Study on Secret Key Reconciliation Protocol ‘CAS-CADE’,” *IEICE Trans. Fundamentals*, Volume **E83-A**, Issue 10 (2000).
484. K. Svore, D. DiVincenzo, and B. Terhal, “Noise Threshold for a Fault-Tolerant Two-Dimensional Lattice Architecture,” *Quantum Information and Computation (QIC)*, Volume **7**, Issue 4 (2007) pp. 297–318.
485. T. Szkopek, V. Roychowdhury, E. Yablonovitch, and D. S. Abrams, “Eigenvalue Estimation of Differential Operators,” *Phys. Rev. A*, Volume **72** (2005) 062318.
486. T. Szkopek, P. O. Boykin, H. Fan, V. P. Roychowdhury, E. Yablonovitch, G. Simms, M. Gyure, and B. Fong, “Threshold Error Penalty for Fault-Tolerant Quantum Computation with Nearest Neighbor Communication,” *IEEE Transactions on Nanotechnology*, Volume **5**, Issue 1 (2006) pp. 42–49.
487. M. Tegmark and J. A. Wheeler, “100 Years of the Quantum,” [arXiv:quant-ph/0101077v1](https://arxiv.org/abs/quant-ph/0101077v1) (2001).
488. H. Terashima and M. Ueda, “Nonunitary quantum circuit,” *Int. J. Quantum Inform.* Volume **3** (2005) pp. 633–647.
489. “The Elliptic Curve Cryptosystem: Remarks on the Security of the Elliptic Curve Cryptosystem,” a Certicom Whitepaper, July (2000) available at <http://www.comms.scitech.susx.ac.uk/ft/crypt/EccWhite3.pdf>.
490. “The Story of Captain Midnight,” Available online at <http://web.archive.org/web/20070128101239/http://www.signaltonoise.net/library/captmidn.htm>.
491. “The TWIRL Integer Factorization Device.” <http://people.csail.mit.edu/tromer/twirl/>—active as of July (2009).
492. “Threats to Fiber Optic Infrastructures,” Opterna, 1st–2nd October (2003) available at <http://www.blackhat.com/presentations/bh-federal-03/bh-fed-03-gross-up.pdf>.
493. W. Tittel, J. Brendel, H. Zbinden, and N. Gisin, “Violation of Bell Inequalities by Photons More Than 10 km Apart,” *Phys. Rev. Lett.*, Volume **81** (1998) pp. 3563–3566.
494. T. Toffoli, “Reversible Computing,” in *Proceedings of Automata, Languages and Programming, Seventh Colloquium*, ed. de Bakker, Springer, Berlin (1980) pp. 632–644.
495. B. F. Toner and D. Bacon, “Communication Cost of Simulating Bell Correlations,” *Phys. Rev. Lett.*, Volume **91** (2003) 187904.
496. J. F. Traub and H. Wozniakowski, “Path Integration on a Quantum Computer,” *Quantum Information Processing*, Volume **1**, Issue 5 (2002) pp. 365–388.
497. D. C. Tsui, H. L. Stormer, and A. C. Gossard, “Two-Dimensional Magnetotransport in the Extreme Quantum Limit,” *Phys. Rev. Lett.*, Volume **48**, Issue 22 (1982) pp. 1559–1562.
498. R. R. Tucci, “A Rudimentary Quantum Compiler, Second Edition,” [arXiv:quant-ph/9902062v1](https://arxiv.org/abs/quant-ph/9902062v1) (1999).
499. R. R. Tucci, “Qubiter Algorithm Modification, Expressing Unstructured Unitary Matrices with Fewer CNOTs,” [arXiv:quant-ph/0411027v1](https://arxiv.org/abs/quant-ph/0411027v1) (2004).
500. L. Turin, “A Spectroscopic Mechanism for Primary Olfactory Reception,” *Chemical Senses*, Volume **21**, Issue 6 (1996) pp. 773–791.
501. A. Turing, “On Computable Numbers with an Application to the Entscheidungsproblem,” *Proceedings of the London Mathematical Society*, Volume **42** (1937) pp. 230–265; Erratum in Volume **43** (1937) pp. 544–546.
502. W. Unruh, “Maintaining Coherence in Quantum Computers,” *Phys. Rev. A*, Volume **51** (1995) pp. 992–997.
503. R. Ursin, T. Jennewein, M. Aspelmeyer, R. Kaltenbaek, M. Lindenthal, P. Walther, and A. Zeilinger, “Quantum Teleportation Link across the Danube,” *Nature*, Volume **430** (2004) p. 849.
504. R. Ursin, F. Tiefenbacher, T. Schmitt-Manderbach, H. Weier, T. Scheidl, M. Lindenthal, B. Blauensteiner, T. Jennewein, J. Perdigues, P. Trojek, B. Ömer, M. Fürst, M. Meyenburg, J. Rarity, Z. Sodnik, C. Barbieri, H. Weinfurter, and A. Zeilinger, “Entanglement based quantum communication over 144 km,” *Nature Physics*, Volume **3** (2007) pp. 481–486.

505. R. Ursin, T. Jennewein, J. Kofler, J. M. Perdigues, L. Cacciapuoti, C. J. de Matos, M. Aspelmeyer, A. Valencia, T. Scheidl, A. Acin, C. Barbieri, G. Bianco, S. Cova, D. Giggenbach, W. Leeb, R. H. Hadfield, R. Laflamme, N. Lütkenhaus, G. Milburn, M. Peev, T. Ralph, J. G. Rarity, R. Renner, N. Solomos, W. Tittel, J. P. Torres, M. Toyoshima, P. Villoresi, I. Walmsley, G. Weihs, H. Weinfurter, M. Zukowski, and A. Zeilinger, "Space-QUEST. Experiments with Quantum Entanglement in Space," in *Proceedings of the 2008 Microgravity Sciences and Process Symposium* (2008).
506. W. van Dam, M. Mosca, and U. Vazirani, "How Powerful is Adiabatic Quantum Computation?", in *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science* (2001) pp. 279–287.
507. L. G. Valiant, "The Complexity of Computing the Permanent," *Theoretical Computer Science*, Volume **8** (1979) pp. 189–201.
508. W. van Dam, "Quantum Cellular Automata," Master's thesis, University of Nijmegen (1996).
509. W. van Dam, M. Mosca, and U. Vazirani, "How Powerful is Adiabatic Quantum Computation?", in *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science* (2001) pp. 279–287.
510. W. van Dam, S. Hallgren, and L. Ip, "Quantum Algorithms for Some Hidden Shift Problems," *SIAM Journal on Computing*, Volume **36**, Issue 3 (2006) pp. 763–778.
511. J. Vartiainen, M. Möttönen, and M. Salomaa, "Efficient Decomposition of Quantum Gates," *Phys. Rev. Lett.*, Volume **92** (2004) 177902.
512. F. Vatan and C. P. Williams, "Optimal Quantum Circuits for General Two-qubit Gates," *Phys. Rev. A*, Volume **69** (2004) 032315.
513. V. Vedral, "High Temperature Macroscopic Entanglement," *New J. Phys.*, Volume **6** (2004) p. 102.
514. F. Verstraete, J. I. Cirac, and J. I. Latorre, "Quantum Circuits for Strongly Correlated Quantum Systems," *Phys. Rev. A*, Volume **79** (2009) 032316.
515. T. Vértesi and E. Bene, "Lower Bound on the Communication Cost of Simulating Bipartite Quantum Correlations," [arXiv:0904.1390v2](https://arxiv.org/abs/0904.1390v2) (2009).
516. D. Verton, "Intelligence Ops in Baghdad Show Need for Physical Security Back Home," *Computerworld*, 8th April (2003).
517. G. Vidal and C. M. Dawson, "Universal Quantum Circuit for Two-qubit Transformations with Three Controlled-NOT Gates," *Phys. Rev. A*, Volume **69** (2004) 010301(R).
518. Video showing eight glass spheres executing the Scottish Split-the-Willow dance driven by OAM states of light <http://www.physics.gla.ac.uk/Optics/play/StripTheWillow/StripTheWillowBIG.mp4>.
519. P. Villoresi, T. Jennewein, F. Tamburini, M. Aspelmeyer, C. Bonato, R. Ursin, C. Pernechele, V. Luceri, G. Bianco, A. Zeilinger, and C. Barbieri, "Experimental Verification of the Feasibility of a Quantum Channel Between Space and Earth," *New J. Phys.*, Volume **10** (2008) 033038.
520. L. Viola, E. Knill, and S. Lloyd, "Dynamical Generation of Noiseless Quantum Subsystems," *Phys. Rev. Lett.*, Volume **85**, Issue 16 (2000) pp. 3520–3523.
521. J. Von Neumann and A. W. Burks, *Theory of Self-reproducing Automata*, University of Illinois Press, Urbana (1966).
522. H. Wang, S. Kais, A. Aspuru-Guzik, and M. Hoffmann, "Quantum Algorithm for Obtaining the Spectrum of Molecular Systems," *Physical Chemistry Chemical Physics*, Volume **10** (2008) pp. 5388–5393.
523. J. Watrous, "On One-dimensional Quantum Cellular Automata," in *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, October (1995) pp. 528–537.
524. C. Weedbrook, A. M. Lance, W. P. Bowen, T. Symul, T. C. Ralph, and P. K. Lam, "Quantum Cryptography Without Switching," *Phys. Rev. Lett.*, Volume **93** (2004) 170504.
525. T. Wei, K. Nemoto, P. M. Goldbart, P. Kwiat, W. Munro, and F. Verstraete, "Maximal Entanglement versus Entropy for Mixed Quantum States," *Phys. Rev. A*, Volume **67** (2003) 022110.

526. G. Weihs, T. Jennewein, C. Simon, H. Weinfurter, and A. Zeilinger, “A. Violation of Bell’s Inequality under Strict Einstein Locality Conditions,” Phys. Rev. Lett., Volume **81** (1998) pp. 5039–5043.
527. E. W. Weisstein, “RSA-640 Factored,” MathWorld Headline News, 8th November (2005), <http://mathworld.wolfram.com/news/2005-11-08/rsa-640/>. See also the closure of the factoring challenge by RSA Laboratories at <http://www.rsa.com/rsalabs/node.asp?id=2092>.
528. D. Welsh, *Codes and Cryptography*, Oxford Science Publications/Clarendon, Oxford (1988), ISBN 0-19-853287-3.
529. C. Wiechers, L. Lydersen, C. Wittmann, D. Elser, J. Skaar, C. Marquardt, V. Makarov, and G. Leuchs, “After-Gate Attack on a Quantum Cryptosystem,” [arXiv:1009.2683](https://arxiv.org/abs/1009.2683) [quant-ph] (2010).
530. M. Wiener and R. Zuccherato, “Faster Attacks on Elliptic Curve Cryptosystems,” in *Selected Areas in Cryptography*, Lecture Notes on Computer Science, Volume **1556**, Springer, Berlin (1999) pp. 190–200.
531. S. Wiesner, “Conjugate Coding,” Special Issue on Cryptography, ACM SIGACT News, Volume **15**, Issue 1 (1983) pp. 78–88.
532. S. Wiesner, “Simulations of Many-Body Quantum Systems by a Quantum Computer,” [arXiv:quant-ph/9603028v1](https://arxiv.org/abs/quant-ph/9603028v1) (1996).
533. F. Wilczek, *From Electronics to Anyonics*, Physics World (2006) pp. 22–23.
534. P. Willan, “E.U. Seeks Quantum Cryptography Response to Echelon,” Source www.security.itworld.com, 17th May (2004).
535. C. P. Williams, “Probabilistic Non-unitary Quantum Computing,” in *Quantum Information and Computation II*, eds. E. Donkor, A. R. Pirich, and H. E. Brandt, SPIE Proceedings, Volume **5436** (2004) pp. 297–306.
536. C. P. Williams and A. Gray, Automated Design of Quantum Circuits,” in *First NASA International Conference on Quantum Computing and Quantum Communications*, Palm Springs, California, USA, February 17th–20th 1998, Lecture Notes in Computer Science, Volume **1509**, Springer, Berlin (1999) pp. 113–125.
537. C. P. Williams and T. Hogg, “Using Deep Structure to Locate Hard Problems,” in *Proc. 10th National Conf. on Artificial Intelligence (AAAI'92)*, AAAI Press, Menlo Park (1992) pp. 472–477.
538. C. P. Williams and T. Hogg, “Extending Deep Structure,” in *Proc. 11th National Conf. on Artificial Intelligence (AAAI'93)*, AAAI Press, Menlo Park (1993) pp. 152–157.
539. C. P. Williams and T. Hogg, “Expected Gains from Parallelizing Constraint Solving for Hard Problems,” in *Proc. 12th National Conf. on Artificial Intelligence (AAAI'94)*, AAAI Press, Menlo Park (1994) pp. 1310–1315.
540. C. P. Williams and T. Hogg, “Exploiting the Deep Structure of Constraint Problems,” Artificial Intelligence, Volume **70** (1994) pp. 73–117.
541. R. Wilson, *Four Colors Suffice: How the Map Problem Was Solved*, Princeton University Press, Princeton (2003) ISBN 0-691-11533-8.
542. T. Winograd, *Understanding Natural Language*, Academic Press, New York (1972).
543. P. Wocjan and J. Yard, “The Jones Polynomial: Quantum Algorithms and Applications in Quantum Complexity Theory,” [arXiv:quant-ph/0603069](https://arxiv.org/abs/quant-ph/0603069) (2006).
544. S. Wolfram, *A New Kind of Science*, Wolfram Media, Champaign (2002) ISBN: 1-57955-008-8.
545. J. Wood, “Banking on Quantum Cryptography: Technology,” Materials Today, Volume **8**, Issue 7 (2005) p. 23.
546. J. Woolsey, Remarks at the Foreign Press Center, Transcript, 3rd July (2000), <http://cryptome.org/echelon-cia.htm>.
547. W. K. Wootters and W. H. Zurek, “A Single Quantum Cannot be Cloned,” Nature, Volume **299** (1982) pp. 802–803.
548. J. S. Xia, W. Pan, C. L. Vicente, E. D. Adams, N. S. Sullivan, H. L. Stormer, D. C. Tsui, L. N. Pfeiffer, K. W. Baldwin, and K. W. West, “Electron Correlation in the Second Landau Level: A Competition Between Many Nearly Degenerate Quantum Phases,” Phys. Rev. Lett., Volume **93** (2004) 176809.

