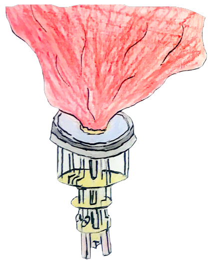

Mathematical Operating System

Shayan Karami

September, 2023

Abstract

This paper is a brief introduction of Mathematical Operating System, which shows the fundemental parts of this system which are uniq on this operating system. In here we give a explnation of considring a Qubit system as a function where each point of x, is one Qubit, and the value of y is the number of spin of this Qubit; And a surface which is created by the connection of a start function f(x) to g(x), which is a path of function f(x) become g(x) and it is our algorithm.

Introduction

In any computation system we have a kind of processing unit, in classical computers bits, and in Quantum computers Qubits. By this we consider a Quantum computer as a set of Qubits with a spin for each, and by sending a specific amount of energy we can change the spin of a Qubit. The foundation of our theory is considering this set of Qubits as a function where numbers on x axis is the Qubit, and the value of y is the spin of the Qubit. The y axis is limited by two numbers 1 and -1, where y = 1 it means spin is UP, and for y = -1 spin is DOWN, and for any number in between is in superposition.

By this theory we use geometric methods and intuition to creating algorithms for quantum computers. We assume at time 0 we have the function f(x) and this function become function g(x) in time t, this means function f(x) converted to function g(x), so we consider this as a conversion surface. This means the surface f(x) -¿ g(x) is the algorithm of giving set of Qubits do a calculation and the result is g(x). Imagine we have a set of Qubits which is A(x), and we have a function B(x) where it is the result of a sum of one number to it self; By this when we make a conversion surface of A(x) to B(x) we have a geometric algorithm of adding a number to it self.

The beautiful part of this operating system is by having this function we can create new set of algorithms by connecting deferent algorithms together, for example continue this function by the rate of the change of A(x) to B(x), and create a new function C(x).This system is similar to making all natural numbers by only having numbers 1 and 0, and operations addition, subtraction, multination and division.And a key application is we are not limited to only a system of companions where function f(x) become g(x), but we can convert n functions to n functions, and the surface which this create is a combination of paths between theses functions.

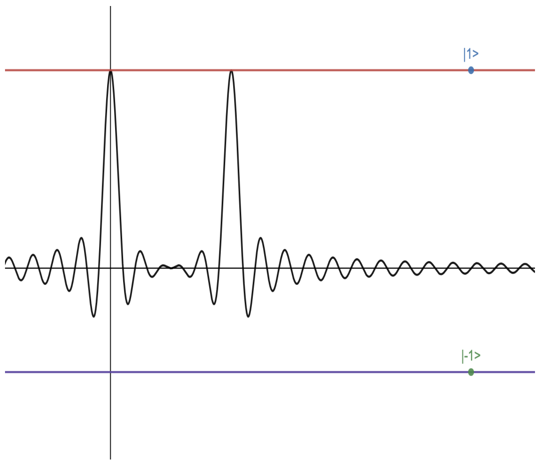

Figure 1: Example of a function limited between two lines of y, spin $\uparrow$, and spin $\downarrow$.

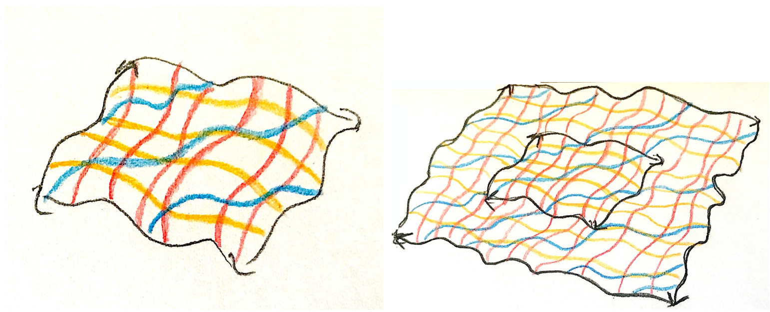

Figure 2: The basic intuition of continuation of one surface by its line integral of $f : A(x) \to B(x)$ to create $f : B(x) \to C(x)$.

