Quantum error correction (Random notes)

QECC – quantum error-correcting code (a mapping of k qubits (Hilbert space of dimension 2^k) into n qubits (Hilbert space of dimension 2^n) where $n > k$. k qubits need to be protected, $n-k$ qubits store them in redundant fashion.

Criteria – Consider a single qubit ~~acting~~ being acted upon by the environment in an arbitrary method. The initial state may be represented by $|0\rangle_E$ a pure state without loss of generality, by

Evolution of qubit and the environment –

$$U: |0\rangle \otimes |0\rangle_E \rightarrow |0\rangle \otimes |e_{00}\rangle_E + |1\rangle \otimes |e_{01}\rangle_E$$

$$|1\rangle \otimes |1\rangle_E \rightarrow |0\rangle \otimes |e_{10}\rangle_E + |1\rangle \otimes |e_{11}\rangle_E$$

where $|e_{ij}\rangle_E$ are mutually orthogonal states of the environment.

Now considering an arbitrary state $|\psi\rangle = a|0\rangle + b|1\rangle$

$$(a|0\rangle + b|1\rangle)|0\rangle_E = a|0\rangle \otimes |0\rangle_E + b|1\rangle \otimes |1\rangle_E$$

$$= a|0\rangle|e_{00}\rangle_E + a|1\rangle|e_{01}\rangle_E + b|1\rangle|e_{10}\rangle_E + b|1\rangle|e_{11}\rangle_E$$

$$= (a|0\rangle + b|1\rangle) \otimes \left(\frac{1}{2}\left[|e_{00}\rangle_E + |e_{11}\rangle_E\right]\right) +$$

$$(a|0\rangle - b|1\rangle) \otimes \frac{1}{2}(|e_{00}\rangle_E - |e_{11}\rangle_E) +$$

$$(a|1\rangle + b|0\rangle) \otimes \frac{1}{2}(|e_{01}\rangle_E + |e_{10}\rangle_E) +$$

$$(a|1\rangle - b|0\rangle) \otimes \frac{1}{2}(|e_{01}\rangle_E - |e_{10}\rangle_E)$$

$$= I|\psi\rangle \otimes |e_I\rangle_E + X|\psi\rangle \otimes |e_X\rangle_E + Y|\psi\rangle \otimes |e_Y\rangle_E +$$

$$Z|\psi\rangle \otimes |e_Z\rangle_E$$

(60) Reed-solomon codes — Galois field — a field satisfying the finite property of the number of elements. Quantum register of length n is obtained by combining n-qubits modelled by the n-fold tensor product $(\mathbb{C}^2)^{\otimes n}$. The canonical orthonormal basis is given by:

$$B = \{ |b_1\rangle \otimes \dots \otimes |b_n\rangle =: |b_1 b_2 \dots b_n\rangle = |b\rangle \mid b_i \in \{0,1\}$$

State of a n qubit register:

$$|\psi\rangle = \sum_{b \in \{0,1\}^n} c_b |b\rangle \text{, where } c_b \in C \text{ and}$$

$$\sum_{b \in \{0,1\}^n} |c_b|^2 = 1$$

ex For a 4 qubit quantum register, the orthonormal basis would be {0000, 0010, 0100, 1000} or one eigenvector in each dimension.

Assumption in quantum error-correcting code — errors are local, i.e. only a small number of qubits are disturbed when transmitting or storing the state of the n qubit register.

Bit flip errors — Pauli X Both — Pauli Y.
Phase flip errors — Pauli Z It is sufficient to consider only these 3 in order to cope with any possible local error.

Errors operating on a n qubit system are tensor products of Pauli matrices and identity.

The weight of an error $e = e_1 \otimes e_2 \otimes e_3 \dots \otimes e_n$, where $e_i \in \{\text{identity}, \sigma_x, \sigma_y, \sigma_z\}$

Quantum reed-solomon code is constructed using a weakly self-dual binary codes.

Def 1: $C = [N, K]$ - weakly self-dual linear binary code, i.e. $C \leq C^{\perp}$ and let $\{ w_j \mid j = 1, \dots, 2^{N-2k} \}$ be a system of representatives of cosets $C^{\perp}/C$. (left coset)
Then basis states of quantum code $C = [[N, N-2K]]$

$$|\psi_j\rangle = \frac{1}{\sqrt{|C|}} \sum_{c \in C} |c + w_j\rangle$$

① Self-dual code is one that is its own dual.
Dual of a code $C \subset F_q^n$ is the linear code defined by $C^\perp = \{x \in F_q^n \mid \langle x, c \rangle = 0 \ \forall c \in C\}$
where $\langle x, c \rangle = \sum_{i=1}^{n} x_i c_i = 0$; or orthogonal ~~matrices~~ vectors.
Since $C = C^\perp$, $G = H$ (G = generator matrix, H = parity matrix of the code).

So if the code is a binary code, it is formed entirely of 0s and 1s.

Weakly self-dual linear binary code : $C \leq C^\perp$

② Cosets : If G is a group and H is a subgroup
$gH = \{gh : h$ an element of $H\}$ is the left coset of H in G with respect to g and
$Hg = \{hg : h$ an element of $H\}$ is the right coset of H in G with respect to g.

ⓐ if H is normal, left coset = right coset

ⓑ $gH \longrightarrow (gH)^{-1} = Hg^{-1}$ is a bijection between the left cosets and the right cosets of H, and their number is same (index of H in G).

— Cosets $C^\perp/C$:— coset of a well defined subgroup in the group. If the condition exists that $C \leq C^\perp$, then my hypothesis is that C shall be the subgroup of $C^\perp$, thus w is the representative of elements of coset of C in $C^\perp$. And there are $N - 2K$ elements in the coset

— The idea is that if G is a group, it is possible to break it into a subgroup H such that pieces of G look like H.

H		bH	
	aH		cH.

G

(62) Let $H < G$, left coset of H in G : $\{gh \mid h \in H\}$ for some $g \in G$. g = representative of coset gH.
Collection of left cosets denoted by G/H.
Collection of right cosets denoted by $H \backslash G$.

- The operation may be multiplication, addition, or anything.

ex $U_{28} = \{1, 3, 5, 9, 11, 13, 15, 17, 19, 23, 25, 27\}$

cyclic subgroup generated by $\langle 9 \rangle = \{1, 9, 25\}$

Left cosets:

$3 \cdot \langle 9 \rangle = \{3, 27, \underset{\uparrow}{75}\}$; $5 \cdot \langle 9 \rangle = \{5, \underset{\uparrow}{45}, \underset{\uparrow}{125}\}$

19 17, 13

Theorem 2: Let d be a minimum distance of the dual code $C^{\perp}$. Then the corresponding quantum code is capable of detecting $<= d-1$ errors, or correcting $\frac{(d-1)}{2}$ errors.

To understand this, some intuition about linear codes is needed.

Linear code – error correcting code for which any linear combination (of codewords) is also a codeword. Proof on page 74 why RS is linear code

Linear code $[N, K]$ = length N and rank K, is a linear subspace C with dimension k of the vector space F_q^n, where F_q is a finite field with q elements. called q-ary code. Codewords = vectors in C and equal q^k.

Distance between two codewords is the Hamming distance between them, or the number of elements in which they differ

Weight = of a codeword is the number of elements that are non-zero.