549. F. Xu, B. Qi, H.-K. Lo, "Experimental Demonstration of Phase-Remapping Attack in a Practical Quantum Key Distribution System," [arXiv:1005.2376v1](https://arxiv.org/abs/1005.2376v1) [quant-ph] (2010).
550. A. Yao, "Quantum Circuit Complexity," in *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society, Los Alamitos (1993) pp. 352–360.
551. X. X. Yi, C. S. Yu, L. Zhou, and H. S. Song, "Noise-Assisted Preparation of Entangled Atoms," *Phys. Rev. A*, Volume **68**, Issue 5 (2003) 052304.
552. A. Yimsiriwattana and S. J. Lomonaco Jr., *Generalized GHZ States and Distributed Quantum Computing*, CONM/381, American Mathematical Society, Providence (2005) pp. 131–147.
553. A. Yimsiriwattana and S. J. Lomonaco, "Distributed Quantum Computing: a Distributed Shor Algorithm," in *Quantum Information and Computation II*, eds. E. Donkor, A. R. Pirich, H. E. Brandt, SPIE Proceedings, Volume **5436** (2004) pp. 360–372.
554. S. Yu, Q. Chen, C. H. Lai, and C. H. Oh, "Nonadditive Quantum Error-Correcting Code," *Phys. Rev. Lett.*, Volume **101** (2008) 090501.
555. Z. S. Yuan, Y. A. Chen, B. Zhao, S. Chen, J. Schmiedmayer, and J. W. Pan, "Experimental Demonstration of a BDCZ Quantum Repeater Node," *Nature*, Volume **454** (2008) pp. 1098–1101.
556. C. Zalka, "Simulating Quantum Systems on a Quantum Computer," *Proc. R. Soc. Lond. A*, Volume **454**, Issue 1969 (1998) pp. 313–322.
557. C. Zalka, "Grover's Quantum Searching Algorithm is Optimal," *Phys. Rev. A*, Volume **60**, Issue 4 (1999) pp. 2746–2751.
558. C. Zalka, "Using Grover's Quantum Algorithm for Searching Actual Databases," *Phys. Rev. A*, Volume **62** (2000) 052305.
559. P. Zanardi, C. Zalka, and L. Faoro, "Entangling Power of Quantum Evolutions," *Phys. Rev. A*, Volume **62** (2000) 30301(R).
560. A. Zeilinger, "Anton Zeilinger: From Quantum Puzzles to Quantum Information Technology," Source: <http://physicsnewsandpress.blogspot.com/2009/07/anton-zeilinger-from-quantum-puzzles-to.html>.
561. A. Zeilinger in an interview originally in German in Die Weltwoche on 3rd January (2006). Source <http://www.signandsight.com/features/614.html>.
562. J. Zhang, J. Vala, S. Sastry, and K. B. Whaley, "Geometric Theory of Nonlocal Two-qubit Operations," *Phys. Rev. A*, Volume **67** (2003) 042313.
563. Q. Zhang, A. Goebel, C. Wagenknecht, Y.-A. Chen, B. Zhao, T. Yang, A. Mair, J. Schmiedmayer, and J.-W. Pan, "Experimental Quantum Teleportation of a Two-qubit Composite System," *Nature Physics*, Volume **2** (2006) pp. 678–682.
564. R. Zoglin and J. Cramer, "Grounding Captain Midnight," *Time*, Monday, Aug. 04, (1986). Available online at <http://www.time.com/time/magazine/article/0,9171,961911,00.html>.
565. M. Zukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert, "Event Ready Detectors: Bell Experiment via Entanglement Swapping," *Phys. Rev. Lett.*, Volume **71** (1993) pp. 4287–4290.
566. W. H. Zurek and R. Laflamme, "Quantum Logical Operations on Encoded Qubits," *Phys. Rev. Lett.*, Volume **77** (1996) pp. 4683–4686.
567. K. Zyczkowski, P. Horodecki, A. Sanpera, and M. Lewenstein, "Volume of the Set of Separable States," *Phys. Rev. A*, Volume **58**, Issue 2 (1998) pp. 883–892.

Index

- $\llbracket 5, 6, 2 \rrbracket$ code, 611
 $\llbracket 9, 12, 3 \rrbracket$ non-additive code, 611
 $\llbracket n, K, d \rrbracket$ code, 586
 $D_{2^n}^{(4)}$ quantum wavelet kernel, 154
 circuit, 158
 factorization, 157
 $\text{GF}(2^m)$, 287
 $\text{GF}(p)$, 287
 $H(X)$, 409
 $I(X : Y)$, 409
 P_{2^n} qubit reversal, 131
 Q_{2^n} amplitude downshift, 137
 R_x , 76
 R_y , 76
 $S_L(\rho)$
 as measure of mixedness, 420
 $S_V(\rho)$
 as measure of mixedness, 421
 XY -Hamiltonian, 341, 342
 XY interaction Hamiltonian, 72
 $\llbracket n, k, d \rrbracket$ code, 607
 $\llbracket n, k, d \rrbracket$ stabilizer code, 604
 $\llbracket 5, 1, 3 \rrbracket$ stabilizer code, 611
 $\llbracket 5, 1, 3 \rrbracket$ code, 605
 $\llbracket 7, 1, 3 \rrbracket$ code, 605
 $\llbracket 9, 1, 3 \rrbracket$ code, 605
 $[\Delta A, \Delta B]$ commutator, 485
 ΔA , operator deviation, 484
 Ω , big omega notation, 225, 226
 Π_{2^n} qubit cyclic left shift, 135
 $\Theta(\cdot)$, big theta notation, 225, 226
 \hbar , Planck constant over 2π , 523
 λ calculus, 207
- $\sqrt{\text{NOT}}$, 72
 \vee , 53
 \wedge , 53
 $\{\Delta A, \Delta B\}$ anti-commutator, 485
 $k\text{-COL}$, 301
 $k\text{-SAT}$, 300
 $k\text{-SAT}, k\text{-satisfiability}$, 222
 $\mathcal{O}(\cdot)$, big “O” notation, 225, 226
BPP complexity class, 226
BQP complexity class, 229
NP, 293, 294
 integer factorization, 273
NP complexity class, 226
NP-Complete, 294
 and naive quantum search, 303
 graph coloring, 295
 hardness, 297
 importance, 295
 polynomial interconversion, 296
 quantum nested search, 302, 308
 satisfiability, 295
 subset sum, 295
 travelling salesman, 295
 ubiquity, 295
NP-Complete complexity class, 226
NP-Hard, 294
NP-Hard complexity class, 226
NP-Intermediate complexity class, 234
QP complexity class, 229
ZPP complexity class, 226
ZQP complexity class, 229
P, 294
5-qubit code, 588
 decoding circuit, 591

- 5-qubit code (*cont.*)
 encoding circuit, 590
- 7-qubit code, 594
- 9-qubit code, 593
- A**
- Abrams, D., 326, 339, 361
- Abrams-Lloyd eigenvalue estimation algorithm, 361
- Adami, C., 303
- Adleman, L., 267
- AES, 520
 re-keying via QKD, 550
- Affinity, 349
- Ajisai satellite, 550
- Algebraic circuit design, 174
 example, 179
 limitation, 185
- Algorithm
 backtracking, 296
 Davis-Putnam-Logemann-Loveland, 296
 generate-and-test, 244
 heuristic, 296
 Lanczos, 351
 Lloyd-Zalka-Wiesner quantum simulation, 327
 number field sieve, 223
 Shor, 272, 273
 worked example, 280
 WalkSAT, 297
- Amdahl's law, 324
- Amplitude amplification, 245–249
 in quantum counting, 372
 multiple solutions, 252
 of selected eigenstate(s), 247
 oracle, 250
- Amplitude downshift permutation, 137
 in controlled-two's complement, 169
- Amplitudes
 cf. probabilities, 212
- Ancilla, 575
 in error correcting code, 583
 in quantum error correction
 measurements determine error, 585
 measure for desired side effect, 576
 measured to induce known error, 592
- Ancilla-assisted readout, 330
 circuit, 330, 331
 using multiple ancillae, 330
 using single ancilla, 328
- AND gate, 53
- AND gate from NAND gates, 56
- Annihilation operator, 340
- Anti-commutator, 485
 of Hermitians is Hermitian, 485
- Anti-symmetric state vector, 335, 336, 338, 339
- Anti-symmetric wavefunction, 350
- Anti-unitary matrix, 471
- Antipodal state, 73
- Anyons, 337
- Application of
 entanglement swapping, 443
- Applications
 quantum search, 255
- Arnesen, M. C., 443
- Arora, S., 206
- Artificial mathematicians, 219
- Aspect, A., 493
- Aspuru-Guzik, A., 351
- Atmospheric channel, 549
- Atmospheric compensation, 550
- Atomic ensemble quantum memories
 in quantum repeaters, 548
- Atoms
 electronic structure, 338
- Atoms per bit, 6
- Attack on Coventry, 266, 267
- Attacks on satellites
 Captain Midnight, 551
 denial of service, 554
 Indonesia, 554
 potential for terrorists, 553
 Tonga, 554
- Audio compression, 162
- Augment-on-fail quantum data compression, 449
- Automata
 classical, 630
 quantum, 631
- Automated circuit design, 172
 algebraic, 174
 applications, 173
 choice of gates, 173
 numerical, 180
 re-use, 184, 187
- Average case complexity, 222
- Axioms, 203

