

METHOD AND PROGRAMMABLE APPARATUS FOR QUANTUM COMPUTING

5

CROSS-REFERENCE TO RELATED APPLICATION

This application claims the benefit of a provisional patent application no. 60/447,566, filed on February 14, 2003.

10

TECHNICAL FIELD

The present invention is related to a method and programmable apparatus for quantum computing, and, in particular, a general method and programmable apparatus that uses quantum state propagation for quantum computing.

15

BACKGROUND OF THE INVENTION

Computer technology has evolved at an amazing rate in the past few decades. Further increases in the density of semiconductor integrated circuits may soon end due to the practical difficulties in patterning features with dimensions smaller than the wave length of ultraviolet light and soft x-rays. Ultimately, features cannot possibly
20 be smaller than the length scale of single atoms and molecules.

At the molecular level, quantum effects may provided huge advantages as well as problems not encountered at larger dimensions. Quantum computing has recently emerged as a possible approach to developing smaller computing devices that operate on the atomic level. In general, a traditional computer encodes information in a series
25 of bits for computation that are manipulated via Boolean logic, which is based on binary mathematics. For example, a 32-bit digital computer generally manipulates one or several 32-bit operands during execution of a single instruction. A single instruction generally produces one or several 32-bit result values, each representing one of 2^{32} different possible result values. Thus, a single instruction maps one or
30 several operands to one or several results, each within a range of 2^{32} values. On the other hand, a typical quantum computer utilizes the $+\frac{1}{2}$ and $-\frac{1}{2}$ nuclear spin states for isotopes such as ^{13}C and ^{19}F to represent the Boolean logic one and zero. In addition, the nuclear spin is also a quantum mechanical object that can exist in a superposition

of $+\frac{1}{2}$ and $-\frac{1}{2}$ nuclear spin states. In theory, a typical 32-quantum-bit quantum computer can produce any combination of 2^{32} values (i.e., $2^{32}!$ combinations) as a result of executing an instruction, providing for massive parallelism. However, in order to take advantage of the inherent $\frac{1}{2}$ -spin state of isotopes such as ^{13}C and ^{19}F ,
5 the processor must be operated at temperatures near absolute zero.

Computer manufacturers, engineers, and physicists have recognized the need for quantum computers that can handle the increasing demand for speed and computational volume and that are capable of operating at room temperature.

10 SUMMARY OF THE INVENTION

One embodiment of the present invention is directed towards a method and apparatus for quantum computing. In a described embodiment, data, a program, and uninitialized variables, or program output is provided. A compiler program converts the source code into an assembler code of computable functions. The computable
15 functions are encoded by an encoding function. A continualization method is employed to determine a first-order, time-dependent, differential equation from the encoded functions. The variational principle is used to construct a Lagrangian whose minimum geodesic is the solution for the first-order, time-dependent, differential equation. The Lagrangian is converted into the quantum, canonical, Hamiltonian
20 operator which is realized as an excitation field produced by an excitation generator. A control and scheduling system is employed to moderate the intervals between discrete applications of the excitation field produced by the excitation generator. The excitation field is repeatedly applied to a quantum processor consisting of a lattice of polymer nodes. The light radiation emitted by the quantum processor lattice of
25 polymer nodes is converted by a transducer into an intensity-versus-vibrational-frequency spectrum. A running average of the intensity-versus-vibrational-frequency spectrum is maintained in the coherent memory. The average spectrum intensity values are used as coefficients in a polynomial approximating the encoding function. The output consists of the original data and program, and the substantiated variables,
30 or program output.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a control-flow diagram for the highest-level view of the method of one of many possible embodiments of the present invention.

Figure 2 shows a control-flow diagram for a mid-level view of the method and apparatus that represents one of many possible embodiments of the present invention.

Figure 3 shows a diagram of the apparatus comprising the “functional architecture” in Figure 2 that represents one of many possible embodiments of the present invention.

Figure 4 is a control-flow diagram for the routine “functional architecture.”
Figures 5A-B illustrate a simple example input function.

Figures 6A-B illustrate the application of the encoding in equations (3) – (6) to the simple example function.

Figure 7A is an illustration of continualization of the encoding function F in Figures 3A-B via the step function in equation (8).

Figure 7B is an illustration of a hypothetical interpolating function $\Phi(y)$.

Figures 8A-F illustrate the continualization method for a hypothetical sequence in three-dimensions.