Minimum distance between codewords = minimum weight of non-zero codewords, or minimum number of non-zero elements in non-zero codewords.

(63) **Obtaining weakly self-dual binary codes from codes over extension fields:**

Field extension is a pair of fields E and F, $E \subseteq F$ such that operations in E are those of F restricted to E.

E is a subfield of F and F is an extension field of E.

Def 3 $C = [N, K, D]$ a linear code of length N, dimension K, and min. distance D over the field F_{2^k} (field with 2^k elements), and let $\beta = (b_1, b_2, \ldots, b_k)$ be a basis of F_{2^k} over F_2. [β = a basis of a field with 2^k elements to a field with 2 elements]

Binary Expansion of C (with respect to the basis β, denoted by $\beta(C)$, is the linear binary code

$$C_2 = [kN, kK, d \geq D]$$

$$C_2 = \beta(C) = \left\{ (c_{ij})_{ij} \in F_2^{kN} \;\middle|\; c = \left(\sum_j c_{ij} b_j \right)_i \in C \right\}$$

Theorem 4. Let $C = [N, K]$ be a linear code over the field F_{2^k} and let $C^\perp$ be its dual. Then

dual of $\beta(C) = \beta^\perp(C^\perp)$

$$\begin{array}{ccc} C & \longrightarrow & C^\perp \\ \text{basis } \beta \downarrow & & \downarrow \text{ dual basis } \beta^\perp \\ \beta(C) & \longrightarrow & \beta^\perp(C^\perp) = \beta(C)^\perp \end{array}$$

Def 5 quantum reed solomon code from classical weakly self-dual RS codes.

Let $C = [N, K, \delta]$ where $N = 2^k - 1$, $K = N - \delta + 1$ and $\delta > \frac{N}{2} + 1$ be a Reed Solomon code over F_{2^k} (with $b = 0$).

(64) The quantum equivalent is found by:

(a) Taking the classical Reed-Solomon code and applying $\beta(C)$, where β is a self-dual basis of F_{2^k} over F_2 (def. 3): or obtaining weakly self-dual binary codes from codes over extension fields. in extension field.

(2) Using definition 1, finding the dual of $\beta(C)$ and the cosets of $\beta(C)$ in its dual, the quantum code can be obtained, $C = [N, N-2K]$.

Mathematics behind this: Initial code $C = [N, K, S]$.

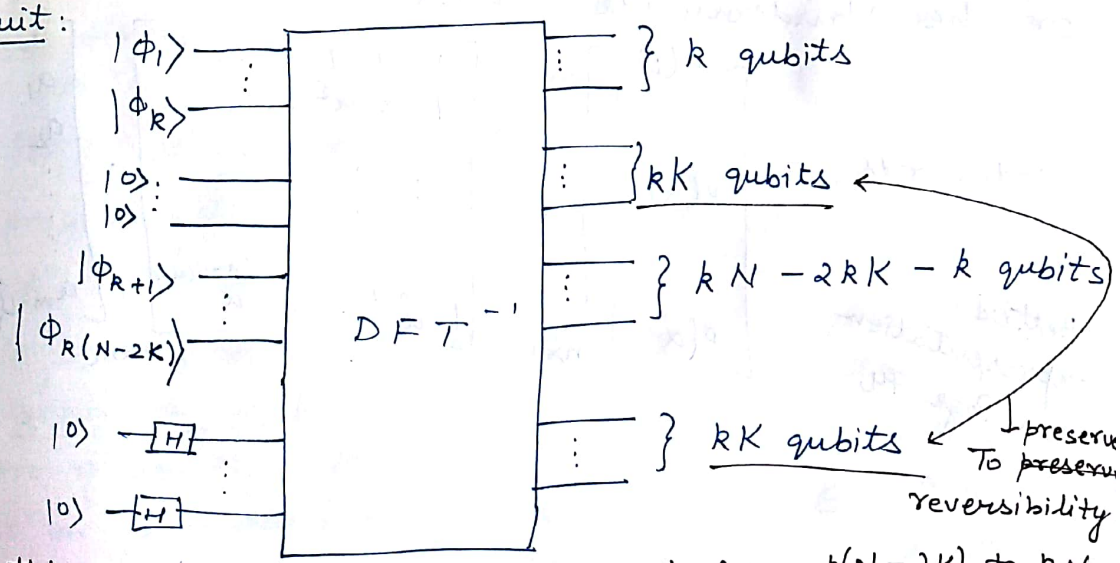
$\beta(C)$ converts it to ~~$\beta(C)$~~ = ~~$[$~~ $[kN, kK, d \geq S]$.

And converting to quantum basis makes it:

$[kN, kN-2kK, d \geq K+1]$ = quantum reed solomon code.

Encodes $kN - 2kK$ qubits using kN qubits. able to detect at least K errors and correct at least $\frac{K}{2}$ errors. This is the quantum RS code because the initial classical code in def 5 was classical RS code $C = [N, K, S]$. The method is applicable to all forms of codes though, it is a general method of obtaining weakly self-dual binary codes from extension fields and then using the coset of $\beta(C)$ in dual to form basis of the quantum code being developed.

Circuit:



Why this architecture – The encoding is from $k(N-2K)$ to kN qubits, where $kN > k(N-2K)$. This adds the need to send $|0\rangle$, numbering $kN - (kN-2K) = 2kK$.

(65) Why half the qubits $|0\rangle$ are in Hadamard?

—

The $k(N-2K)$ qubit input state $|\phi\rangle$ is transformed into a superposition of different cosets of the RS code, determined in the frequency domain.

Notes on classical Reed-Solomon codes:

Galois field: a field of finite elements. G.F.(n) Galois field with n elements; in RS code, n is prime. Elements: $0, \alpha, \alpha^2, \alpha^3, \ldots, \alpha^{n-1}$ such that $\alpha^{n-1} = 1$. (where $\mathbb{N}$ denotes the vector space of the encoding).

Let the information bits be $a_0, a_1, a_2, \ldots, a_{m-1}$. Consider the polynomial:

$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_{m-1} x^{m-1}$$

where $a_i \in GF(n)$.

Generation of codewords:

$$P(1) = a_0 + a_1 + a_2 + \ldots + a_{m-1}$$

$$P(\alpha) = a_0 + a_1 \alpha + a_2 \alpha^2 + \ldots + a_{m-1} \alpha^{m-1}$$

$$\vdots$$

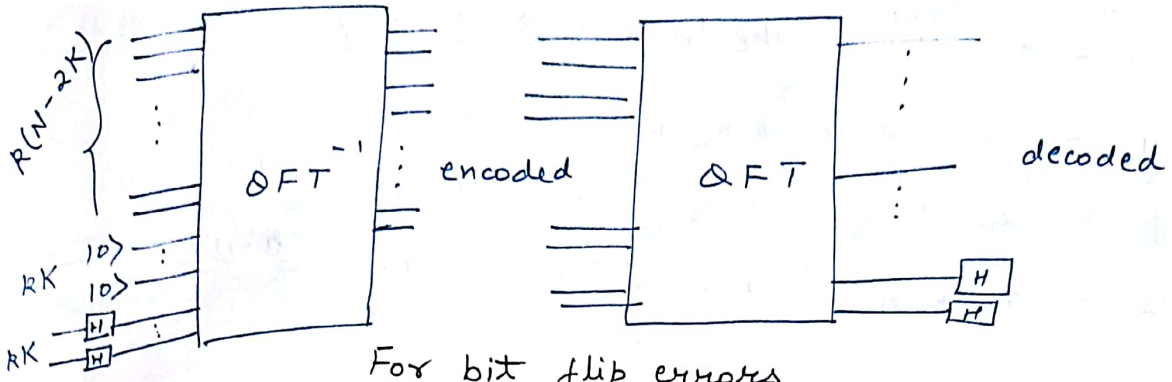
$$P(\alpha^{n-1}) = a_0 + a_1 \alpha^{n-1} + a_2 \alpha^{2(n-1)} + \ldots + a_{m-1} \alpha^{(n-1)(m-1)}$$