B

- B92 QKD protocol, 539
 potential security holes, 540
- B92 QKD protocol:success-not-assured
 polarization measurement, 539
- Babylonian cuneiform, 264
- Backtracking, 296
- Bad index values, 252
- Bank of England
 interest in QKD, 547
- Bass, A., 494
- BB84 QKD protocol, 529, 531, 532
 long range free space, 550
 using orbital angular momentum states,
 537
- worked example
 with eavesdropping, 536, 537
 without eavesdropping, 534–536
- BB84 QKD protocol: success-assured
 polarization measurement, 532
- BBN quantum network, 547
- BBO crystal, 524
- BDCZ quantum repeater, 548
- Bell, J., 489
- Bell state, 629
 synthesis, 496
- Bell state analyzer, 441, 442
 experimental implementation, 443
- Bell states, 431, 454, 496
 definition, 454
 interconversion, 454, 455
 quantum circuit for synthesizing, 496
 synthesis, 454
- Bell-basis measurement, 634–636
- Bell's inequality, 489
 application to QKD, 541
 defined, 491
 experimental tests, 492
 thought experiment, 489
 violated, 493
 visual proof of violation, 492
- Belt trick, 13
- Benioff, P., 210, 230
- Bennett, C., 210, 436, 486, 529, 539, 545
- Bernstein, E., 213, 233
- Bernstein, H., 436
- Berthiaume, A., 230, 233
- Beta barium borate crystal, 524
- Beth, T., 611
- Beyond Shannon compression, 453
 at communication time, 453
- Binary fraction, 146, 353
- Birefringent crystal, 528, 532
- Bit flip error, 568, 569, 583, 590
 cause, 579
 undoing, 593
- Black-box function, 243, 244, 250
- Bletchley Park, 265, 266
- Bloch ball, 415
 quantum clones on, 462
- Bloch sphere, 11, 12, 73
 cf. Bloch ball, 415
 orthogonal states, 12
 visualize states in BB84 QKD protocol,
 531
- Boeing 777
 majority voting, 573
- Bogoliubov transformation, 343
- Boltzmann's constant, 570
- Bond angles, 349
- Bose, S., 443
- Bose-Einstein statistics, 337
- Bosons, 336, 523
 composite particles, 336
 under particle interchange, 336
- Bound
 Nyquist, 144
 quantum Gilbert-Varshamov, 607
 quantum Hamming, 606, 607
 quantum Singleton, 606, 607
- Bouwmeester, D., 500
- Bra vector, 10
- Branciard, C., 494
- Brassard, G., 230, 233, 252, 372, 486, 529,
 545
- Breaking ECC, 513
- Breaking RSA, 513
- Briegel, H., 548
- British intelligence, 266
 role in inventing public-key
 cryptosystem, 267
- British Nationality Act
 as logic program, 219
 logical inconsistencies, 219
- British Post Office, 266
- Broadcasting quantum information, 470
- Brukner, C., 443
- Bruss, D., 538

- Bruss' 6-State QKD protocol, 538
 enhances bit error rate of eavesdropper, 539
- Bucky-ball molecules
 interference of, 502
- Bugging devices, 510
- Buggy state
 in error reduction by symmetrization, 575
- Bullock, S., 130, 174
- Bužek, V., 460
- C**
- Caesar cipher, 264
- Calcite crystal, 528
 birefringence, 528, 529
 double image, 528
- Calcite crystals, 532
- Calderbank, R., 607
- Calderbank-Shor-Steane code, 605
- California Institute of Technology, 615
- Caltech, 615
- Canary Islands
 in free space QKD experiment, 550
- Cantor diagonal slash, 216
- Captain Midnight, 551
 detective work, 553
 identity revealed, 551
 message, 552
 motive for attack, 551
- CASCADE protocol for error reconciliation, 544
- Cattle problem of Archimedes, 375
- Cauchy-Schwarz inequality
 in Heisenberg Uncertainty Principle, 484
- Causes of errors, 568
 decoherence, 569
 dissipation, 568
 unwanted entanglement, 569
- Cavity QED based quantum repeater, 548
 resurgence, 548
- CCSD(T), 350
- Cellular automata, 630
- Central Florida Teleport, 551
- Central Intelligence Agency, 509
- Certicom, 287
- Change of basis, 32
- Channel
 memoryless, 409
- Channel coding theorem, 403, 408
- Characteristic polynomial, 465
- Chebyshev polynomial
 first kind, 249
 second kind, 249
- Cheeseman, P., 222, 316
- Chen, Q., 611
- Chen, Y.-A., 501
- Cheung, D., 287, 632, 633
- Chromatic number, 299
- Chuang, I., 634, 635
- Church, A., 207, 208
- Church-Turing thesis, 208
 cf. Deutsch's thesis, 209
 quantum challenge to, 209
- Churchill, W., 266
- Cirac, I., 342, 501, 548
- Circuit complexity
 arbitrary unitary, 173
- Circuit design
 algebraic, 174
 numerical, 180
 re-use, 184, 187
- Circuit rewrite rules, 178
- Circularly polarized light, 522
- Circularly polarized photon, 523
- Classical cellular automata, 630
- Classical complexity classes, 225
- Classical entropy, 405, 407
- Classical error correction, 567
 majority voting, 572
 why it is easy, 571
- Classical information, 404
 Shannon view, 404
- Classical information cf. quantum, 572
- Classical (n, K, d) code, 587
- Classical NOT cf. quantum NOT, 470
- Classical simulation
 ab initio, 321
 density functional theory, 323
 full configuration interaction, 322
 Hartree-Fock, 323
 inefficient for quantum systems, 326
 of classical systems, 326
 of quantum systems, 326
 problem of entanglement, 321
 problem of fidelity, 322
 problem of memory, 321
 quantum systems, 320
 tight-binding, 322

- Classical-to-quantum encoding, 382
Classically controlled gates, 636, 639
Clauser, J., 493
Clip on coupler, 511
Clique number, 299
Cloning quantum information, 457
CNOT gate, 58
 distributed, 629, 630
 teleported, 638
Code
 degenerate, 587
 earliest, 264
 imperfect, 588
 impure, 587
 minimal distance of a, 586
 non-degenerate, 587
 perfect, 588
 pure, 587
Code-breaking, 263, 264
 American, 266
 British, 265, 266
 elliptic curve cryptosystems, 285
 Enigma, 265
 RSA cryptosystems, 280
Code-making, 264
 Babylonian, 264
 Greek, 264
 Roman, 264
Codespace dimension, 611
Codeword, 579, 605
 quantum, 580
 stabilizer code, 597
Coherence loss, 568
Colossus computer, 266
Command and control of satellites
 concerns, 553
Commercial quantum cryptography
 systems, 554, 555
Commonsense intuitions
 confounded by logic, 218
Communication complexity, 221
Communications
 reliable, 408
Commutator, 485
 of Hermitians is anti-Hermitian, 485
Compactification, 174, 178
Complex number, 212
Complexity, 201, 202, 213
 average case, 222, 312
 based on cost scaling not absolute cost,
 222
circuit, 230
communication, 221
computational, 221
multiplication cf. factoring, 223
physics-inspired view of, 316
quantum structured search, 312
query, 221
size of a number, 223
subexponential, 223
superpolynomial, 223
worst case, 222, 296
Complexity class
 NP, 294
 NP-Complete, 294
 NP-Hard, 294
 P, 294
Complexity classes
 classical, 225
 poor naming convention, 201
 quantum, 229
Compressibility
 of strings, 408
Compressing information
 quantum, 444
Compression
 message blocks, 447
 quantum, 445
Compressor, 447
 in terms of *Q* and TOFFOLI, 451
 in terms of QFT, 451
 unitary matrix, 450
Computability, 201, 202, 213
 classical matches quantum, 215
 quantum, 214
Computational basis, 16, 17
 eigenstate, 18
Computational complexity, 221
Computational correctness, 206
Computational intractability
 use of, 267
Computational material design, 324
Computational models, 202
 equivalence, 208
Computational phase transition, 297, 299,
 303, 316
Computational phase transitions, 201
Computational universality, 208
Computer science
 foundations, 202

- Concatenated codes, 617, 619, 620
 Concurrence, 433
 Conditional entropy, 409
 Conditional gate, 128
 in error reduction via symmetrization, 575
 Conjugate coding, 529
 Conjunctive normal form, 301
 Constrainedness, 300
 Constraint satisfaction problems, 305
 Constructive interference, 20
 Controlled modular-add-one (C-MAO), 169
 Controlled SWAP gate, 577
 Controlled-add-one gate, 438
 Controlled-controlled-NOT gate, 61
 Controlled-modular add one (C-MAO)
 in quantum cosine transform, 170
 Controlled-one's complement (C-OC), 168
 in quantum cosine transform, 170
 Controlled-SWAP gate, 61
 Controlled-two's complement (C-TC), 169
 in quantum cosine transform, 170
 Conversion
 quantum circuit to unitary matrix, 124
 unitary matrix to quantum circuit, 172
 Cooper pairs, 336
 Copying quantum information, 457
 Corrective action
 5-qubit code, 591
 Correctness, 206
 Correlation functions, 333
 Cosine series, 163
 Coupling to environment, 568
 ambient thermal heat bath, 568
 stray particles, 568
 why hard to avoid, 568
 Coventry attacked, 266, 267
 Cover, T., 410
 Crandall, R., 224
 Creation operator, 340
 Crepeau, C., 486
 Critical ratio
 clauses to variables, 300, 301
 Critical temperature, 298
 Crossed polarizers, 529
 Cryptographic security
 unconditional, 508
 Cryptography, 264
 ethical and legal considerations, 508
 Cryptosystem
 Diffie-Helman, 267
 OTP, one time pad, 515
 public key, 267
 RSA, 267, 268
 unconditionally secure, 515
 Cryptosystems
 approved for “top secret”, 513
 CSS code, 594, 605
- D**
- Dancing glass spheres, 537
 DARPA quantum network, 547
 Daubechies, I., 153
 Daubechies wavelets, 154
 David-Putnam-Logemann-Loveland algorithm, 296
 DCT, 162
 Decidability, 215
 Decision making
 reliable, 573
 Decision problem, 293
 Decoherence, 568, 569
 affect on density operator, 569
 appears environment measures qubit, 569
 appears to preclude arbitrarily long quantum computations, 571
 de-phasing, 569
 induces phase shift, 569
 max steps before succumbing, 570
 mitigation, 570
 pure state becomes mixed, 569
 simplified model, 569
 time scale, 569
 undoing its affects, 571
 Decoherence time
 at temperature of liquid helium, 570
 cf. dissipation, 570
 defined, 569
 estimated from Heisenberg Uncertainty Principle, 570
 examples, 570
 of different physical systems, 570
 pressure dependence, 570
 temperature dependence, 570
 typical at room temperature, 570
 Decoherence times
 factors affecting, 570
 Decompressor, 448, 449

- Decoy pulses, 547
Definition
 density operator, 412
 entangled state, 423
 entanglement concentration, 440
 entanglement of distillation, 430
 entanglement of formation, 430
 linear entropy, 420
 negativity, 430
 partial transpose, 425
 relative entropy of entanglement, 430
 Schmidt decomposition, 433
 separable state, 423
Degenerate code, 587, 588
 efficiency cf. non-degenerate codes, 588
Dense coding, 453
Density functional theory, 323
Density matrix
 transformed non-unitarily, 191
Density operator, 411
 and Schrödinger's equation, 416
 as statement of partial ignorance, 417
 as statement of partial knowledge, 411
 definition, 412
 diagonalize, 421
 maximally entangled mixed states, 433
 maximally mixed state, 422
 mixed state, 412
 non-uniqueness of, 416
 properties of, 416
 pure state, 414
Derkacs, D., 550
DES
 vulnerable to quantum attack, 287
Design by re-use, 184
 circuit template, 187
 example (QFFT), 188
 example (QHT), 188
Destructive interference, 20
Detector loophole closed, 494
Determinism, 218
Deterministic evolution, 573
Deterministic Turing machine, 204, 205
 equivalence to modern computers, 205
Deutsch, D., 209, 214, 230, 231
 quantum Turing machine, 211
 wikiquote ranking, 210
Deutsch's thesis, 209
Di Vincenzo, D., 172
Diagonalization, 421
Diagonally polarized photon, 523, 526, 532
Dieks, D., 458
Diffie-Hellman public key cryptosystem, 267
 basis of security, 267
Dimension of codespace, 611
Diophantine equation, 375
Diplomatic communiqüs, 507
Dirac notation, 9, 10
Dirac's belt trick, 13
Direct product, 19, 127, 574, 575
Direct sum, 128
Directional radio beams, 266
Discard-on-fail quantum data compression,
 447
Disco ball in Earth-to-Space experiment,
 550
Discrete cosine transform, 162
 cf. discrete Fourier transform, 163
 DCT-I to DCT-VIII, 163
Discrete logarithm problem, 285
 elliptic curve, 286
 possibility of better classical algorithm,
 514
 possibility of special purpose hardware,
 514
Discrete logarithms, 272
Discrete-logarithm
 in Diffie-Hellman cryptosystem, 267
Disruptions to aircraft communications
 unintentional interference, 554
Disruptions to spacecraft communications
 Captain Midnight, 554
 NASA Space Shuttle, 554
Dissipation, 568
 cf. decoherence time, 570
 induces bit flip, 568
 modelled by Pauli X , 568
Dissipation-assisted quantum computing,
 621
Distance
 between bit strings, 579
 from entangled state to nearest separable
 state, 430
 Hamming, 579
Distinguishable versus not completely
 distinguishable symbols, 421
Distributed CNOT gate, 629, 630
Distributed QFT, 630