Manifold's surface structer

We define the surface as a $\mathbb{R}^2$ manifold where its boundary c is the sum function A(x), B(x) and A(1) $\to$ B(1) and A(n) $\to$ B(n). Let $\vec{A}$(x, y, z) = $\langle$P, Q, R$\rangle$. Manifold S: x=x, y=y, z=g(x,y). And the tangent vectors are $\vec{t_x}$=$\langle 1,0,g_x \rangle$ and $\vec{t_y}$=$\langle 0,1,g_y \rangle$, there fore $\vec{t_x}$X $\vec{t_y}$=$\langle -g_x, -g_y, 1 \rangle$.

$$\iint_S curl\ \vec{F}.D\ \vec{s} = \iint_D [-(R_y - Q_z)z_x - (p_z - R_x)z_y + (Q_x - P_y)]\, dA$$

Then we define the line integral of the boundary as a sum of four functions,

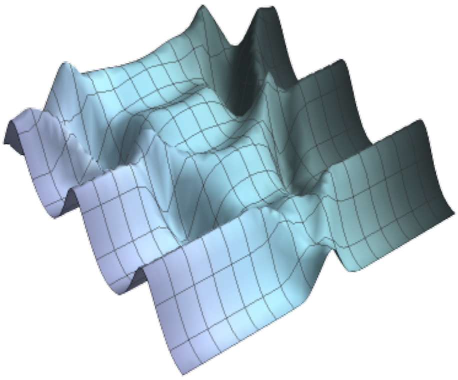

Figure 3: An example of conversion surface of f(x) to g(x).

$$\int_{A(1)}^{A(n)} A'\ dx + \int_{A(n)}^{B(n)} f(z)\ dz + \int_{B(n)}^{B(1)} B'\ dx + \int_{B(1)}^{A(1)} f(z)'\ dz$$

$$\int_c f\ \vec{f}.d\ \vec{r} = \int_a^b (Px'(t) + Qy'(t) + Rz'(t))\ dt$$

$$= \int_a^b [(P + R\frac{\partial z}{\partial x})x'(t) + (Q + R\frac{\partial z}{\partial y})y'(t)]\ dt$$

$$= \int_{c'} (P + R\frac{\partial z}{\partial x})\ dx + (Q + R\frac{\partial z}{\partial y})\ dy$$

$$= \int_D [\frac{\partial}{\partial x}(Q + R\frac{\partial z}{\partial y}) - \frac{\partial}{\partial y}(P + R\frac{\partial z}{\partial x})]\ dA$$

$$= \iint_D (\frac{\partial Q}{\partial x} + \frac{\partial Q}{\partial z}\frac{\partial z}{\partial x} + \frac{\partial R}{\partial x}\frac{\partial z}{\partial y} + \frac{\partial R}{\partial z}\frac{\partial z}{\partial x}\frac{\partial z}{\partial y} + R\frac{\partial^2 z}{\partial x \partial y}) - (\frac{\partial P}{\partial y} + \frac{\partial P}{\partial z}\frac{\partial z}{\partial y} + \frac{\partial R}{\partial z}\frac{\partial z}{\partial y}\frac{\partial z}{\partial x} + R\frac{\partial^2 z}{\partial y \partial x})$$