Figure 9 is an illustration of a hypothetical geodesic solution $\mathbf{x}(t)$ of equation (23) on the three-dimensional carrier manifold.

Figures 10A-C illustrate the Morse potential and spectrum for a hypothetical diatomic molecule.

Figure 11 is an illustration of a quantum processor having a two-dimensional array of nodes representing one of many possible embodiments of the present invention.

Figure 12 shows a simple cubic lattice of equally spaced nodes, reflective plates, and insulating boundaries representing one of many possible embodiments of the present invention.

Figure 13 shows realization of the excitation field H_f generated by the excitation generator.

Figure 14 is a diagram of the two-level automaton for a single node.

Figure 15 illustrates the relationship between the chattering combination in equation (51) and the semi-open time intervals in equation (52).

Figure 16 illustrates chattering for a single hypothetical node using three pure states in the chattering combination (51).

Figure 17 shows examples of lattice node interactions.

Figure 18 illustrates the interactions characterized by the interaction
5 Hamiltonian H_I .

Figure 19 is an illustration of forward and backward propagating wave trains.

Figures 20A-C illustrate a single hypothetical run of the excitation field Hamiltonian H_f used to obtain a hypothetical spectrum from the quantum processor.

Figure 21 shows example hypothetical spectra and the average spectrum after
10 n hypothetical runs of the excitation field H_f .

Figure 22 is a control-flow diagram depicting determination of the transducer read out period that represents one of many possible embodiments of the present invention.

15 DETAILED DESCRIPTION OF THE INVENTION

The present invention relates to a method and programmable apparatus termed the Quantum Function Evaluator that uses quantum state propagation for computational processing. Figure 1 shows a control-flow diagram for the highest-level view of the method of one of many possible embodiments of the present
20 invention. In step 102, the input consisting of data 104, a program 106, and uninitialized variables 108, or program output, is provided. The data 104 may be in the form of a separate file, input by an operator at the keyboard of computer, or actually incorporated into the text of the program 106. The program 106 may be a source code written in a high level programming language such as C, C++,
25 FORTRAN, etc., or a lower level language such as assembler, or high level language such as scripting languages, shell scripts, or higher level mathematical programming languages. The uninitialized variables 108 represent program output, and may include data values transferred to any of many different output devices and types, including display terminals, electronic communications media, files or database
30 management system data stored on mass storage devices, and many other types of output. Edge 110 represents executing one embodiment of the present invention that in a single iteration converts the input in step 102 into the output in step 112. In step

112, output includes the same data 104 and program 106. The uninitialized variables 108, however, are now instantiated variables 114 determined by execution of the method and apparatus of the present invention. Note that if the data 104 in step 102 is changed, then a separate execution of steps 102 and 112 is needed to determine the instantiated variables 114.

Figure 2 shows a control-flow diagram for a mid-level view of the method and apparatus that represents one of many possible embodiments of the present invention. Step 202 is the data 104, program 106, and uninitialized variables 108 described in step 102 in Figure 1. Steps 204 through step 216 provide a more detailed description of the method represented by edge 110 in Figure 1. In step 204, a compiler program converts the source code and data in step 202 into an assembly code representing a set of discrete valued computable functions. In step 206, the assembly code is converted into a discrete encoding of partial recursive functions. In step 208, a continualization method is employed to generate a continuous, first-order, time-dependent, differential equation from the encoded function determined in step 206. In step 210, the variational principle is employed to construct a Lagrangian whose geodesic is the solution for the first-order, time dependent, differential equation determined in step 208. In step 212, the Lagrangian is converted into a quantum, canonical, Hamiltonian operator. In step 214, the quantum, canonical, Hamiltonian operator is physically instantiated by producing an excitation field described by the quantum, canonical, Hamiltonian operator. In step 216, the routine "functional architecture," representing one of many possible embodiments of the present invention, is employed to determine the coefficients of a polynomial approximation to the encoding function in step 206. After the polynomial approximation is determined in step 216, steps 206 through 202 are performed in reverse order. In step 204, the polynomial approximation of the encoding function can be used to determine the computable functions in step 204, which leads to step 202, where the instantiated variables 114 in Figure 1 are given.