- Distributed quantum computer, 628
 Distributed quantum computing, 443
 Distributed Shor's algorithm, 630
 DiVincenzo, D., 570, 627
 DiVincenzo criteria, 627
 DLCZ quantum repeater, 548
 Dog urine, 527
 Dot product, 126
 Dowling, J. P., 263
 Downconversion, 524
 Downshift permutation, 137
 DTM, 231
 cf. QT M, 211
 DTM, deterministic Turing machine, 205
 Duan, L. M., 190
 Duan, L.-M., 469
 Durr, C., 632
 Dürr, W., 548
- E**
- E91 entanglement-based QKD protocol, 541, 542
 long range free space, 550
 novelty, 541
 Earth to Space
 QKD, 557
 quantum key distribution, 548, 557
 Earth-to-Space
 quantum communications with
 International Space Station, 550
 quantum cryptography, 548
 single photon exchange demonstrated, 550
 Earth-to-Space QKD, 551
 Earth-to-Space quantum cryptography, 549, 550
 Eavesdropping
 competitive disadvantages caused by, 510
 detectable, 511
 fiber-optic communications, 511
 on fiber-optic communications, 510
 on satellite communications, 508
 on submarine fiber-optic cables, 511
 undetectable, 511
 with ECHELON, 509
 Ebits, 453
 ECC, 286, 507, 513
- ECHELON electronic surveillance system, 508, 509
 cell phone intercepts, 509
 Effect of logical irreversibility, 57
 Efficiency of simulation, 326
 Efficient codes, 611
 Eibl, M., 500
 Eigenbasis
 in partial transpose, 425
 in Schmidt decomposition, 434
 Eigenvalue estimation
 importance of relative phase, 359
 via Lanczos Algorithm, 351
 Eigenvalue kick-back, 357, 361
 Einstein, A., 488
 Eisert, J., 629
 Ekert, A., 442, 541, 542, 606, 607
 Ekert's entanglement-based QKD protocol
 entangled photon source, 524, 525
 El Gamal encryption scheme, 285
 Electro-mechanical relay based computer, 266
 Electromagnetic wave, 522
 Electron in an atom, 338
 Electron orbital, 338
 Electron shells, 338
 Electron-hole pairs, 336
 Electronic communications
 security, 508
 Electronic structure, 349
 Elliptic curve cryptosystem, 285, 286, 513
 Elliptic curve discrete logarithm problem, 286
 Elliptically polarized light, 522
 Encoding a message as integers, 271
 Encryption, 513
 Energy eigenspectrum, 349
 Energy loss to environment
 dissipation, 568
 Energy loss when information is erased, 57
 Enigma code, 265, 266
 broken by British, 265
 broken by Polish, 265
 intercept on Coventry attack, 266
 Enigma machine, 265
 deemed "unsuitable" for military use by
 British, 265
 patent, 265
 Entangled
 mixed state, 428

- Entangled (*cont.*)
pure state, 427
- Entangled photon source, 524
iconic image, 525
- Entangled state, 423
5-qubit code, 589, 597
7-qubit code, 594
9-qubit code, 593
as sum over direct product of states, 433
in error reduction via symmetrization,
575
- Entangled states
interconversion, 454, 455
- Entanglement, 422
3-way without 2-way, 429
concentration, 436
concentration cf. purification, 436
distillation, 436
in error correction, 573
in quantum repeaters, 548
in superdense coding, 453
in warm bulk matter, 443
inequivalent types of, 431
maximal mixed state, 432
maximal pure state, 431
monotones, 429
of outputs in quantum cloning, 465
persistence of, 443
proof via constraint solving, 423
proof via Peres-Horodecki criterion, 425
proof via witness, 423
purification, 441
quantifying, 429
specially designed, 573
swapping, 441
- Entanglement concentration, 436
- Entanglement distillation, 436
from mixed states, 441
from pure states, 436
- Entanglement monotones, 430
desired properties, 429
- Entanglement of distillation, 430
- Entanglement of formation, 430
- Entanglement purification, 441
in quantum repeaters, 548
mixed states, 548
- Entanglement swapping, 441
applications, 443
circuit, 442
in quantum repeaters, 548
- Entanglement witness, 423, 424
based on mean energy, 424
- Entropy
cf. conditional entropy, 409
cf. joint entropy, 409
cf. mutual information, 409
classical, 405
 H , 407
linear, 420
quantum, 421, 445
Shannon, 407
von Neumann, 421, 445, 446
- Entropy of formation, 433
- Entscheidungsproblem, 202, 203
reduced to halting problem, 216
resolution of, 215, 217
- Environmental coupling, 568
ambient thermal heat bath, 568
stray particles, 568
why hard to avoid, 568
- EPR effect, 488
- EPR pair, 629
- Equally weighted superposition
number of components not power of two,
576
of n factorial states, 576
- Equivalence of computational models, 205,
208
- Error correcting codes, 580
- Error correction, 567
by symmetrization, 573–575
assumes errors are uncorrelated, 575
- Error model, 581
bit flip error, 579, 581, 582
bit flip in 5-qubit code, 591
errors that arise during storage, 590
mathematical description, 581, 582, 623
multiple qubits, 583
no error, 581, 582
phase flip, 581, 582
phase flip in 5-qubit code, 591
qubit entangles with environment, 582
simultaneous bit flip and phase flip, 581,
582
spontaneous emission, 579
- Error models
bit flip, 568, 569
- Error operator
for 5-qubit code, 600

- Error processes, 568
 decoherence, 569
 dissipation, 568
 model, 569
- Error reconciliation, 542–544, 549
 CASCADE protocol, 544
- Error reduction by symmetrization, 575
 circuit, 576
 unitary matrix, 577
 usefulness, 579
 worked example, 577, 578
- Error reduction via symmetrization, 574
- Error syndrome, 585
 5-qubit code, 591
 corrective action, 591
 of non-degenerate code, 587
- Euclidean traveling salesman problem, 206, 222
- European Parliament
 allegations on improper use of intercepts, 509
 concern about intercepts, 508, 509
 report on ECHELON, 508
- European Union
 illegal telephone taps, 510
- Exact probabilistic quantum cloning, 468
- Expectation value, 424
 of anti-Hermitian operator, 485
 of Hermitian operator, 485
- Expectation value of operator, 328, 329
- Expectation value or observable
 based on quantum clones, 467
 worked example, 467
- Exponential data compression, 452
- Exponential growth, 221
 cf. polynomial, 223
- Exponential of a matrix, 345, 346
- Exponentiation
 in Diffie-Hellman cryptosystem, 267
- Extending the range of QKD
 in fiber, 547
- Extracting joint properties, 211
- Extracting results from quantum simulations, 328
 ancilla-assisted readout, 328, 330
 multi-ancilla-assisted readout, 330
- F**
- Factoring cf. multiplication, 223
- Factoring feats, 269
- Factoring in **NP**, 273
- Factoring integers
 largest factored to date, 513
 possibility of better classical algorithm, 514
 possibility of special purpose hardware, 514
- Fault-tolerant quantum computing, 611
 defined, 615
 principles, 615
- FCI, 322, 350
- Fedichkin, L., 190
- Fermi-Dirac statistics, 337, 338
- Fermionic algebra simulated by Pauli algebra, 341
- Fermionic simulation, 339, 341
- Fermionic systems simulation, 334
- Fermions, 336, 338, 523
 composite particles, 336
 spinless, 340
 under particle interchange, 336
- Feynman, R., 209, 213, 325
 problem of simulating fermions, 339
 simulating physics with computers, 326
 universal quantum simulator, 211
 wikiquote ranking, 210
- Feynman's plate trick, 13
- Fiber-optic cables, 510
 capacity relative to satellite links, 510
 splicing, 511
 tapped, 510
 tapping, 511
 Verizon tapped, 511
- Fiber-optic communications
 eavesdropping on, 510
- Fibonacci numbers, 408
 recursive formula, 207
- Fidelity, 460, 462
 non-unitary quantum computation, 193
 quantum data compressor, 449
- Fijany, A., 172
- Finite combinatorial processes, 207
- Fish code, 265
- FLASH paper, 458
- Foreknowledge of solutions not required, 251
- Formal system
 consistency unprovable within self, 218

- Formal system (*cont.*)
 incomplete if consistent, 217
- Formalist, 203
- Four color theorem, 219
 automated proof, 219
- Fourier series, 141
- Fredkin, E., 325
- FREDKIN gate, 61
 in error correction via symmetrization, 577
- Free space QKD, 550
- Free space quantum cryptography, 550
- Freedman, M., 339
- Frontier states in tangle-linear entropy plane, 433
- Full configuration interaction, 322, 350
- Fullerenes, 502
- Functions of matrices, 184, 185
- G**
- Galaxy 1 satellite, 552
 threats to, 553
- Gate
 controlled-add-one, 438
 downshift permutation, 137
 qubit cyclic left shift, 135
 qubit reversal, 131
 R_x , 76
 R_y , 76
 R_z , 76
- Gates
 teleporting, 633
- GCHQ, 267
- Generalized singular value decomposition, 174
- Generate-and-test algorithm, 244
 cf. quantum search, 248
 described quantumly, 244
- Genomics, 382
- German military
 use of Enigma machine, 265
- Ghirardi, G. C., 458
- GHZ state, 431
- Giant component, 299
- Gingrich, R., 190
- Gisin, N., 494
- Global phase
 no measurable consequence, 12, 359
- God
 wikiquote ranking, 210
- Gödel, K., 202, 207, 208
- Gödel's incompleteness theorem, 217
- Gödel's undecidability theorem
 physical analog, 218
- Goebel, A., 501
- Good index values, 252
- Goods, 306
- Gottesman, D., 594, 604, 634
- Gram-Schmidt orthogonalization, 383
- Graph coloring, 295
 critical point, 301, 302
- Grassl, M., 611
 database of quantum codes, 611
- Green laser tracking system
 atmospheric compensation, 550
- Groups, 189
- Grover, L., 230
- Grover's algorithm, 241, 245, 246
 approximate analysis, 247
 exact analysis, 249
 for **NP-Complete** problems, 303
 nested, 308, 309
 optimal number of steps, 248
 optimality, 254
 oracle, 243, 250
 oscillations in success probability, 249
 speeding up randomized algorithms, 255
 square root speedup, 249
 state synthesis, 256
 when number of solutions is unknown, 254
 with multiple solutions, 251
- GSVD, 174
- Gubernatis, J.E., 329, 339
- Guo, G.-C., 190, 469
- H**
- Hadamard gate, 74
- Hallgren, S., 375
- Hallgren's algorithm, 378, 379
- Halting problem, 216, 217
- Hamiltonian, 70, 72
 built from non-unitary operator, 191
 local easy to simulate, 327
 maximally general too hard to simulate, 327
- non-commuting, 327
- Trotter formula, 327