Higher dimensional computation path

In the classic view, we look at processing as a linear path, where like a function we take a number of inputs, and run the calculations, then it return a number of outputs. But in this theory we are able to have path of processing in higher dimensions. We consider the algorithm or the processing as a $\mathbb{R}^2$ manifold, which has n number of functions which are its boundary.By this we are able to do more complex processing. We have three different types of functions in the boundary, input which we show with +, output by -, and neutral ø. When we combine deference algorithms in a two dimensional structure, we can also design the path of calculation, where + with + become +, and - with - become -, + with - = - with + = ø, and ø with ø = ø, and ± with ø = ±. By this structure were are able to create fractal systems for the computation of an algorithm. By shadowing our manifold in higher diminution, we are even able to create more complex systems of computation, for example the following is making S in $\mathbb{R}^2$ to a $\mathbb{R}^3$ manifold. We define three areas in $\mathbb{R}^3$: First we make a shadow of $\mathbb{R}^2$ manifold to three $\mathbb{R}^2$ shapes, then we make a $\mathbb{R}^3$ manifold from these shapes.

$$\begin{cases} \alpha_1(y,z) \le x \le \beta_1(y,z) \\ (y,z) \in D_1 \end{cases},$$

$$\begin{cases} \alpha_2(x,z) \le y \le \beta_2(x,z) \\ (x,z) \in D_2 \end{cases},$$

$$\begin{cases} \alpha_3(x,y) \le z \le \beta_3(x,y) \\ (x,y) \in D_3 \end{cases}.$$

We define G = G_1I + G_2j + G3k, and n = n_1I + n_2j + n3k,

$$\iiint_W \frac{\partial G_1}{\partial x}\,dV = \iint_{\partial W} G_1 n_1\,dS,$$
$$\iiint_W \frac{\partial G_2}{\partial y}\,dV = \iint_{\partial W} G_2 n_2\,dS,$$
$$\iiint_W \frac{\partial G_3}{\partial z}\,dV = \iint_{\partial W} G_3 n_3\,dS.$$

Now we proof this equilibrium:

$$\iiint_W \frac{\partial G_1}{\partial x}\,dV = \iint_{D_1} (\int_{\alpha_1(y,z)}^{\beta_1(y,z)} \frac{\partial G_1}{\partial x}\,dx)\,dA_{y,z},$$

$$\iiint_W \frac{\partial G_1}{\partial x}\,dV = \iint_{\partial D_1} [G_1(\beta_1(y,z),y,z) - G_1(x_1(y,z),y,z)]\,dA_{y,z},$$

On the other hand to calculate $\iint_{\partial W} G_1 n_1\,dS$, we should calculate n and dS for α_1 and β_1 diagrams. By using r(y,z) = (β_1(y,z),y,z) for diagram β_1,

$$ndS = r_y X r_z dA_{y,z}$$
$$= (\frac{\partial \beta_1}{\partial y}, 1, 0) X (\frac{\partial \beta_1}{\partial z}, 0, 1) dA_{y,z}$$
$$= (1, -\frac{\partial \beta_1}{\partial y}, -\frac{\partial \beta_1}{\partial z}) dA_{y,z}$$

And doing the same calculations for α diagram we get:

$$ndS = (-1, \frac{\partial \alpha_1}{\partial y}, \frac{\partial \alpha_1}{\partial z}) dA_{y,z}$$

Thus $n_1 dS = dA_{y,z}$ is the result of

$$\iiint_W \frac{\partial G_1}{\partial x}\,dV = \iint_{D_1} (\int_{\alpha_1(y,z)}^{\beta_1(y,z)} \frac{\partial G_1}{\partial x}\,dx)\,dA_{y,z},$$

which is equal to $\iint_{\partial W} G_1 n_1 \, dS$, and we proof this for the first equation,

$$\iiint_W \frac{\partial G_1}{\partial x} \, dV = \iint_{\partial W} G_1 n_1 \, dS,$$

and for other two equations are same as this proof. From this theorem we can use the equilibrium of the $\mathbb{R}_2$ manifold to three diagrams, to create a $\mathbb{R}_3$ manifold from these three equations.

Conclusion

This was a short introduction of our project Mathematical Operating System in Laplace's demon where we are trying to make a this theory and structure to an operating system which is run by a combination of classical computers and quantum computers. For contact to us, you can email Shayan Karami: ftfsk8203@gmail.com, and Ali Karami: Erfun.frmn@gmail.com.