or the Vandermonde matrix

$$\text{codewords} = \begin{bmatrix} P(1) \\ \vdots \\ P(\alpha) \\ \vdots \\ P(\alpha^{n-1}) \end{bmatrix}_{n \times 1} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \alpha & \alpha^2 & \ldots & \\ \vdots & \vdots & \vdots & & \\ 1 & \alpha^{n-1} & \ldots & & \alpha^{(n-1)(m-1)} \end{bmatrix}_{n \times m} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{m-1} \end{bmatrix}_{m \times 1}$$

Better representation on Page 74

(66) Encoding and decoding messages with quantum RS code:



For bit flip errors

Recall that bit flip is Pauli X gate, or σ_x
Phase flip errors are actually Pauli Z gate, or σ_z

Note: $H \sigma_x H^{-1} = \sigma_z$ and $H \sigma_z H^{-1} = \sigma_x$

Fourier transform - decomposes a function of time into constituent frequencies.

$f : R \to C :: \hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i x \xi}\, dx$ for any real number ξ

Where $\hat{f}$ is a circumflex f representation.
When independent variable x represents time, transform variable ξ represents frequency. f is determined from $\hat{f}$ by: $f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\, e^{2\pi i x \xi}\, d\xi$ for any real x.

Quantum Fourier transform - a linear transformation on quantum bits, and is the quantum analogue of inverse discrete Fourier transform. (Quantum Fourier Transform) Achieves decomposition into a product of simpler unitary matrices.
The QFT on 2^n amplitudes can be done using $O(n^2)$ Hadamard gate and phase shift gates. Classical takes $O(n 2^n)$.

QFT is discrete classical Fourier transform applied to vector of amplitudes of a quantum state, usually vectors of length $N = 2^n$. The classical transform acts on a vector $(x_0, x_1, x_2, \ldots, x_{N-1}) \in C^N$ and maps it to the vector $(y_0, y_1, y_2, \ldots, y_{N-1}) \in C^N$

$$y_k = \left[\sum_{i=0}^{N-1} x_i \cdot \omega_i^k \right] \cdot \frac{1}{\sqrt{N}} \quad ; \quad k = 0, 1, 2, 3, \ldots, N-1$$

where $\frac{1}{\sqrt{N}}$ is the constant to achieve normalization (but that $\sqrt{N}$ is quantum).

$\omega_n = e^{\frac{-2\pi n i}{N}}$, the **nth root of unity**

$$y_0 = x_0 + x_1 + x_2 + \cdots + x_{n-1}$$
$$y_1 = x_0 + x_1 \cdot \omega_n^{1\cdot 1} + x_2 \cdot \omega_n^{2\cdot 1} + \cdots + x_{n-1} \cdot \omega_n^{(n-1)\cdot 1}$$
$$y_2 = x_0 + x_1 \cdot \omega_n^{1\cdot 2} + x_2 \cdot \omega_n^{2\cdot 2} + \cdots + x_{n-1} \cdot \omega_n^{(n-1)\cdot 2}$$
$$\vdots$$

Matrix for quantum fourier transform:

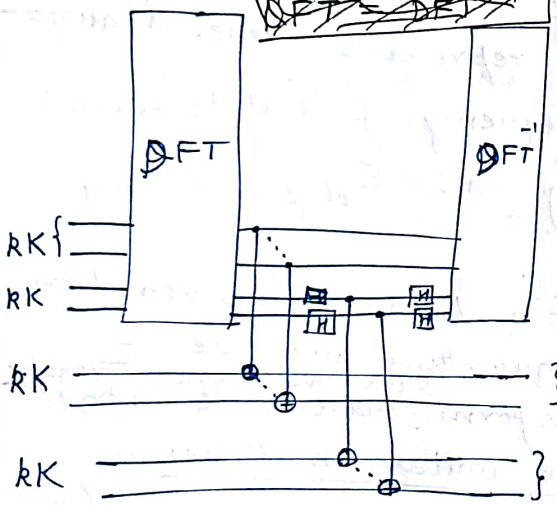
$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n^1 & \omega_n^2 & \cdots & & \\ 1 & \omega_n^2 & \omega_n^4 & \cdots & & \\ 1 & \omega_n^3 & \omega_n^6 & \cdots & & \\ \vdots & & & & & \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & & & \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

where $\omega_n = -\frac{2\pi i}{N} \cdot n$ or nth root of unity

~~QFT = DFT~~ I doubt now: QFT = DFT

Grassl used DFT for encoding

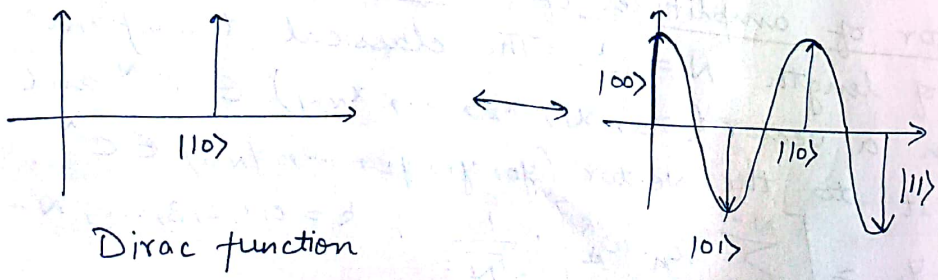
The original Grassl, Geiselmann, and Beth paper uses extra 2kK qubits to prepare syndromes



Circuit implementation: Hadamard gate

$H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ and $R_m := \begin{pmatrix} 1 & 0 \\ 0 & \omega_m \end{pmatrix}$ ← controlled phase gate.

ex $|10\rangle \leftrightarrow |00\rangle - |01\rangle + |10\rangle - |11\rangle$