- Hammerer, K., 501
 Hamming distance, 579
 defined, 579
 Hamming weight, 440
 Hardest problem instances, 300, 302
 Hartree-Fock, 323
 Hawking, S. W., 201
 HBO interrupted by Captain Midnight, 551, 552
 Heath-Robinson code-breaking machine, 266
 Heisenberg Uncertainty Principle, 483, 484, 521
 definition, 485
 in energy and time, 570
 Heralded single photon source, 524
 Herapathite crystal, 527
 Herbert, N., 458
 Hermitian operators, 484
 Heuristic, 296
 Hidden variables, 488
 Hijacking satellites, 550
 Captain Midnight, 551
 Hilbert, D.
 mathematical challenges, 202
 Hilbert's Entscheidungsproblem, 215
 Hillery, M., 460
 Hogg, T., 307, 316
 Horizontally polarized photon, 523
 Horne, M., 442
 How classical world arises from quantum mechanics, 568
 How the “No Cloning theorem” got its name, 458
 Høyer, P., 252, 372
 Hubbard model, 324
 Hughes, R., 550
- I**
 Ideal quantum cloner, 459
 IdQuantique, 526, 547
 IdQuantique (Switzerland), 554
 Illusion of language understanding, 219
 Image compression, 162
 Imperfect code, 588
 Implementation
 QKD, 545
 Implicit assumptions, 235
 about classical computation, 7
 about classical information, 410
- Impossibility of exact deterministic quantum cloning, 458
 Impure code, 587
 Incompleteness theorem, 217
 Incompressibility
 of strings, 408
 Indistinguishability, 334, 335
 Industrial espionage, 509
 Inference, 203
 Information, 404
 is physical, 410
 Shannon view, 404
 Information transmission, 408
 Inner product, 10
 Integer factorization, 272, 273
 possibility of better classical algorithm, 514
 possibility of special purpose hardware, 514
 worked example, 280
 Integer factorization in NP, 273
 Integer factorization milestones, 224
 Integer factorization records, 269
 Intel, 4
 Intelligence agencies
 Australia, 508
 Canada, 508
 legitimate needs of, 508
 New Zealand, 508
 tapping fiber-optic cables, 511
 United Kingdom, 508
 USA, 508
 Intelligence gathering, 510
 Intercepts
 fiber-optic communications, 510
 satellite communications, 508, 509
 Interchange of particles, 337
 Interconversion between Bell states, 454, 455
 Interference, 20, 213
 of fullerenes, 502
 Intractable, 221
 Intractable problems, 267
 Inverse circuit
 in quantum error correction, 591
 Inverse unitary matrix
 in error reduction via symmetrization, 576
 Is quantum cloning useful?, 464

- Ising Hamiltonian, 72
Ising model, 298
ISWAP gate, 173
Italian navy, 265
- J**
Jacobs, K., 629
Joint bit flip and phase flip, 583
Joint bit flip and phase flip error, 590
 undoing, 593
Joint entropy, 409
Joint property functions
 computable by quantum parallelism, 215
 measurement of, 211
Jordan-Wigner transformation, 339, 340,
 342
Jozsa, R., 215, 230, 231, 382, 445, 486
JPEG, 162
Julsgaard, B., 501
Jussy, Switzerland
 in speed test of non-local influences, 495
Justus Lipsius building eavesdropping
 incident, 510
- K**
Kaye, P., 287, 437
Kendon, V., 190, 386
Ket as column vector, 18
Ket vector, 10
Key distribution
 via “trusted” couriers, 519
Key distribution problem, 519
Kick-back
 eigenvalue, 357, 361
Killer ap, 233
Kim, Y.-H., 443
Kirkpatrick, S., 316
Kitaev, A., 339
Klappenecker, A., 184, 189
Knill, E., 329, 339, 606
Known error is fixable, 585
Knox, D., 265
Koblitz, N., 286
Kolmogorov complexity, 408
Kowalski, R., 219
Krauter, H., 501
Kronecker product, 194, 574, 575
Kulik, S., 443
Kwiat, P., 538
- L**
La Palma
 in free space QKD experiment, 550
Laflamme, R., 329, 339, 606
Laflamme-Miquel-Paz-Zurek’s 5-qubit
 code, 588
 decoding circuit, 591
 encoding circuit, 590
 is not a CSS code, 605
 is stabilizer code, 605
Laguerre-Gaussian light beam, 537
Lai, C., 611
Lambda calculus, 207
Lanczos Algorithm, 351
Land, E., 527
Largest composite integer factored, 513
Latorre, J., 342
Lattice, 304, 305
Law
 Amdahl’s, 324
 Moore’s, 4
Law enforcement agencies
 legitimate needs of, 508
Legal requirements for protecting
 information, 512
 satellite communications, 512
Letter frequencies in English, 406
Leung, D., 635
Light
 as electromagnetic wave, 522
Light harvesting, 364
Linear entropy, 433
 as measure of mixedness, 423
 cf. von Neumann entropy, 422
 definition, 420
 for testing separability, 423
Linear optical quantum computers
 for quantum repeaters, 548
Linear polarization, 33
Linearly polarized light, 522
Linearly polarized photon, 523
Lloyd, S., 218, 326, 339, 361
 universal quantum simulator, 211
Lloyd-Zalka-Wiesner quantum simulation
 algorithm, 327
Local interaction, 486
 cf. non-local interaction, 488
 defined, 487
Locality loophole closed, 494

- Logical conjunction, 53
- Logical disjunction, 53
- Logical inference, 203
- Logical qubit, 593, 605, 614
 - cf. physical qubit, 605
- Logical reversibility, 63
- Lomonaco, S., 630
- Long range quantum communication
 - in fiber-optics, 548
 - in free space, 549
- Lorenz code, 265
- Lorenz machine, 265, 266
- Los Alamos National Laboratory, 550
- Loss of coherence, 568
- Low density parity check code, 453

- M**
- Macchiavello, C., 606, 607
- MacDougall, J.
 - a.k.a. "Captain Midnight", 551
- MagiQ Technologies, 547
- MagiQ Technologies (USA), 554
- Magnetic susceptibility
 - anomaly explained by entanglement, 443
- Magnetization, 298
- Magnons, 336
- Mair, A., 501
- Majority voting, 572
 - impossible quantumly in same way, 574
 - is advantageous when, 573
 - quantum analog, 573, 574
 - success of group versus individual, 573
- Malicious commands uplinked to satellite, 551
- Maloyer, O., 190
- Markov, I., 130, 174
- Maslov, D., 287
- Materials
 - computational design of, 324
- Materials science
 - role of entanglement, 443
- Mathematical proofs, 219
- Mathematical reasoning, 203
- Mathew, J., 287
- Matrix
 - direct product, 127
 - direct sum, 128
 - dot product, 126
- Matrix exponential, 345, 346
 - application, 247
- Matrix function, 184, 185
- Matrix permanent, 222
- Mattle, K., 500
- Maximally entangled
 - mixed state, 432
 - pure state, 431
- Maximally entangled pure states, 454
- Maximally entangled qubits, 629
- Maximum correctable errors, 605
- Maxwell-Boltzmann statistics, 334, 337
- Measure error syndrome, 591, 592
- Measure of mixedness, 420, 421
 - linear entropy, 420
- Measurement
 - of orthogonal states, 444
- Measurement-based quantum computation
 - one way, 640
 - teleportation-based, 633
- Measurements determine error, 585
- Measuring
 - single qubit, 15
- Measuring correlation functions, 333
- Medical records, 507
- Memoryless channel, 409
- Merkle, R., 267
- Message to integers, 271
- Milestones in factoring integers, 224
- Miller, V., 286
- Miniaturization, 4
 - atoms per bit, 6
 - transistors per chip, 4
- Minimal distance of a code, 586
- Minimal distance of pure $[n, k, d]$ code
 - approximate, 607
- MIP-year, 224
- Miquel, C., 329
- Mitchell, D., 303
- Mixed state
 - as part of larger pure state, 419
 - Bloch ball, 415
 - cf. pure state, 411
 - entangled, 428
 - from partial ignorance, 417
 - from partial knowledge, 411
 - maximally mixed, 415
 - non-maximally mixed, 415
 - purification, 419
 - quantifying mixedness, 420
 - separable, 427

- Mixed states, 444
Mixedness, 433
Model of errors, 569, 581
bit flip, 568, 569
bit flip error, 579, 581, 582
bit flip in 5-qubit code, 591
mathematical description, 581, 582, 623
multiple qubits, 583
no error, 581, 582
phase flip, 581, 582
phase flip in 5-qubit code, 591
phase shift, 569
qubit entangles with environment, 582
simultaneous bit flip and phase flip, 581, 582
simultaneous bit flip and phase shift, 570
spontaneous emission, 579
Models of computation, 202
equivalence, 213
Moore, G., 4
Mosca, M., 252, 437
adiabatic algorithm, 303
Most important quantum algorithm, 361
Mother wavelet, 153
MPEG, 162
Multi-ancilla-assisted readout, 330
Multiplication cf. factoring, 223
Munro, W., 433
Mutual information, 409
- N**
NAND gate, 55
NASA
error correcting codes, 580
National Security Agency, 263
approved cryptosystems, 513
tapping underwater fiber-optic cables, 511
Nature of reality, 493
Necessary and sufficient test for entanglement, 425
Need for stronger cryptography, 508
fiber-optic cables tapped, 510
regulatory pressure, 512
retroactive vulnerability, 512
satellite communications tapped, 508
Negating a qubit, 73
Negating quantum information, 470
Negativity, 430
Negrevergne, C., 329
Nested quantum search, 308
New Yorker magazine, 553
Newman, M., 203
Nielsen, M., 635
NMAJORITY gate, 57
No cloning theorem, 458, 460
discovery, 458
precludes error correction via majority voting, 572
No-Cloning theorem, 521
Nogoods, 306
Noise
in light harvesting, 364
Noise assisted quantum transport, 364
Noise-assisted quantum computing, 620
Noiseless source coding theorem, 403, 407
Noisy channel, 403
Noisy channel coding theorem, 403, 408
Non-additive code, 611
outperforms optimal stabilizer code, 611
union of additive codes, 611
Non-classical gate, 72
Non-commuting Hamiltonians, 327
Non-degenerate code, 587
Non-determinism, 16, 488
Non-local interaction
cf. local interaction, 488
defined, 488
Non-local interactions, 489
Non-locality, 486, 488
Non-stabilizer code, 611
Non-uniqueness of density operator, 416
Non-unitary quantum computation, 190
fidelity, 193
success probability, 193
Non-unitary transformation, 190
Nonlinear Kerr medium, 635
NOR gate, 55
Nordholt, J., 550
NOT gate, 58, 72
NOT gate from NAND gate, 56
NOT MAJORITY gate, 57
NSA, 263
Number field sieve, 263, 381, 514
complexity of, 513
Number field sieve algorithm, 223
Numerical circuit design, 180
example, 181, 182

- Numerical circuit design (*cont.*)
 limitation, 185
 Nyquist criterion, 144
- O**
 OAM states of light, 537
 Scottish dance, 537
 superdense coding, 538
 unsuitable for long distance QKD, 538
- Observables, 484
 mixed states, 424
- Observing
 single qubit, 15
- Oh, C., 611
- Olsson, R., 501
- One time pad cryptosystem, 515
- Operator
 expectation value, 328, 329
- Operator algebra, 339
- Operators
 mean square deviation, 484
- Optical lattice, 633
- Optical tweezers, 537
- Optimality
 Grover's algorithm, 254
- OR gate, 53
- Oracle, 233, 243
 in amplitude amplification, 250
 in mythology, 250
 in quantum search, 246
- Orbital, 338, 350
- Orbital angular momentum
 photon, 537
- Orbiting disco ball, 550
- Orthogonal subspace, 583
- Ortiz, G., 329, 339
- Oscillations in success probability, 249
- OTP cryptosystem, 515
 courier key distribution, 519
 example pad, 517
 fixed alphabet, 516
 impracticality of, 519
 integers to message, 517
 key pads must be protected, 518
 loopholes if used improperly, 518
 made practical by QKD, 521
 message to integers, 516
 needs true random numbers, 518
 problem of key distribution, 519
 protocol, 517
- simplicity, 516
 uses keys only once, 518
 voracious consumer of keys, 519
 worked example, 517
- Outer product, 10
- Over amplification, 249
- Overlap, 247, 249
- P**
 P complexity class, 226
- Packet sniffer, 511
- Pan, J.-W., 500, 501
- Papadopoulos, P., 629
- Parallelism
 quantum, 211
- Parametric downconversion, 524
- Partial solutions lattice, 305
- Partial trace, 417
 over GHZ state, 432
 over W state, 432
 to analyze quantum cloning, 461
 worked example, 418
- Partial transpose, 465
 definition, 425
 example, 427–429
- Particle statistics, 334
- Pattern matching, 382
- Pauli Exclusion Principle, 337, 338
 applied to atoms, 338
 applied to lattice of spinless fermions, 340
- Pauli matrices
 as a basis, 582
 in modeling errors, 582
- Pauli spin matrices, 71
- Pauli *X* gate, 73
- Pauli *X* not NOT for qubits, 74
- Pauli *X* gate, 72, 568
- Pauli *Y* gate, 570
- Pauli *Z* gate, 569
- Paz, J. P., 329
- Pell's equation, 375
 significance, 381
 why it is hard, 376
- Peres, A., 218, 458, 486
- Peres-Horodecki criterion, 425
 applied in quantum cloning, 465
- Perez-Delgado, C., 632, 633
- Perfect code, 588

