Introduction to: Quantum Computing For Computer Scientists

James Oswald

IEEE

UAlbany Student Branch

Information about the talk

- The talk was originally given at Microsoft in 2018 by Andrew Helwer
- The talk covers a basic introduction to quantum computing and the problems it can solve.
  - Representing computation with basic linear algebra (matrices and vectors)
  - The computational workings of qbits, superposition, and quantum logic gates
  - Solving the Deutsch oracle problem: the simplest problem where a quantum computer outperforms classical methods
  - quantum entanglement and teleportation
  - A live demonstration of quantum entanglement on a real-world quantum computer
  - A demo of the Deutsch oracle problem implemented in Q# with the Microsoft Quantum Development Kit

Linear Algebra Recap

It’ll be quick I promise.

Linear Algebra Review : Vectors

- Vectors are mathematical objects representing lists of numbers, but can also be thought of from a physics perspective as a quantity with direction and magnitude. Both perspectives represent the same abstract idea.

1x2 Row Vector: $\begin{pmatrix} 12 & 3 \end{pmatrix}$

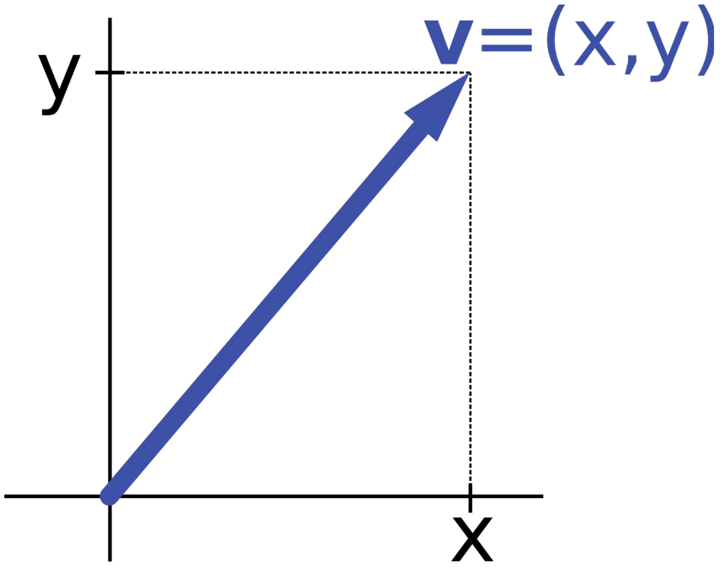
1x3 Row Vector: $\begin{pmatrix} 3 & 0 & 7 \end{pmatrix}$

1xn Row Vector: $\begin{pmatrix} a_1 & a_2 & \cdots & a_n \end{pmatrix}$

2x1 Column Vector: $\begin{pmatrix} 23 \\ 7 \end{pmatrix}$

3x1 Column Vector: $\begin{pmatrix} 7 \\ 8 \\ 1 \end{pmatrix}$

nx1 Column Vector: $\begin{pmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_n \end{pmatrix}$



What we'll be using vectors for today?

- Representing bits, qbits, individual state representations of multiple qbits (It'll make sense later I promise). In a quantum computer we represent our classical bits as vectors of the form:

One bit with the value 0, also written as $|0\rangle$ (Dirac vector notation)

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

One bit with the value 1, also written as $|1\rangle$

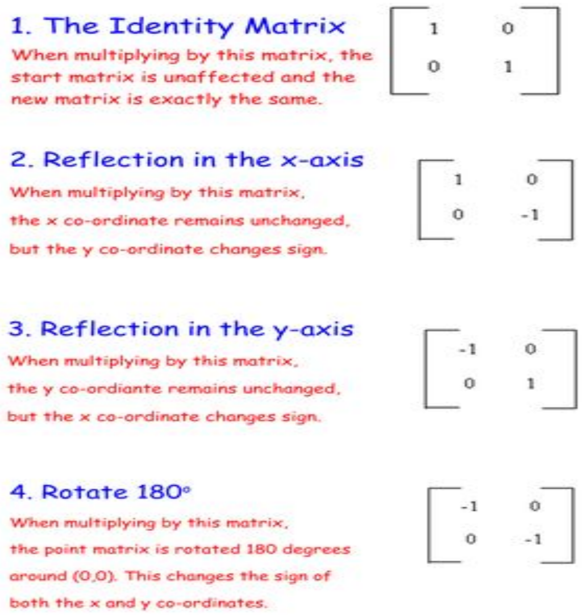
$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

*We will build to representing multiple bits as tensor products of single bits that are represented by 2^n tall column vectors.