Figure 3 shows a diagram of the apparatus comprising the "functional architecture" 216 in Figure 2 that represents one of many possible embodiments of the present invention. Located in the center of Figure 3 is a control and scheduling system 301 that controls and regulates the execution of a function input 302, an excitation generator 303, a quantum processor 304, a transducer 305, and a coherent

memory 306, as indicated by edges 307 - 311. Edges 312 - 317 indicate the direction in which information flows between the elements of the functional architecture. The function input 302 is employed to carry out steps 202 through 214 in Figure 2 to determine the excitation field and pass the Hamiltonian characterizing the excitation field to an excitation generator 303. The excitation generator 304 realizes the excitation field which is used to induce an emission of light radiation from the polymer molecule nodes of the quantum processor 304. The excitation field generated by the excitation generator 303 may be an electric field, light, radio waves, or any other means that can be used to generate vibrational excitation of polymer molecules. The transducer 305 converts the emitted light radiation from the quantum processor 304 into a coherent spectrum of intensities and corresponding frequencies that are passed to, and stored by, the coherent memory 306. The transducer 305 is also responsible for clearing the light radiation detector after each reading of the emitted light radiation from the quantum processor 304, as indicated by edge 318. The coherent memory 306 maintains a running average of the intensities and frequencies received by the transducer 305. The control and scheduling system 310 repeatedly executes the excitation generator 303, the quantum processor 304, and the transducer 305 until the average spectrum intensities converge. Upon convergence of the spectrum, the average spectrum intensities are passed to read out 319 and subsequently used as the coefficients of the polynomial that approximates the encoding function in step 206 in Figure 2.

Figure 4 is a control-flow diagram for the routine "functional architecture." In step 402, the function input determines the excitation field Hamiltonian according to steps 202 through 214, described above in relation to Figure 2. In step 404, a *do-loop* representing the control and scheduling system 310, described above in relation to Figure 3, executes steps 406, 408, 410, and 412. In step 406, the excitation generator realizes the excitation field Hamiltonian by bringing into physical existence an actual excitation field. In step 408, the quantum processor is impinged by the excitation field causing the polymer nodes to emit light radiation. In step 410, the emitted light radiation is converted into an intensity-versus-light-wave-frequency spectrum and a running average is maintained. In step 412, if the running average converges, then in step 414, the spectrum of intensities are used as coefficients of the approximating

polynomial to the encoding function in step 206 in Figure 2. In step 412, if the running average has not converged, then steps 406, 408, 410, and 412 are repeated.

I. Quantum Computing

5

A. Input Function and Encoding

The computable functions generated in step 104 above are discrete valued functions referred to as “input functions.” The input functions may be of the form:

10

$$f : S_1 \times \cdots \times S_N \rightarrow D$$

(1)

where N is the dimension of the domain $S_1 \times \cdots \times S_N$, and the sets S_i and D are finite subsets of the natural numbers given by:

15

$$S_i = \{0, \dots, N_i\}$$

$$D = \{0, \dots, N_d\}$$

where N_i and N_d are the largest natural numbers in the sets S_i and D , respectively. The function f in equation (1) maps an N -tuple denoted $(x_0, \dots, x_{N-1})$ in the domain $S_1 \times \cdots \times S_N$, where x_{i-1} is an element in the set S_i , to a single natural number in the range D . The N -tuples $(x_0, \dots, x_{N-1})$ are referred to as “points” in the domain $S_1 \times \cdots \times S_N$.

Figures 5A-B illustrate a simple example input function having the form of equation (1) given by:

25

$$f : S_1 \times S_2 \rightarrow D$$

where the domain $S_1 \times S_2$ is the set of all possible combination of points (x_0, x_1) such that x_0 and x_1 are respectively elements in the set:

30

$$S_1 = \{0, 1, 2\}$$

and

$$S_2 = \{0, 1, 2\}$$

The range of the example function f is given by the set:

$$D = \{0, 1, 2\}$$

10

Figure 5A shows a three column table of domain and range values of example function f . The x_0 -column 501 and x_1 -column 502 contain the domain elements of $S_1 \times S_2$, and the range values are located in f -column 503. Figure 5B is a three-dimensional, Cartesian-coordinate depiction of the points (x_0, x_1, f) given in the table in Figure 5A. The collection of points (x_0, x_1, f) are plotted with respect to the x_0 -axis 505, the x_1 -axis 506, and the f -axis 507. For example, the point 508 in Figure 5B is the graphical representation of the point in row 504 in Figure 5A.