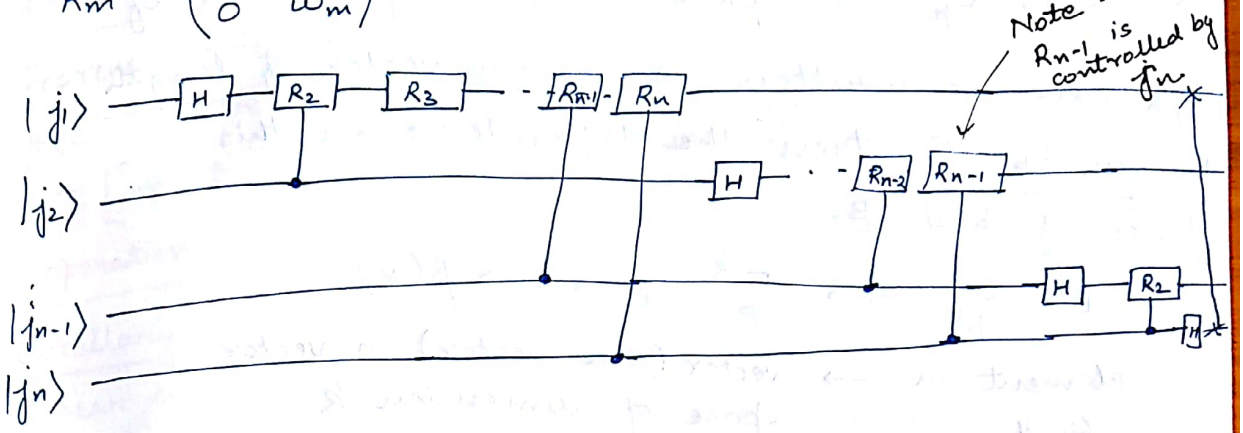
(68) Iterative application of QFT:

$$|j_1 j_2 j_3 \cdots j_n\rangle = \frac{(|0\rangle + e^{2\pi i \cdot \frac{j_1}{2}}|1\rangle)(|0\rangle + e^{2\pi i\left(\frac{j_1}{2} + \frac{j_2}{4}\right)}|1\rangle)}{\sqrt{2}}$$

$$\cdots \frac{|0\rangle + e^{2\pi i\left(\frac{j_1}{2} + \frac{j_2}{4} + \cdots + \frac{j_n}{2^n}\right)}|1\rangle}{\sqrt{2}}$$

In the end swap the first and last, second and second last and so on.

$$R_m := \begin{pmatrix} 1 & 0 \\ 0 & \omega_m \end{pmatrix} \qquad \omega_m = -\frac{2\pi i \cdot m}{2^m}$$



Points:

① each qubit first gives the requisite phase ~~chase~~ change to the above qubit

② it then Hadamards ③ then receives phase change from lower qubits.

Qubit 1 $|j_1\rangle$

① $H \rightarrow \frac{1}{\sqrt{2}}[|0\rangle + |1\rangle] = \frac{1}{\sqrt{2}}\left[|0\rangle + e^{2\pi i\left(\frac{j_1}{2}\right)}|1\rangle\right]$

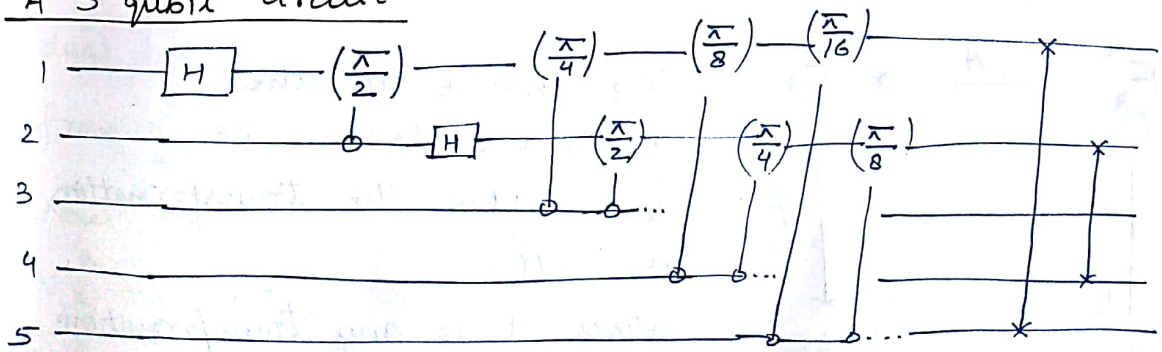
② $R_2 \rightarrow \frac{1}{\sqrt{2}}\left[|0\rangle + e^{2\pi i\left\{\frac{j_1}{2} + \frac{j_2}{4}\right\}}|1\rangle\right]$

⋮

③ $R_n \rightarrow \frac{1}{\sqrt{2}}\left[|0\rangle + e^{2\pi i\left\{\frac{j_1}{2} + \frac{j_2}{4} + \frac{j_3}{8} + \cdots + \frac{j_n}{2^n}\right\}}|1\rangle\right]$

where $e^{i\phi} = \cos\phi + i\sin\phi$

A 5 qubit circuit –



(69) Qubit 1 : $\frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\left(\frac{\pi}{2}+\frac{\pi}{4}+\frac{\pi}{8}+\frac{\pi}{16}\right)}|1\rangle\right)$ and so on.

More ideas on Grassl, Geiselmann, Beth —

Binary codes from codes over F_{2^k} :

① Any finite field of characteristic p i.e. a finite field F_q where $q = p^k$ is a vector space of dimension k over F_p. For a fixed basis β of F_q over F_p (or F_{p^k} over F_p), any element of F_{p^k} can be written as a row vector of length k over F_p. To stress the dependence on this choice of basis β,

$$\beta : F_{p^k} \longrightarrow F_p^k \; ; \; \alpha \longmapsto \beta(\alpha)$$

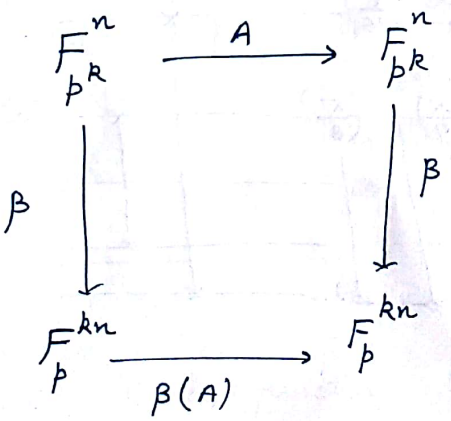
element in field → vector (row vector) in vector space of dimension k

② Dual basis: Given a basis $\beta = (b_1, \ldots, b_k)$ of a finite field F_q over F_p, the dual basis of β is a basis $\beta^\perp = (b_1', \ldots, b_k')$

$$\forall i, j : tr(b_i b_j') = \delta_{ij}$$ → Kroneckar delta symbol.

or the set of vectors that are orthogonal to β, linearly independent, and necessarily spanning the dual vector space.

– existence of unique dual basis to every basis β. For fields with characteristic two, there exist a self-dual basis



The change to the ground state can be done after the transformation as well.
where A is any transformation

(70) **Binary expansion**: For every c in $F_{2^k}^N$ and a basis $\beta = \{b_1, b_2, \ldots, b_k\}$,

$c = \alpha_1 b_1 + \alpha_2 b_2 + \ldots + \alpha_k \cdot b_k$ where $\alpha_i \in F_2^{kN}$

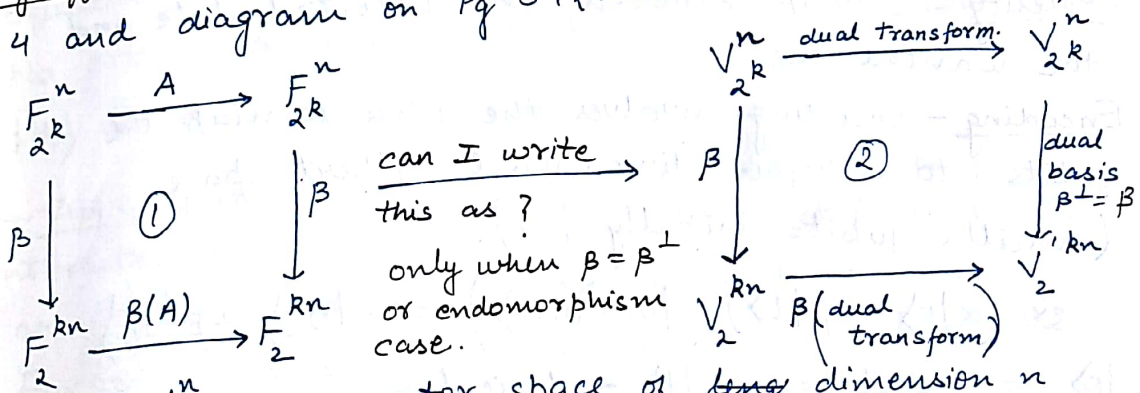
Thus an element of $F_{2^k}^N$ can be written as a vector in F_2^{Nk}. or $c \longrightarrow [\alpha_1, \alpha_2, \alpha_3 \ldots, \alpha_k]$

The above ideas should be able to fully describe #Def 3 on Page 63. Although dimension of C_2 is still mathematically not coming, for me at least.