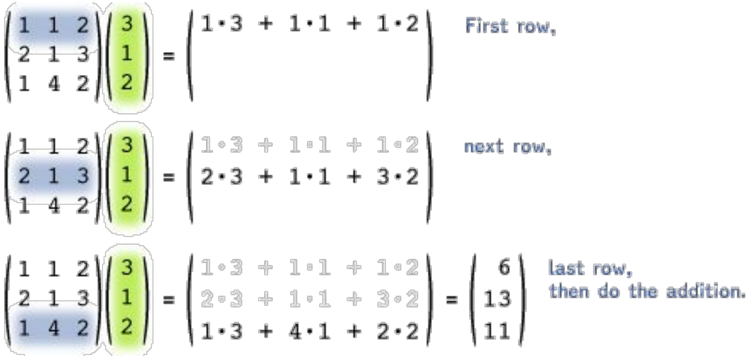
$$|4\rangle = |100\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Linear Algebra Review : Matrices

- Matrices 2d “grids” of numbers that can be used as “transformations” when applied to vectors using matrix multiplication.
- Simple examples of matrices transforming 2d vectors.

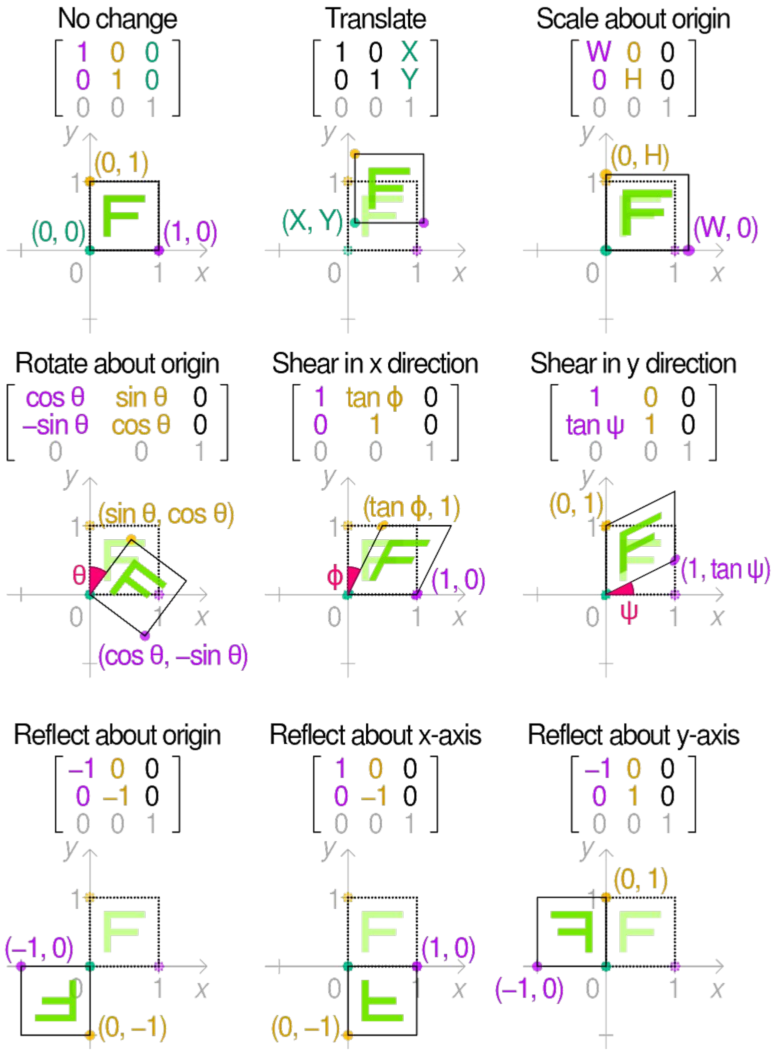


$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 2 \end{bmatrix} \qquad \begin{bmatrix} 2 \\ -2 \end{bmatrix}$$