- Period 4π rotations, 13
Period finding
 irrational period, 382
Periodic table of the elements, 338
Permanent of matrix, 222
Permutation
 of states in direct product, 575, 576
Permutation matrix, 131
 downshift, 137, 450
 Q , 450
 qubit cyclic left shift, 135
 qubit reversal, 131
Peterson, C.G., 550
Phase factor
 as binary fraction, 353, 355, 356
Phase flip error, 583, 590
 undoing, 593
Phase fronts
 in OAM states, 537
Phase inversion, 246, 247
Phase (of a wave), 522
Phase shift error, 569, 572
Phase state in eigenvalue estimation, 352
Phase transition
 3-SAT, 300, 303
 chromatic number, 299
 clique number, 299
 computational, 297, 299
 graph connectivity, 299
 size of giant component, 299
Photon
 polarization, 522, 523
Photon angular momentum, 523
Photon orbital angular momentum, 537
Photon polarization states, 523
Photon source
 entangled, 524
 heralded, 443, 524
 Poisson statistics, 524
 polarized
 via Pockels cell, 527
 via polarizers, 527
 single, 523
Physical analog
 Gödel's undecidability theorem, 218
Physical qubit, 593, 604
 cf. logical qubit, 605
Physics experiment resolves philosophical question, 489
Plasmons, 336
Plate trick, 13
Plenio, M., 629
Pockels cell, 527
Pockels cells
 as polarized photon source, 527
Podolsky, B., 488
Polarization, 522
 circular, 522, 523
 diagonal, 523
 elliptical, 522
 horizontal, 523
 linear, 522, 523
 measurement via birefringent crystal, 528
 measurement via polarizer, 529
 photon, 523
 quantum viewpoint, 523
 rectilinear (horizontal or vertical), 522
 vertical, 523
Polarization-entangled photons 144 km apart, 550
Polarizers, 527
 as polarized photon source, 527
 crossed, 529
Polarizing filters, 539
Polarons, 336
Pollard rho algorithm, 289
Polynomial growth, 221
 cf. exponential, 223
Polynomial in matrices, 186
Polzik, E., 501
Popescu, S., 436
Post, E., 202, 207, 208
Post Office, 266
Pottery glaze, 264
Pradhan, D., 287
Prepare-evolve-measure
 efficiency in simulations, 326
Prepare-evolve-measure cycle, 70
Preskill, J., 615
Primality testing, 219
Privacy amplification, 542, 544, 545, 549
Privacy rights, 508, 509
Private key, 267, 268
Probabilistic algorithms, 206
Probabilistic non-unitary quantum computation, 190
 fidelity, 193
 success probability, 193

- Probabilistic quantum cloning, 468, 469
 only works for linearly independent states, 469
 optimal, 470
 related to unambiguous state discrimination, 470
- Probabilistic Turing machine, 205
- Problem constrainedness, 301, 307
- Problem of key distribution, 519
- Product
 direct, 127
 direct product versus direct sum, 128
 dot, 126
- Product state, 423
- Projection
 into symmetric subspace, 575
- Proof versus truth, 217
- Prooffeats, 219
- Proos, J., 287
- Proving versus providing proof, 218
- PTM, 233
 cf. QTM, 211
- PTM, probabilistic Turing machine, 206, 207
- Public key, 267, 268
- Public key cryptosystem, 267, 268
 Diffie-Helman, 267
 retroactively vulnerable, 507
 RSA, 267, 268, 270
 vulnerable to quantum attack, 272, 507
- Public key cryptosystems, 513
- Pure code, 587
 quantum Singleton bound, 607
- Pure state
 cf. mixed state, 411
 density operator for, 416
 entangled, 427
 Schmidt decomposition, 433
 separable, 426
- Pure states
 maximally entangled, 436
 statistical mixture of, 444
- Purification of a mixed state, 419
- Q**
- Q gate, 450
- QBER, 539
- QCA, 631, 632
- QCT, 162, 171
 circuit, 166
- definition, 165
 matrix, 165
- QECC, 593
- QFFT, 186
- QFT, 140
 circuit, 150
 matrix, 148
 of computational basis state, 145
 unentangled, 146
 of superposition state, 147
 symmetric peaks, 147
 used in factoring, 273
- QHT, 186
- QinetiQ, 547
- QKD, 520
 and Heisenberg Uncertainty Principle, 521
 and No-Cloning theorem, 521
 Bank of England meeting, 547
 banking transactions, 547
 basis for security of, 520
 Bennett and Brassard's BB84 protocol, 521, 529, 531
 Bennett's B92 protocol, 521
 Bruss' 6-state protocol, 521
 concept, 520
 DARPA quantum network, 547
 Earth-to-Space, 549, 550
 Ekert's entanglement-based protocol, 521
 error reconciliation, 542–544
 extending range in fiber, 547
 free space, 549, 550
 implementations, 545
 in fiber-optics, 545
 BBN, 547
 makes OTP cryptosystem practical, 521
 maximum key distribution rate, 547
 maximum range in fiber, 547
 multiple protocols for, 521
 physical effects underpinning, 522
 polarized photons, 522
 practical for local area networks, 547
 privacy amplification, 542, 544, 545
 SECOQC quantum network, 547
 stray background photons, 549
 with coherent states, 521
 with orbital angular momentum states, 521

- QRNG, 526
QST, 171
QTM, 230, 231
 cf. DTM and PTM, 211
 cf. PTM, 211
QTM, quantum Turing machine, 210–212
 inability to see proof steps, 219
Quadrature mirror filter, 153
Quantifying entanglement, 429
 of outputs in quantum cloning, 466
Quantifying mixedness, 420, 421
 linear entropy, 420
Quantum algorithm
 algebraic number theory, 375
 eigenvalue estimation, 352, 361
 for chemical dynamics, 364
 for finding function collisions, 370
 for maxima and minima, 370
 for mean estimation, 370
 for median estimation, 370
 for **NP-Complete** problems, 302
 for protein conformations, 364
 in photosynthesis, 364
 mean estimation, 370
 pattern matching, 382
 Pell's equation, 375
 phase estimation, 354
 quantum counting, 371, 373
 quantum image processing, 382
 quantum walk, 385
 sequence comparison, 382
 Shor, 272, 273
 worked example, 280
 template matching, 382
Quantum bit error rate, 539
Quantum broadcasting
 cf. cloning, 470
Quantum cellular automata, 630–632
 local unitary, 632
 universality, 633
Quantum chemistry, 349
Quantum circuit
 augment-on-fail compression, 449
 complexity of, 130
 compressor, 452
 determining spectral density, 333
 discard-on-fail compression, 448
 efficient, 130
 eigenvalue estimation, 361
 entanglement swapping, 442
 equivalent unitary matrix, 124
 error reduction via symmetrization, 576
 extracting correlation functions, 334
 for given unitary matrix, 172
 for inverse operations, 450
 inefficient, 130
 multiple ancilla-assisted readout, 331
 quantum counting, 374
 quantum nested search, 312
 quantum structured search, 312
 quantum tree search, 312
 Schumacher-Jozsa compressor, 450
 Schumacher-Jozsa decompressor, 450
 Shor's algorithm, 273
 single ancilla assisted readout, 330
 spectroscopy, 332
 superdense coding, 457
 tomography, 332
 universal approximate quantum cloning,
 463
 universal quantum NOT, 471
 UNOT, 471
Quantum circuit complexity
 arbitrary unitary, 173
Quantum circuit design
 algebraic, 174
 algebraic example, 179
 applications, 173
 automated, 172
 choice of gates, 173
 compactification, 178
 genetic, 172
 numerical, 180
 numerical example, 181, 182
 re-use, 184, 187
Quantum circuit diagrams, 123
Quantum circuits, 123
 probabilistic non-unitary, 190
Quantum cloning, 457
 cf. broadcasting, 470
 cf. quantum universal NOT, 471
 circuit, 463
 degree of entanglement of clones, 466
 entanglement of clones, 465
 expectation value of observable of a
 clone, 467
 fidelity input independent, 460
 historical roots, 457
 ideal, 459
 impossible exactly deterministically, 458

- Quantum cloning (*cont.*)
 outputs are weakly entangled, 466
 proof clones are entangled, 465
 protocol, 460
 universal approximate, 460
 usability of clones, 464
 usability of the clones, 467, 468
 used for eavesdropping, 458
- Quantum code-breaking, 263, 272
- Quantum codebreaking
 relative speedup breaking RSA cf. ECC,
 287
- Quantum codes
 types, 593
- Quantum complexity, 201
- Quantum complexity classes, 229
- Quantum compression, 444
- Quantum compressor, 447
- Quantum computability, 201, 214
 cf. classical, 235
 matches classical, 215
- Quantum computation
 alternative models, 627
 distributed, 628
- Quantum computers
 topological, 507
- Quantum computing
 based on teleportation, 633
 causes of errors, 568
 distributed, 443
 indefinitely without error, 617
 noise-assisted, 620
- Quantum cosine transform, 162, 171
 circuit, 166
 definition, 165
 matrix, 165
- Quantum counting, 371, 373
- Quantum counting algorithm, 254
- Quantum cryptography, 507
 are foundations solid?, 556
 banking transactions, 547
 certification, 556
 choice of classical cryptosystem affects
 security, 558
 commercially available systems, 554,
 555
- DARPA quantum network, 547
- Earth-to-Space, 549, 550
- European collaboration, 510
 free space, 547, 549, 550
- human factors, 558
 implementations, 545
 in fiber-optics, 545
 is it needed?, 555
 key distribution rate, 558
 loopholes due to imperfections?, 556
 man-in-the-middle attack, 557
 maximum key distribution rate, 547
 maximum range in fiber, 547
 need for authenticated channel, 557
 overkill?, 556
 practical for local area networks, 547
 QKD, 520
 quantum key distribution, 520
 range in fiber, 557
 repeater, 557
 SECOQC quantum network, 547
 stray background photons, 549
 wide area networks, 557
 with satellites, 549
- Quantum data compression, 445
 augment-on-fail, 449
 circuit, 448, 449, 452
 discard-on-fail, 447
 fidelity, 449
 unitary matrix for compressor, 450
- Quantum data processing, 382
- Quantum eigenvalue estimation, 352, 361
 Abrams-Lloyd circuit, 361
 Abrams-Lloyd scheme, 361
 Kitaev circuit, 362
 Kitaev scheme, 361
 phase state, 352
 restoration of starting state, 363
- Quantum entropy, 445
- Quantum error correcting code, 584
 bounds, 605
 first, 593
 Laflamme-Miquel-Paz-Zurek 5-qubit,
 588
 Laflamme-Miquel-Paz-Zurek 5-qubit
 code
 decoding circuit, 591
 encoding circuit, 590
- $\llbracket n, k, d \rrbracket$, 607
 $\llbracket n, k, d \rrbracket$, 587
 $\llbracket n, K, d \rrbracket$, 587
 maximum number of correctable errors,
 605
 Shor 9-qubit, 593