Theorem 4 extension - Page 63 — dual of the binary expansion $\beta(c)$ is the binary expansion of $c^\perp$ with respect to $\beta^\perp$ [or $\beta^\perp(c^\perp)$] — I can't understand the trace operation.

Corollary 1: If $C = [N, K]$ be a weakly self-dual linear code over field F_{2^k}, then $\beta(c)$ where β is a self-dual basis is also a weakly self-dual linear code.

My effort to explain - using prior knowledge of theorem 4 and diagram on Pg 69 (last section).



where $V_{2^k}^n$ is a vector space of ~~long~~ dimension n over F_{2^k} and $V_{2^k}^{'n}$ is the dual vector space. β is the basis of $V_{2^k}^n$ over V_2^{kn} and $\beta^\perp$ is the corresponding dual basis ($V_{2^k}^{'n}$ over $V_2^{'kn}$). Note the dual transform from $V_{2^k}^n$ to V_2^{kn} is also transformed as β(dual transform).

Now self-dual basis $\Rightarrow \beta = \beta^\perp$. In figure 2, $V_{2^k}^n$ and $V_{2^k}^{'n}$ may be different vector spaces.

If endomorphism occurs ($\beta = \beta^\perp$) and if $V_{2^k}^n = V_{2^k}^{'n}$ (when the code is self dual), one can conclude that corollary 1 is correct.

(71) In a n-qubit register, operation on jth qubit given by $I_{2^{j-1}} \otimes U_2 \otimes I_{2^{n-j}}$ where U_2 is a 2×2 mat

---

Notes on Theory of quantum error-correcting codes (Knill and Laflamme)

- referenced under Grassl, Geiselmann, and Beth
- based on encoding states into larger Hilbert spaces subject to known interactions.
- decoherence with the environment (or all degrees of freedom that can have unwanted interactions with the computer.
- because of the inadequacy in isolating the ~~informa~~ state from the environment.
- the no-cloning theorem (Wootters and Zurek) prevents the redundant approach to error correction.
- it is possible to correct a state against certain known ~~st~~ errors by spreading the information over several qubits (Shor).
- Goal: find an encoding which behaves in a desired ~~beg~~ under evolution by an interaction superoperator.
- Fidelity - overlap between the corrected state and the wanted one.

Encoding - encoding involves the idea to map the qubits into a higher dimensional Hilbert space (ancilla qubits initially $|0\rangle$).

ex $(\alpha|0\rangle + \beta|1\rangle)(|0000\ldots\rangle) = \alpha|0_L\rangle + \beta|1_L\rangle$

$|0\rangle_L \equiv$ logical zero ; $|1_L\rangle \equiv$ logical one.

$\alpha|0_L\rangle + \beta|1_L\rangle \equiv$ such that the errors map it into a family of 2D subspaces which preserve the ~~the~~ relative coherence of quantum information (or in each subspace, the state of the computer is a tensor product with the environment) or the state is not entangled, else retrieval is impossible.

Performing a measurement maps this state back to a 2D state, and the outcome determines a unitary transformation that can convert this 2D state to the original state (the essence of quantum correcting codes).

(72) How the environment interacts with the ~~or~~ system:

Let the initial state be ψ_i. Assuming the environment is not initially entangled, the reduced density form of the system: $\rho_f = \sum_a A_a \rho_i A_a^\dagger$ where ρ_i is the ~~initial~~ state and A_i are some operators (Kraus; States, Effect, and operations), where A_a is determined from: $A_a = \langle \mu_a | U | e \rangle$; where

$|\mu_a\rangle$ = orthonormal basis of the environment

$|e\rangle$ = environment's initial state; U = evolution operator of the system. Adjoint = complex conjugate and transpose

Now, ~~$A_a^\dagger = \langle e | U^\dagger | \mu_a \rangle$. Then $\sum_a A_a^\dagger A_a = \sum_a \langle e | U^\dagger | \mu_a \rangle \langle \mu_a | U | e \rangle$~~

$$\sum_a A_a^\dagger A = \sum_a \langle e | U^* | \mu_a \rangle^* \langle \mu_a | U | e \rangle = I$$ ← Messed up things.

A_a: linear operators of the Hilbert system of the environment, called interaction operators. Any family of A_a satisfying $\sum_a A_a^\dagger A$ = superoperator. These are not unique though, their choice depends on $|\mu_a\rangle$. If different $|\mu_a\rangle$ yield different A_a, they are physically equivalent.

No prior knowledge of A_a corrupting $_L \Rightarrow$ inability to recover $|\psi_i\rangle$ consistently

— Approximation — for a system of qubits, the interaction is independent of each qubit. Thus, A_a are tensor products of one qubit interactions. For small error rates, the one qubit interaction operators are near I.

Necessary and sufficient conditions to recover $|\psi_i\rangle$ ::

① $\langle 0_L | A_a^\dagger A_b | 1_L \rangle = 0$ or $|0_L\rangle$ and $|1_L\rangle$ are orthogonal (under any error)

② $\langle 0_L | A_a^\dagger A_b | 0_L \rangle = \langle 1_L | A_a^\dagger A_b | 1_L \rangle$

— length and inner products of projections of corrupted $|0_L\rangle$ and $|1_L\rangle$ should same

→ sufficient but not necessary — equation is 0 if A_a and A_b are different, implies each error maps the state to orthogonal subspaces, permitting retrieval by projecting on these subspaces.

Most probably: retreiving $|\psi_i\rangle \Rightarrow$ equation is 0 when $a \neq b$ or orthogonal subspaces.

(73) Detailing some mathematics in this:-

① $\langle 0_L | A_a^\dagger A_b | 1_L \rangle = (A_a | 0_L \rangle)^\dagger (A_b | 1_L \rangle)$ or the inner product of erroraneous transformation on $|0_L\rangle$ and $|1_L\rangle$.

② $\langle 0_L | A_a^\dagger A_b | 0_L \rangle$ = length and inner product of projections of corrupted zeros

— Realistic quantum computers can have only a fraction of errors corrected.

Fidelity - overlap between final state ρ_f and $|\psi_i\rangle$

A = combined superoperator consisting of an environment interaction and recovery operation

$$F(|\psi_i\rangle, A) = \sum_a \langle \psi_i | A_a | \psi_i \rangle \langle \psi_i | A_a^\dagger | \psi_i \rangle$$

depends on the choice of $|\psi_i\rangle$. Best quantum code minimizes F_{min}.