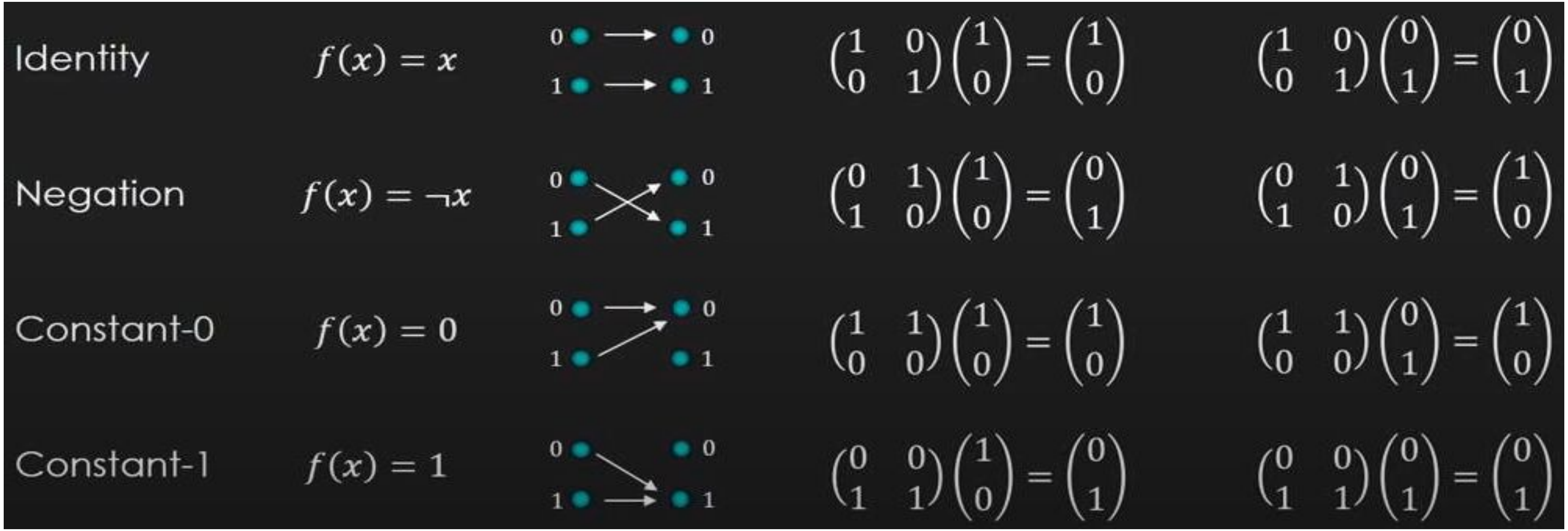
More Complex Matrix Transformations

Why we need 3x3 Matrices for more advanced 2d transformations?
https://stackoverflow.com/questions/10698962/why-do-2d-transformations-need-3x3-matrices



What we’ll be using matrices for today?

We’re going to see that we can use matrices to represent functions on bits, using our previous definitions of bits as 2d vectors, as well as more complex multi-bit transformations.