- Quantum error correcting code (*cont.*)
stabilizer, 587
Steane 7-qubit, 594
- Quantum error correcting codes
approximate bounds, 607
assumptions of when and where errors occur, 611
correcting more than one error per block, 611
feasible despite skeptics, 593
imperfect error correction, 615
non-additive, 611
non-stabilizer, 611
proven bounds, 611
re-exposing the protected state, 614
stabilizer, 594
tighter bounds, 611
- Quantum error correction, 567, 572
by coding, 579
by symmetrization, 573–575
assumes errors are uncorrelated, 575
circuit, 576
unitary matrix, 577
usefulness, 579
worked example, 577, 578
- cf. classical, 571, 572
concatenated codes, 617, 619, 620
error model, 581, 582
fault tolerant, 611, 615
skepticism, 567
the trick, 573
threshold theorem, 617, 619, 620
why is it hard?, 571
- Quantum error reduction by symmetrization, 575
circuit, 576
unitary matrix, 577
usefulness, 579
worked example, 577, 578
- Quantum factoring, 272, 273
worked example, 280
- Quantum Fourier transform, 140
circuit, 150
matrix, 148
of computational basis state, 145
of superposition state, 147
symmetric peaks, 147
used in factoring, 273
- Quantum fractional Fourier transform, 186
- Quantum gate
controlled modular-add-one (C-MAO), 169
noise-assisted, 620
- Quantum gates
for classical reversible chips, 139
- Quantum Gilbert-Varshamov bound, 605, 607
- Quantum Hamming bound, 605–607
- Quantum Hartley transform, 186
- Quantum image processing, 382
two-dimensional transforms, 384
- Quantum information, 403, 411
broadcasting, 470
cloning, 460
compression, 444
negation, 470
speed of non-local influences, 494
- Quantum information cf. classical, 572
- Quantum interference, 213
- Quantum key distribution, 520
defined, 520
- Quantum logic gates, 69
- Quantum $[n, K, d]$ code, 587
- Quantum $\llbracket n, k, d \rrbracket$ code, 587
- Quantum mean estimation, 370
- Quantum memory, 501
in quantum repeaters, 548
- Quantum memory register, 17
- Quantum money, 530, 531
- Quantum nested search, 302, 308, 309
circuit, 312
complexity analysis, 309
- Quantum NOT gate, 73
- Quantum numbers
interrelationships between, 338
Pauli Exclusion Principle, 338
- Quantum parallelism, 211, 214, 230
class of functions computable by, 215
- Quantum pattern matching, 382
- Quantum permutation
downshift, 450
 Q , 450
- Quantum permutations, 131
- Quantum phase estimation, 354
- Quantum random number generator, 526
commercially available (idQuantique), 526
- Quantum repeater, 443, 502
Briegel-Dürr-Cirac-Zoller scheme, 548

- Quantum repeater (*cont.*)
 demonstrated, 548
 diagram, 548
- Quantum repeaters, 547
- Quantum search, 241, 245, 246
 approximate analysis, 247
 exact analysis, 249
 foreknowledge of solutions not required, 246
 optimal number of steps, 248
 optimality, 254
 oracle, 243, 250
 oscillations in success probability, 249
 oscillatory success probability, 248
 real databases, 260
 speeding up randomized algorithms, 255
 square root speedup, 249
 state synthesis, 256
 when number of solutions is unknown, 254
 with multiple solutions, 251
 with prior knowledge, 173
- Quantum sequence comparison, 382
- Quantum signal processing, 382
- Quantum simulation, 325
 ancilla-assisted readout, 328, 330
 between operator algebras, 339, 340
 bosons simulating fermions, 339
 extracting correlation functions, 333
 extracting result, 328
 fermionic, 339
 fermions, 326, 334
 fermions versus bosons, 334
 Lloyd-Zalka-Wiesner algorithm, 327
 multi-ancilla-assisted readout, 330
 non-interacting Hamiltonians, 341
 of quantum systems, 326
 tomography versus spectroscopy, 332
 topological field theories, 339
 via disentanglement, 342
 via Jordan-Wigner transformation, 341
 via mapping to non-interacting Hamiltonians, 342
- Quantum sine transform (QST), 171
- Quantum Singleton bound, 605–607
 tighter for pure codes, 607
- Quantum source, 444, 453
- Quantum speedup
 Grover's algorithm, 251
- Quantum state
 pure versus mixed, 411
- Quantum state synthesis, 173
- Quantum state tomography, 332
 circuit, 332
 computational cost, 424
 inefficient, 328
- Quantum structured search, 309
 circuit, 312
 complexity analysis, 309
- Quantum teleportation, 496, 497
 after measurement in Bell basis, 498
 between objects of dissimilar type, 501
 cf. cloning, 486
 cf. faxing, 486
 cf. science fiction teleportation, 486
 circuit, 497
 corrective actions, 498
 experiment, 500
 experiments, 486
 in quantum repeaters, 502
 initial state, 498
 initial state rewritten, 498
 larger objects, 501
 multi-particle entangled state, 501
 not superluminal, 499
 protocol, 498
 requires original is destroyed, 486
 state vs. object, 486
 working prototypes, 500
- Quantum template matching, 382
- Quantum tree search, 309
 circuit, 312
 complexity analysis, 309
- Quantum Turing machine, 210–212
 entangled states, 211
 inability to see proof steps, 219
- Quantum universal NOT, 470
 cf. quantum cloning, 471
 exact predictions from approximate negated state, 472
 ideal is impossible, 471
 quantum circuit, 471
 usability of negated state, 472
- Quantum universality, 201
- Quantum walk
 bias, 390, 391
 excluded locations, 390
 one-dimensional, 387
 rate of diffusion, 395

- Quantum walk (*cont.*)
 unbiased, 393, 395
 variance in position of walker, 396
 worked example, 389, 391–393
- Quantum walks, 385
- Quantum wavelet transform, 151
 circuit, 158
 Daubechies, 153
 factorization of $D_{2^n}^{(4)}$, 157
 packet, 158
 pyramidal, 160
- Quantum-to-classical readout, 385
- Quatum cloning
 exact predictions based on approximate clones, 468
- Qubit, 9
 ancilla, 575
- Qubit cyclic left shift, 135
- Qubit reversal, 131
- Qudit
 using OAM states of light, 537
- Query complexity, 221
- Quinine, 527
- Quintessence Laboratories (Australia), 554
- Quotation
 Asimov, Isaac, 293
 Aspuru-Guzik, Alan, 361
 Babbage, Charles, 567
 Bruguiere, Jean Louis
 hijacking satellites, 553
 Cerf, Nicolas, 470
 Clarke, Arthur C., 554, 627
 Deutsch, David, 3, 201
 Dirac, Paul, 349
 Einstein, Albert, 483
 Ekert, Artur, 541
 European Parliament on ECHELON, 509
 Feynman, Richard, 51, 319, 410
 simulating bosonic vs. fermionic systems, 326
 from Computerworld on tapping fiber-optic cables, 511
- Fuchs, Kristen, 241
 Gisin, Nicolas, 403
 Hardy, Godfrey Harold, 369
 Hayes, Jim, 511
 Knapp, Bruce, 123
 Landauer, Rolf, 410
 Newton, Isaac, 184, 488
 on Feynman by Los Angeles Times, 326
- Peres, Asher, 458
- Pescatore, John, 511
- Preskill, John, 583
- Rose, Geordie, 319
- Rutherford, Ernest, 429
- Schrödinger, Erwin, 422
- Shakespeare, William, 507
- Shannon, Claude, 404
- Shor, P., 375
- Woolsey, James
 on use of intelligence, 509
- Zak, Michail, 263
- Zeilinger, Anton, 404, 493
- Quotation: Artur Ekert, 4
- Quotation: Ekert, 4
- Quotation: Feynman, 7
- QWT, 151
 circuit, 158
 Daubechies, 153
 factorization of $D_{2^n}^{(4)}$, 157
 packet, 158
 pyramidal, 160
- R**
- Rains, E., 607, 611
- Random graph, 299
- Random walks
 quantum, 385
- Randomized algorithm
 Euclidean traveling salesman, 206
- Randomized algorithms, 222
- Randomness
 as incompressibility, 408
- Range of QKD
 in free space, 550
- Range record for QKD, 547
- Ratio of clauses to variables, 300
- Re-use method of circuit design, 184
 circuit template, 187
 example (QFFT), 188
 example (QHT), 188
- Reactivity, 349
- Read error syndrome, 591, 592
- Reading
 single qubit, 15
- Reading a qubit, 16
- Reading quantum memory register
 disrupts state, 574
- Reality, 493
 is non-local, 493

- Record-setting factoring feats, 269
 Rectilinearly polarized light, 522
 Rectilinearly polarized photon, 523, 526, 532
 Recursive functions, 207
 Reduced density matrix, 468
 Reductio ad absurdum, 215
 Redundant information
 in symmetrized state, 573, 574
 used to boost reliability, 572
 Regulator, 377
 Hallgren's algorithm, 378
 period finding, 378
 Relative entropy of entanglement, 430
 Relative phase, 359, 572
 Reliable communications, 408
 Reliable computing, 573
 Report on ECHELON, 508
 Retroactive vulnerability, 507, 512, 514
 Reversibility, 63
 Reversible classical gates, 57
 Reversible computers, 63
 Reversible computing, 140, 210
 Reversible gate, 131
 Reversible gate as permutation matrix, 61
 Reversible Turing machine, 210
 Rewrite rules, 178
 canonical, 178
 Church-Rosser, 178
 example, 179
 Rivest, R., 267
 Röettler, M., 184, 189
 Rose, G., 319
 Rosen, N., 488
 Rotation gate, 76
 Rotations with period 4π , 13
 RSA, 507
 RSA algorithm, 270
 RSA cryptosystem, 513
 RSA public key cryptosystem, 267, 268, 270
 basis for security, 268
 basis of security, 267
 decryption algorithm, 268, 270
 efficiency, 272
 encryption algorithm, 268, 270
 worked example, 271
 RSA-100, 268
 RSA-129, 224
 RSA-200
 factored, 269
 factoring record, 513
 RSA-576
 change in naming convention, 269
 factored, 269
 RSA-640, 269
 S
 Salart, D., 494
 Saraceno, M., 329
 Satellite
 Captain Midnight, 551
 hacker attack, 550
 potential loss of command and control, 550
 Satellite communications
 eavesdropping on, 508
 to be made more secure, 512
 Satigny, Switzerland
 in speed test of non-local influences, 495
 Satisfiability, 295–297
 critical point, 300
 Scherbius, A., 265
 Schmidt decomposition, 433
 worked example, 435
 Schmiedmayer, J., 501
 Schrödinger equation, 320
 deterministic evolution, 573
 molecular, 349
 numerical solution, 320
 time dependent many-electron, 349
 time independent local Hamiltonian, 327
 Schrödinger's equation, 70
 deterministic evolution, 574
 in terms of density operator, 416
 Schumacher, B., 436, 445
 Schumacher-Jozsa quantum data compression, 445
 Scottish dance
 split-the-willow, 537
 Scumacher, B., 632
 Scytale, 264
 Search algorithm, 241, 245
 Second World War, 265, 266
 cooperation on intelligence, 508
 SECOQC
 demonstration, 547
 SECOQC quantum cryptography system, 510
 demonstration, 510