---

investigation of decoherence: randomisation of phase of initial system. $|\psi_i\rangle$.

Decoherence ~~minimizes the~~ decreases the size of the diagonal element of the density matrix in a basis determined by the interaction Hamiltonian with the environment.

Note on density matrix: for a finite dimensional function space, most general density operator of form - $\rho = \sum_j p_j |\psi_j\rangle\langle\psi_j|$, $|\psi_j\rangle\langle\psi_j|$ = outer product notation.

p_j = probability that system is in pure state $|\psi_j\rangle$

Notes on Gao et. al. A new (classical) algorithm for decoding solomon-reed codes.

— uses Fast Fourier Transforms to compute the message symbols directly without explicit finding of error locations or error ~~symbols~~ magnitudes

— Main advantage of RS codes —

ⓐ high capability of correcting both random and burst errors

ⓑ existence of efficient decoding algorithms for them — Berlekamp-Massey (first syndromes and the

error locations and error magnitudes. $O(n^2)$ and efficient in practice.

- New algo. doesn't explicitly finds the error locations and error magnitudes.
- Operations: Interpolation, partial gcd, long division (all implementable by fast algorithms using Fast Fourier transforms)

Encoding Reed-Solomon codes - q = prime; F_q = finite field with q elements, such that $1 \le k < n \le q$.

To encode information of k symbols $(m_1, m_2, \ldots, m_k)$, use of the following polynomial -

$$f(x) = m_1 + m_2 x + m_3 x^2 + \ldots + m_k x^{k-1}.$$

For a codeword of length n, choose any n elements from F_q. Then codeword $c = (c_1, c_2, c_3, \ldots, c_n)$ such that:

~~$c_1 = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \ldots + \alpha_k x^{k-1}$~~

$$c_1 = m_1 + m_2 \alpha_1 + m_3 \alpha_1^2 + \ldots + m_k \alpha_1^{k-1}$$
$$c_2 = m_1 + m_2 \alpha_2 + m_3 \alpha_2^2 + \ldots + m_k \alpha_2^{k-1}$$
$$\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots$$
$$c_n = m_1 + m_2 \alpha_n + m_3 \alpha_n^2 + \ldots + m_k \alpha_n^{k-1}$$

Vandermonde matrix

$$c = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \alpha_1^3 & \ldots & \alpha_1^{k-1} \\ 1 & \alpha_2 & \alpha_2^2 & \alpha_2^3 & \ldots & \alpha_2^{k-1} \\ & & \vdots & & & \\ 1 & \alpha_n & \alpha_n^2 & \alpha_n^3 & \ldots & \alpha_n^{k-1} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ \vdots \\ m_k \end{bmatrix}$$

Generator matrix for classical RS code.

Hence the code is a mapping from a m-tuple to q^k tuple $(f(0), f(\alpha), f(\alpha^2), \ldots, f(1))$

These codewords forms a linear code over F_q, linear code $\Rightarrow$ error correcting code for which any linear combinations of codewords is also a codeword.

Determinant of Vandermonde matrix $= \prod_{1 \le i < j \le k} (\alpha_j - \alpha_i)$ when $n = k$.

Square matrix $\begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \alpha_1^3 & \ldots & \alpha_1^{k-1} \\ 1 & \alpha_2 & \alpha_2^2 & \alpha_2^3 & \ldots & \alpha_2^{k-1} \\ & & \vdots & & & \\ 1 & \alpha_k & \alpha_k^2 & \alpha_k^3 & \ldots & \alpha_k^{k-1} \end{bmatrix}$

→ since none of $\alpha_j = \alpha_i$ (they were distinct to begin with), this is non-zero and thus these k codewords are linearly independent, making this a linear code.

Any of the k combinations can be chosen in this way, and therefore the entire code is linear (or linear combination of independent vectors).

- modern methods use Gorenstein and Zieler's [9] note on RS codes that ~~are~~ they are cyclic codes (special) and can be encoded via a generator polynomial. But FFTs can easily implement the above method.

- Min. dist. $d = n - k + 1$ is the best possible for any (n, k) linear code.

Summary up to this point: To encode $(m_1, m_2, \ldots, m_k)$ in a higher dimensional space, we selected n elements from F_q ($n \leq q$) and applied $f(x)$ on it. These n elements may be represented as α_i or a_i, $\forall i \in \{1, 2, 3, \ldots, n\}$.

Decoding RS code: Suppose the encoded code was $C = (c_1, c_2, c_3, \ldots c_n)$ where $c_i = f(a_i)$. Let ~~be~~ b be a received word $B = (b_1, b_2, \ldots, b_n)$ which comes from C with t errors, where $t \leq \frac{d-1}{2}$. Goal is to find the message polynomial $f(x)$ that defined C originally.

Precompute the polynomial:

$$g_0 = \prod_{i=1}^{n} (x - a_i) \in F_q[x]$$

g_0 is known ~~for~~ for many cases, like $n = q$ or $n \mid q-1$ and so on.

Interpolation - find unique polynomial $g_1(x) \in F_q[x]$ of degree $\leq n-1$ such that

$$g_1(a_i) = b_i \; ; \; 1 \leq i \leq n$$

Partial gcd - Apply extended Euclidean algorithm to $g_0(x)$ and $g_1(x)$. Stop when remainder $g(x)$ has degree $< \frac{1}{2}(n+k)$.

$$u(x)\, g_0(x) + v(x) \cdot g_1(x) = g(x)$$

Long division - divides $g(x)$ by $v(x) \Rightarrow g(x) = f_1(x) \cdot v(x) +$ degree $r(x) <$ deg. $v(x)$.

(76) If $f_1(x)$ has degree $< k$ and $r(x) = 0$, output $f_1(x) \cdot v(x)$ = error locator polynomial whose roots contain all positions a_i where errors have occured.

Fast Fourier transforms –

For any fixed distinct points $a_1, a_2, \ldots, a_n \in F_q$, transform from $f \in F_q[x]$ to $f(a_1), f(a_2) \ldots, f(a_n)$ is a discrete fourier transform

$$DFT(f_0, f_1, \ldots, f_{n-1}) = \left(f(a_1), f(a_2), \ldots, f(a_n)\right).$$

where $f = f_0 + f_1 x + f_2 x^2 + \ldots + f_{n-1} x^{n-1}$

Interpolation is the inverse discrete fourier transform DFT^{-1}. → Did the coefficients transform to the values of the fn.

From Singhal 1992 (Interpolation using FFTs) –

Consider a function $g(s)$ of variable s. If $g(s)$ can be represented as a Taylor series

$$g(s) = \sum_{0}^{\infty} a_m s^m \text{ with real coefficients, it is}$$

truncated at some point to obtain a polynomial approximation valid in a region in the neighbourhood. Method below – generates a_m; applicable when $g(s)$ can be evaluated at complex points on a unit circle.

Let $g(s)$ be evaluated at $M = N + 1$ points equally spaced on the unit circle. Then,

$$g(s_\mu) = \sum_{m=0}^{N} \hat{a}_m W^{m\mu} \; ; \; \mu = 0, 1, 2, \ldots, N$$

with $W = \exp(2\pi i / M)$; $i = \sqrt{-1}$

and $\hat{a}_m = a_m + \sum_{l=1}^{\infty} a_m + Ml$

→ aliasing error in the coefficient a_m.