Tensor Product

- Fancy looking new operator X in circle: $\otimes$
- Don’t worry about what it is or why it works like this, just realize what it does and how it can be used.
- We use the tensor product for representing multiple bits as a single vector called the <u>individual state representation</u> as shown on side 5

Tensor Product Formula

$$\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \otimes \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \\ x_1 \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} x_0 y_0 \\ x_0 y_1 \\ x_1 y_0 \\ x_1 y_1 \end{pmatrix}$$

$$\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \otimes \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \otimes \begin{pmatrix} z_0 \\ z_1 \end{pmatrix} = \begin{pmatrix} x_0 y_0 z_0 \\ x_0 y_0 z_1 \\ x_0 y_1 z_0 \\ x_0 y_1 z_1 \\ x_1 y_0 z_0 \\ x_1 y_0 z_1 \\ x_1 y_1 z_0 \\ x_1 y_1 z_1 \end{pmatrix}$$

Individual state representation of 4

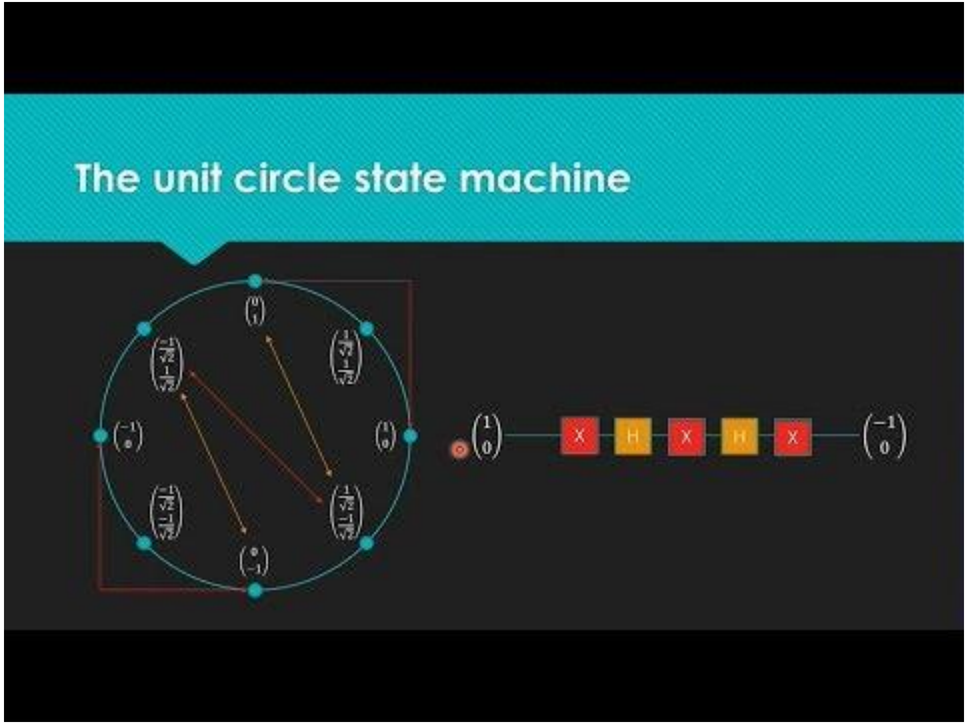
$$|4\rangle = |100\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

The Talk

Some things that will be mentioned

- Shor’s Algorithm: Famous quantum algorithm developed by Peter Shor to factor really large numbers, breaks current encryption that relies on really large numbers being really hard to factor.
- Deutsch–Jozsa Algorithm: Solves the more complex case of the Deutch oracle problem, in 1 operation on a quantum computer making it an O(1) constant time algorithm, while a classical computer requires 2^(n-1) + 1 operations making it an O(2^n) exponential time algorithm.
- Google being expected to achieve Quantum Supremacy by the end of the year? Didn’t happen, they achieved it one year later and even this claim has been hotly contested by competitors.

Let’s Watch the talk!



Let’s Discuss!

- What did we learn? What questions do we have?
    - Mathematics behind quantum computing
    - Quantum gates & quantum circuits
    - The Deutch oracle problem
    - Entanglement & Teleportation
    - A brief look at Q#
- Did this talk help get you interested in Quantum Computing or scare you away?
- Can you think of any interesting applications?
- Would you play around with IBMs Free Quantum Computation in your free time?

Resources

- Run Quantum Algorithms on a real quantum computer through IBM: https://quantum-computing.ibm.com/docs/
- Quantum Computing For Computer Scientists Talk Page: https://www.microsoft.com/en-us/research/video/quantum-computing-computer-scientists/

Thanks For Coming!