- Secure communications
 - Enigma, 265
- Security
 - unconditional, 508
- Security foundations of QKD, 520
- Security of OTP cryptosystem
 - if used improperly, 518
- Selman, B., 300, 303, 316
- Separability
 - of mixed state, 423
- Separable
 - mixed state, 427
 - pure state, 426
- Separable state, 423
- Sequence comparison, 382
- Shamir, A., 267
- Shannon, C., 403
 - noiseless source coding theorem, 403, 407
 - noisy channel coding theorem, 403, 408
 - view of information, 404
- Shannon entropy, 405, 444
 - can equal von Neumann entropy, 453
 - definition of H , 407
- Shannon limit, 452
- Shared prior entanglement, 454, 629
- Shell, 338
- Shende, V., 130, 174
- Sherson, J., 501
- Shih, Y., 443
- Shor, P., 230, 233, 263, 593, 607, 611
- Shor's 9-qubit code, 593
- Shor's algorithm, 263, 272, 273
 - discrete logarithms, 285
 - distributed, 630
 - quantum circuit, 273
 - relative speedup breaking RSA cf. ECC, 287
 - worked example, 280
- SHRDLU robot, 219
- Signal
 - as superposition, 144
 - cosine series, 163
 - discretization, 142
 - sampling, 142
 - sampling rate, 144
- Signalling entanglement
 - via entanglement witness, 423
 - via Peres-Horodecki criterion, 425
- Simon, D., 233
- Simulating fermions, 334
- Simulating physics with computers, 213
 - representing quantum states, 214
- Single photon source, 523
 - exactly one photon per pulse, 524
 - heralded, 524
 - less than one photon per pulse, 524
 - Poisson statistics, 524
- Singlet, 423
- Singlet state, 496
 - special property of, 496
- Singular value decomposition
 - in circuit design, 174
 - in Schmidt decomposition, 434
- Size computational problem, 221
- Skepticism for quantum error correction, 567
- Sloane, N.J.A., 607, 611
- SmartQuantum (France), 554
- Smolin, J., 172
- Solenov, D., 190
- Somma, R., 329, 339
- Source
 - classical, 405
 - quantum, 444, 445
- Source coding theorem, 403, 407
 - cf. superdense coding, 453
- Space Shuttle
 - communications hacked, 554
 - majority voting, 573
 - reliable computing, 573
- Spacelike separated, 487
- Spacelike separated entangled particles, 489
- Special purpose factoring engine
 - TWINKLE, 514
 - TWIRL, 515
- Special relativity
 - cf. non-local effects, 488
- Spector, L., 172
- Spectral density in vicinity of known energies, 333
 - circuit, 333
- Spectroscopy
 - circuit, 332
- Spedalieri, F., 537, 538
- Spedalieri's orbital angular momentum scheme for BB84, 537
- Speed of non-local influences, 494
 - at least 10, 495

- Speedup
 - superpolynomial, 381
 - Spin, 523
 - bosons, 523
 - fermions, 523
 - in magnetism, 298
 - Spin-flip operator, 466
 - Spinless fermions, 340
 - Spintronics, 140
 - Splicing
 - fiber-optic cables, 511
 - Split-the-willow
 - OAM states of light, 537
 - Spontaneous emission
 - bit flip error, 579
 - Spooks, 263
 - Spooky action at a distance, 488
 - Square root of NOT, 72
 - Stabilizer
 - 5-qubit code, 596
 - codewords, 597
 - error operators commute with, 601
 - for 5-qubit code, 600
 - tensor product of Pauli matrices, 595
 - Stabilizer code, 587, 594, 604, 611
 - advantage of, 594
 - if pure is non-degenerate, 587
 - Laflamme-Miquel-Paz-Zurek's 5-qubit code, 588
 - State
 - GHZ, 431
 - mixed entangled, 428
 - mixed separable, 427
 - pure entangled, 427
 - pure separable, 426
 - separable versus entangled, 422
 - singlet, 423
 - W, 431
 - State synthesis
 - applications, 383
 - via Grover's algorithm, 257
 - Steane's 7-qubit code, 594
 - Stirling's formula, 307, 350
 - Stray photons in free space QKD, 549
 - String compressibility, 408
 - Strong Church-Turing thesis, 208
 - cf. Deutsch's thesis, 209
 - quantum challenge to, 209
 - Strong cryptography, 515
 - need for, 508
 - Strong electron correlations, 341
 - Subexponential, 223
 - Submarines for eavesdropping, 511
 - Subset sum, 295
 - Substitution code, 271
 - Success probability
 - non-unitary quantum computation, 193
 - Superconductivity, 337
 - Superdense coding, 453
 - circuit, 457
 - does not violate Source Coding Theorem, 453
 - of "maximally compressed" classical message, 453
 - protocol, 455, 456
 - Superfluidity, 337
 - Superluminal communicator, 458
 - Superoperator, 424
 - Superpolynomial, 223, 273
 - Superpolynomial speedup, 381
 - Superposition, 9, 17
 - equally weighted, 576
 - number of components not power of two, 576
 - of n factorial states, 576
 - SWAP gate, 58
 - controlled a.k.a. "FREDKIN", 577
 - SWAP gate from CNOT gates, 59
 - Symmetric private key cryptosystem, 515
 - Symmetric state vector, 335, 336
 - Symmetric subspace, 574, 575
 - projecting into, 575
 - Symmetrization, 573, 574
 - assumes errors are uncorrelated, 575
 - buggy state, 575
 - circuit, 576
 - operation, 575
 - unitary matrix, 577
 - usefulness, 579
 - worked example, 577, 578
 - Symmetrized state, 576
 - Synthesis of Bell states, 454
 - Synthesize mixed state, 398
 - Synthesize pure state, 383
- T**
- Tangle, 429, 433
 - in analyzing quantum clones, 466
 - worked example, 466

- Tangle vs. linear entropy plane, 433
Tapp, A., 252, 372
Tapping
 detectable, 511
 fiber-optic cables, 510, 511
 fiber-optic communications channels, 510
 satellite communications channels, 508
 underwater fiber-optic cables, 511
 challenges, 512
 undetectable, 511
Telephone traffic
 mainly fiber-optic based, 510
Teleport
 unknown quantum state, 486
Teleportation
 circumvents apparent limitations, 486
 CNOT gate, 638
 in quantum computation, 633
 in science fiction, 483
 supposed impossibility, 483
Teleportation-based quantum computing, 633
Template matching, 382
Tenerife
 in free space QKD experiment, 550
Tensor product, 194, 574, 575
Terashima, H., 190
Terror web article
 New Yorker magazine, 553
Threshold theorem for concatenated codes, 617, 619, 620
Tight-binding, 322
Time scale
 decoherence, 569
TOFFOLI gate, 61, 450
Tokyo teleport station, 486
Tomography versus spectroscopy, 332
Topological quantum computers, 507
Torque of a photon, 537
Toshiba Research Europe
 QKD, 547
Trace
 in expectation values, 424
Tractable, 221
Tradeoff between physical qubits, logical qubits, and number of correctable errors, 605
Tradeoffs in determinism, verifiability and universality, 218
Transformation
 Bogoliubov, 343
 Jordan-Wigner, 339
Transistors per chip, 4
Traveling salesman problem, 206
Travelling salesman, 295
Tree search
 analyzed using lattice, 305
 quantum, 309
Tree-search, 304, 305
Trotter formula, 327, 328, 346
 high order approximations, 328
True random number generator, 173, 532
True random numbers, 214, 518
 creating, 525
 quantum generator, 526
Truth or falsity of mathematical proposition, 203
Truth or falsity of mathematical propositions, 215, 217
Truth table, 54
 AND, 54
 CNOT, 58
 NAND, 56
 NOT, 58
 \oplus , 55
 OR, 54
 SWAP, 58
 XOR, 55
Truth versus proof, 217
TSP, traveling salesman problem, 206
Tucci, R., 172, 174
Turbo code, 453
Turbulence in free space QKD, 549
Turing, A., 202, 203, 265
 Turing machine, 204, 205
 view of mathematical reasoning, 203, 204
Turing bombs, 265, 266
Turing machine, 205, 215, 265
 deterministic, 204
 equivalence to modern computers, 205
 fatal flaw, 209
 probabilistic, 205
 quantum, 210, 211
 universal, 208
Turing machine, reversible, 210

- Turing's halting problem, 216
 Tute, B., 265
 TWINKLE factoring engine, 514, 515
 TWIRL factoring engine, 515
 Twisting intertwined phase fronts, 537
 Two electrons per orbital, 338
 Two-dimensional quantum transforms, 384
- U**
- U-boats
 Enigma machine, 265
 UK Department of Trade and Industry
 interest in QKD, 547
 Ueda, M., 190
 UKUSA alliance, 508
 Unambiguous state discrimination, 540
 Unambiguously distinguishable states, 453
 Uncertainty principle, 483, 484
 Uncomputable, 213
 Unconditional security, 508
 Unconditionally secure cryptosystem, 515,
 521
 Uncorrelated errors, 575
 Uncounterfeitable banknote, 530
 Undetected monitoring
 satellite communications, 509
 Unique computational tasks, 235
 Unitary evolution, 70
 Unitary matrix
 complexity to implement, 173
 conversion to quantum circuit, 172
 embedding non-unitary matrix, 191
 equivalence to diagonal matrix, 185
 error reduction via symmetrization, 577
 for given quantum circuit, 124
 inverse, 576
 Unitary matrix: logical reversibility, 70
 Universal approximate quantum cloning,
 460
 quantum circuit, 463
 Universal gate: FREDKIN, 60
 Universal gate: TOFFOLI, 60
 Universal gates: reversible computing, 55
 Universal NOT, 470
 exact predictions from approximate
 negated state, 472
 quantum circuit, 471
 usability of negated state, 472
 Universal NOT gate, 73
- Universal probabilistic quantum cloning,
 468, 469
 Universal quantum simulator, 211, 326
 Universal Turing machine, 208
 Universality, 201, 202, 208, 213, 218
 distributed quantum computation, 630
 quantum cellular automata, 633
 UNOT, 470
 exact predictions from approximate
 negated state, 472
 ideal is impossible, 471
 quantum circuit, 471
 usability of negated state, 472
 Unsimulatable phenomena, 322
 Unstructured quantum search, 241, 245, 246
 approximate analysis, 247
 exact analysis, 249
 foreknowledge of solutions not required,
 246
 optimal number of steps, 248
 optimality, 254
 oracle, 243, 250
 oscillations in success probability, 249
 real databases, 260
 speeding up randomized algorithms, 255
 square root speedup, 249
 state synthesis, 256
 when number of solutions is unknown,
 254
 with multiple solutions, 251
 Unstructured search problem, 242, 245
 Unwanted entanglement with environment,
 569
 Urine of a dog fed quinine, 527
 Usability of quantum clones, 464
 Use entanglement to fight entanglement, 583
 UTM, universal Turing machine, 208
- V**
- Vacuum tube based computer, 266
 Van Dam, W., 632
 adiabatic algorithm, 303
 Van der Waals forces, 323
 Vazirani, U., 213, 230, 233
 adiabatic algorithm, 303
 Vec operation, 384
 Vedral, V., 443
 Verifiability, 218
 Vernam cipher, 515

- Verstraete, F., 342
Vertically polarized photon, 523
Video compression, 162
Visualizing quantum clones, 462
Von Neumann, J., 630
Von Neumann entropy, 445, 446, 453
 as measure of mixedness, 421, 422
 can equal Shannon entropy, 446
 cf. linear entropy, 422
 of a maximally mixed state, 422
 of a pure state, 422
- W**
W state, 431
Wagenknecht, C., 501
Wall Street Journal
 on submarines for eavesdropping, 512
Walsh-Hadamard gate, 245
Wang, Z., 339
Watrous, J., 632
Wavefunction
 anti-symmetric, 335, 336, 339, 350
 symmetric, 335, 336
 under interchange of electrons, 350
Wavelet filter coefficients, 153, 154, 157
Wavelet transforms, 152
Wavelets, 151
Weinfurter, H., 500
What can computers do, 213
Wiesner, S., 529
Williams, C.P., 172, 190, 307, 316
 adiabatic algorithm, 303
 where are the hard problems?, 222
Wired communications
 insecurity, 510
- Wireless communications
 insecurity, 510
Witness, 423
Wolfram, S., 630
Woolsey, J., 509
Wootters, W., 458, 486
Worst case complexity, 222, 296
Wright, L.
 terror web article, 553
- X**
Xerox PARC, 201
- Y**
Yamamoto, Y., 547, 635
Yang, T., 501
Yimsiriwattana, A., 630
Yu, S., 611
- Z**
Zalka, C., 254, 287
Zbinden, H., 494
Zeiliner, A., 500
Zeilinger, A., 442, 443, 493, 537, 632
Zhang, Q., 501
Zhao, B., 501
Zoller, P., 548
Zoo
 classical complexity, 225
 quantum complexity, 229
Zukowski, M., 442
Zurek, W., 218, 458