$g(s)$ is known at M points; we use $M-1$ of them denoted by $s_0, s_1, s_2, \ldots, s_N$. Values known are denoted by $g(s_0), g(s_1), g(s_2), \ldots, g(s_N)$. The equation: $g(s_N) = \sum_{m=0}^{\infty} \hat{a}_m . W^{N.m}$ gives the value of a_N (with some error – aliasing error).

Truncated approximation will then be – $a_0 s^0 + a_1 s^1 + \ldots + a_n s^N$.

Now how to ensure $g(a_i) = b_i$

(76) Notes from Optimal Quantum Algorithm for polynomial interpolation – Andrew Childs et. al.

Nothing. I can't understand it yet.

Notes from An improvement in Quantum Fourier Transform – Laszlo Gyongyosi and Sandor Imre.

- uses quantum SVD (singular value decomposition) decomposes rectangular real or complex matrix into 2 orthogonal matrix and one diagonal matrix
- Overview of quantum information processing – $|0\rangle$ and $|1\rangle$ – Dirac notation. $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ where α and β are complex numbers. $|\psi\rangle$ is a vector in 2D complex space where $|0\rangle$ and $|1\rangle$ form an orthonormal basis.

Norm of $|\psi\rangle \equiv \|\psi\| = \sqrt{|\alpha|^2 + |\beta|^2} = 1$ as $|\alpha|^2 + |\beta|^2 = 1$ (Normalization condition).

Unitary transformation: $U^\dagger U = U U^\dagger = I$ where U is the unitary transformation.

Tensor product – If we have vector spaces W and U with dimensions m and n, then tensor products tend to map to a vector space of dimension mn. $(W \otimes U)$.

- Any element in $W \otimes U$ is a linear combination of $|w\rangle \otimes |v\rangle$. For $x \in C$,

① $x(|w\rangle \otimes |v\rangle) = x|w\rangle \otimes |v\rangle = |w\rangle \otimes x|v\rangle$

② $(|w_1\rangle + |w_2\rangle) \otimes |v\rangle = (|w_1\rangle \otimes |v\rangle) + (|w_2\rangle \otimes |v\rangle)$

③ $|w\rangle \otimes (|v_1\rangle + |v_2\rangle) = (|w\rangle \otimes |v_1\rangle) + (|w\rangle \otimes |v_2\rangle)$

④ Non-commutative or order preserving

Linear operator – If two linear operators A and B on W and U, then $A \otimes B$ on $W \otimes U$ is defined as: $(A \otimes B)(|w\rangle \otimes |u\rangle) = A|w\rangle \otimes B|u\rangle$

$$A \otimes B = \begin{bmatrix} A_{11}B & \cdots & A_{1n}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \cdots & A_{nn}B \end{bmatrix}$$

If A is $m \times m$ and B is $n \times n$, then $A \otimes B$ becomes $mn \times mn$.

(77) n-qubit quantum register – superposition of 2^n basis states – $|0\rangle, |1\rangle, |2\rangle, \ldots, |2^n - 1\rangle$

$|\psi\rangle = \sum_{i=1}^{2^n-1} \alpha_i |i\rangle$, such that $\sum_{i=1}^{2^n-1} |\alpha_i|^2 = 1$ where α_i are respective amplitudes.

– reason why linear increase in number of qubits causes exponential increase in no. of qubits.

Hilbert space – Any complex vector space having an inner product such that:

① $\langle\psi||\phi\rangle = \langle\phi||\psi\rangle^*$ ② $\langle\psi|(a|u\rangle + b|v\rangle) = a\langle\psi||u\rangle + b\langle\psi||v\rangle$

③ $\langle\phi||\phi\rangle$ = essentially square of distance from origin > 0 if $\phi \neq 0$

dual – linear operator from a vector space to the complex numbers $\langle\phi|(|v\rangle) = \langle\phi|v\rangle$

outer product – $|\psi\rangle\langle\phi|$ satisfying $(|\psi\rangle\langle\phi|)|u\rangle = |\psi\rangle\langle\phi|u\rangle = (|\psi\rangle \otimes \langle\phi|)|u\rangle$

If $|\psi\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$ and $|\phi\rangle = \begin{bmatrix} c \\ d \end{bmatrix}$, then $|\psi\rangle\langle\phi| = |\psi\rangle \otimes \langle\phi|$

$= \begin{bmatrix} a \\ b \end{bmatrix} [c^* \quad d^*] = \begin{bmatrix} ac^* & ad^* \\ bc^* & bd^* \end{bmatrix}$

Quantum SVD approach – same complexity as QFT $[O(N \log N)]$ but more precise approximations.

Main task – find and retain the singular values – retain big coefficients – to eliminate input's redundancy.

Classical procedure of SVD –

① find singular values σ_i by finding eigenvalues of AA^T and square rooting them

ex if $A = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix}$; $AA^T = \begin{bmatrix} 17 & 8 \\ 8 & 17 \end{bmatrix} \Rightarrow \lambda = 9, 25, 0$

$\sigma_1 = 5, \sigma_2 = 3$ (singular values)

② SVD of $A = U \Sigma V^T$. Now find right singular vectors (columns of V) = orthonormal set of eigenvectors of $A^T A$

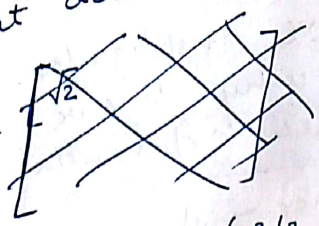
$\lambda = 25$

$\begin{pmatrix} -12 & 12 & 2 \\ 12 & -12 & -2 \\ 2 & -2 & -17 \end{pmatrix}$ | $\lambda = 9$ $\begin{pmatrix} 4 & 12 & 2 \\ 12 & 4 & -2 \\ 2 & -2 & -1 \end{pmatrix}$

(78) Row reduced – $\begin{pmatrix} 1 & 0 & -1/4 \\ 0 & 1 & 1/4 \\ 0 & 0 & 0 \end{pmatrix}$ and $\begin{pmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

Unit length vectors that describe the matrices

$\begin{pmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \equiv^{v_1} \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{pmatrix}$ = normalized



$\begin{pmatrix} 1 & 0 & -1/4 \\ 0 & 1 & 1/4 \\ 0 & 0 & 0 \end{pmatrix} \overset{v_2}{\equiv} \begin{pmatrix} 1/\sqrt{18} \\ -1/\sqrt{18} \\ 4/\sqrt{18} \end{pmatrix}$ and $v_3 = \begin{pmatrix} 2/3 \\ -2/3 \\ -1/3 \end{pmatrix}$

So $V = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{18} & 2/3 \\ 1/\sqrt{2} & -1/\sqrt{18} & -2/3 \\ 0 & 4/\sqrt{18} & -1/3 \end{bmatrix}$

$U \equiv u_i = \frac{1}{\sigma_i} A v_i$ where i corresponds to the specific eigenvalues! singular values.

$A = U \Sigma V^T$

$= \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{18} & -1/\sqrt{18} & 4/\sqrt{18} \\ 2/3 & -2/3 & -1/3 \end{pmatrix}$

Back to quantum SVD – computation of coefficient vector requires $O(LN)$ :: L = number of non-zero singular values and coefficient vector = the N-length vector that approximates the polynomial.
– since the input usually has high redundancy, $L \ll N$ is often satisfied.

Quantum SVD shall use quantum searching to achieve increase in precision. Our purpose is to find the space generated by basis vectors that achieves better approximations.

Quantum SVD :: $B = USV^+$ such that S is a diagonal element, U and V are unitary matrices Unitarity is the quantum analogue of orthogonality. such that $A^+A = I$ if A is unitary and complex.

(19) Columns of U are BB^T eigenvectors and rows of V are B^TB eigenvectors. Diagonal elements of S are the singular values of BB^T, square roots of eigenvalues of BB^T or B^TB (both are equal). Essence of singular value decomposition.

- Quantum SVD can process long sequences at one time. Classical SVD needs many small sections to work on

→ expansion of original data in an orthogonal basis:

$$x_{ij} = \sum_k b_{ik} v_{jk} = \sum_k u_{ik} s_k v_{jk} = \sum_k c_{ik} e^{\frac{i 2\pi jk}{2}}$$

$v_k = e^{i 2\pi jk/2}$ is the normalized vector with unit length.

This section needs a detailed re-read.

Basic notations and operations –

The quantum fourier transform is an unitary operation on n b qubits, given by: $N = 2^n$

$$\text{QFT } |x\rangle : \quad F|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{i2\pi}{N} \cdot xy} |y\rangle$$

Let me decode this a bit:

Probably right.

Wrong idea.

Because $|x\rangle$ is a single column vector

$$F|0\rangle = e^0|0\rangle + e^0|1\rangle + \dots + e^0|N-1\rangle$$

$$F|1\rangle = e^{\frac{i2\pi}{N}\cdot 0}|0\rangle + e^{\frac{i2\pi}{N}\cdot 1}|1\rangle + e^{\frac{i\cdot 2\pi}{N}\cdot 2}|2\rangle + \dots$$

$$\vdots$$

$$F|N-1\rangle = e^{i\frac{2\pi}{N}(N-1).0}|0\rangle + e^{\frac{i\cdot 2\pi}{N}(N-1)\cdot 1}|1\rangle + \dots\dots$$

$|x\rangle = |x_0\rangle \otimes |x_1\rangle \otimes |x_2\rangle \dots \otimes |x_{n-1}\rangle$ and y similarly.

So having 2 vectors being the tensor product of n qubits, the Quantum fourier transform yields:

$$= \frac{1}{\sqrt{N}} \left(e^{\frac{i\cdot 2\pi}{N} \cdot x \cdot y} \dots \right)$$

What exactly are x and y?

– Actually dummy variables that can be changed to any other variables. Better to refer to next page for better understanding

Better method could be:

$$|x_0\rangle = \frac{1}{\sqrt{N}} (\dots) = F|0\rangle \text{ ; and so on. And thus}$$

$$|x\rangle = |x_{n-1}\rangle \otimes |x_{n-2}\rangle \otimes \dots \otimes |x_1\rangle \otimes |x_0\rangle$$

(80) Easier version of quantum fourier transform -

DFT: acts on a vector $X = (x_0, x_1, \dots, x_{N-1}) \in \mathbb{C}^N$ and maps to $Y = (y_0, y_1, \dots, y_{N-1}) \in \mathbb{C}^N$.

$$y_k = \left(\sum_{n=0}^{N-1} x_n \omega_n^k \right) \frac{1}{\sqrt{N}} \quad \text{or:} \quad y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \cdot e^{\frac{-2\pi i}{N} \cdot nk}$$

So there are basically 2 elements here - n and k such that k controls y_k and n controls the list of x_n for each y_k.

Such that:

$$y_0 = \frac{1}{\sqrt{N}} \left(x_0 + x_1 + x_2 + \dots + x_{N-1} \right)$$

$$y_1 = \frac{1}{\sqrt{N}} \left(x_0 \omega_0^1 + x_1 \omega_1^1 + x_2 \omega_2^1 + \dots + x_{N-1} \omega_{N-1}^1 \right)$$

Quantum equivalent:

Just try to think of $x_0, x_1, \dots, x_{N-1}$ as coefficients of basis vectors in a N-dimensional Hilbert space

Then; $|x\rangle = \sum_{j=0}^{N-1} x_j |i\rangle$ is mapped to $|y\rangle = \sum_{j=0}^{N-1} y_j |i\rangle$

Such that

$$y_j = \left(\sum_{k=0}^{N-1} x_k e^{\frac{-2\pi i}{N} \cdot j \cdot k} \right) \frac{1}{\sqrt{N}}$$ where y_j and x_j are the coefficients of the basis vectors making up $|x\rangle$ and $|y\rangle$.

Inverse quantum fourier transform -

The classical inverse fourier transform can be given by; considering 2 vectors $X = (x_0, x_1, x_2, \dots, x_n)$ and $Y = (y_0, y_1, y_2, \dots, y_n)$:

$$x_k = \left(\sum_{j=0}^{n-1} y_j e^{\frac{-2\pi i(-k)}{N} j} \right) \frac{1}{\sqrt{N}}$$

Such that there is some function $f(x_k) = y_k$

Function to values = DFT

Values to function = DFT^{-1}

Inverse quantum fourier transform:

Having 2 vectors $|y\rangle$ and $|x\rangle$ in a n-dimensional Hilbert space, an IQFT can be applied on the amplitudes of $|y\rangle$:

$$x_j = \sum_{n=0}^{N-1} e^{\frac{-2\pi i(-j)n}{N}} y_k$$ inverted sign.

Basic idea is to be able to depict one amplitude as some combination of all other amplitudes.

(11) The obtained thing is a Fourier basis.

- Some thoughts I have before moving (finally) to SVD: if DFT can map from a function to its value

Gao et. al Pg 76, recall the Solomon property:

$f(x) = m_0 + m_1 x + m_2 x^2 + \dots + m_k \cdot x^{k-1}$ to encode $(m_0, m_1, m_2, \dots m_k)$. We picked any $n (\leq q)$ elements from the Galois field F_q where q was prime, The encoding came out as $C = (c_1, c_2, c_3, \dots, c_n)$ such that

$$\begin{aligned} c_1 &= m_0 + m_1 \alpha_1 + m_2 \alpha_1^2 + \dots + m_k \alpha_1^{k-1} \\ &\vdots \\ c_n &= m_0 + m_1 \alpha_n + m_2 \alpha_n^2 + \dots + m_k \alpha_n^{k-1} \end{aligned}$$

where $\alpha_1, \alpha_2, \dots, \alpha_n$ are the n elements we chose.

So where does DFT fit into this? And more importantly, QDFT?

So, according to Gao et. al., α_1 can be mapped to $f(\alpha_1)$ by doing something like:

$$m_1 = f(\alpha_1) = \sum_{p=1}^{n} \alpha_p e^{-\frac{2\pi i}{N} \cdot 1 \cdot p}$$

I feel some problem (as of June 15, 2019). Gao et. al. Pg 76 talks about $(f_0, f_1 \dots) \to (f(a_1), f(a_2)$ where f_0, f_1 are coefficients. So the coefficients take on new values.

And in the quantum form, we are transforming to the Fourier basis by applying Fourier transform on the amplitudes of the basis vectors. And basis vectors were from the coset def. from Page 60.

If F be the quantum fourier transform and its effect is known on the basis vectors, then its effect on any vector can be determined.