It should be noted that, although two and three-dimensional illustrations, such as Figures 5A-B are provided in the following discussion, they are provided only for illustration purposes. In real-world applications, there may be many thousands, hundreds of thousands, or millions of variables, leading to extremely high dimensional problem domains. In general, these problem domains may be considered to be hyper-dimensional volumes, manifolds, or carrier manifolds. It is not possible to illustrate such problem domains, but techniques used to address hyper-dimensional domains may be analogized to three-dimensional illustrations.

The input function may also be a solution of an iterative process of the form:

$$y_{n+1,i} = f_i(y_{n,1}, \dots, y_{n,k}) \quad (2)$$

30 where $i = 1, \dots, k$, and each

$$f^i : S^k \rightarrow S$$

where $S = \{0, 1, \dots, N_S\}$.

The encoding of computable functions as described above in step 206 of
 5 Figure 2 can be accomplished using the following function:

$$F : [0, 1, \dots, p^{N-1}] \rightarrow [0, 1, \dots, p] \quad (3)$$

where

10

$$x = \sum_{s=0}^{N-1} x_s \cdot p^{N-1-s} \quad (4)$$

and p is a natural number defined by:

$$15 \quad p = \max_{i,d} \{ \{N_i | i=1, \dots, N\}, N_d \} + 1 \quad (5)$$

and

$$F_i(x) = \begin{cases} f_i(x_0, \dots, x_{N-1}) & \text{if defined} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

20

The encoding functions (3) - (6) map the domain values in $S_1 \times \dots \times S_N$ to a subset of points of the real numbers such that the values of F coincide with the values of f in the range of f . Furthermore, the encoding functions (3) - (6) map x to 0 for any x that corresponds to an N -tuple not in the domain of f .

25 Figures 6A-B illustrate application of the encoding in equations (3) - (6) to the simple example function. The value for the natural number p given by equation (5) is given by:

$$p = \max_{i,d} \{\{2, 2\}, 2\} + 1 = 3$$

The encoding function in equation (4) maps the points (x_0, x_1) in the domain $S_1 \times S_2$ to a subset of the real numbers according to the expression:

5

$$x = \sum_{s=0}^1 x_s \cdot p^{1-s} = x_0 3 + x_1$$

(7)

10 In Figure 6A, the x -column 601 are the x -values calculated according to equation (7), and F -column 602 are the corresponding encoded function values F . In Figure 6B, the collection of points (x, F) displayed in Figure 6A are plotted with respect to the x -axis 603 and the F -axis 604. For example, the point $(1, 2, 2)$ 508 in Figures 5B is mapped to the point $(5, 2)$ 605 in Figure 6B.

15 The example input functions given in equations (1) and (2) are functions of a larger class of discrete-valued functions generated by the compiler program in step 204 in Figure 2. The class of functions generated in step 204 is referred to as the "Class I " of computable functions and are given as follows:

20 (i) Class I contains all the functions F encodable from a finite function f as in equations (3) – (6).

(ii) If

$$\{f_1, \dots, f_K \mid f_i : S_1 \times \dots \times S_N \rightarrow S_{N+1}, K \text{ finite}, S_i = \{1, \dots, N_i \mid N_i \text{ finite}\}\}$$

25

is a subset of Class I , so is the direct sum:

$$f_1 \oplus \dots \oplus f_K : S_1 \times \dots \times S_N \rightarrow S_{N+1}^K$$

30 (iii) Class I contains the projection functions:

$$p_i : S_1 \times \cdots \times S_N \rightarrow S_i$$

$$\text{where } p_i(n_1, \dots, n_i, \dots, n_N) = n_i$$

5 (iv) If

$$\{f_1, \dots, f_K \mid f_i : S_1 \times \cdots \times S_N \rightarrow S_{N+1}, K \text{ finite}, S_i = \{1, \dots, N_i \mid N_i \text{ finite}\}\}$$

is a subset of Class I and

10

$$g : S_{N+1}^K \rightarrow S_{N+1}$$

is in Class I , so is the composition:

15

$$g(f_1, \dots, f_K) : S_1 \times \cdots \times S_N \rightarrow S_{N+1}$$

(v) If for each $n \in N$,

$$g : N \times S_1 \times \cdots \times S_K \rightarrow N$$

20

is in Class I so is the following:

$$\min\{n, g(n, n_1, \dots, n_K) = 0\}$$

25 (vi) If

$$g : N \times S_K \rightarrow S_K$$

is in Class I , then the family of functions given by:

30

$$\left\{ f : N \times S_1 \times \cdots \times S_K \rightarrow S_{K+1} \mid f(n+1, n_1, \dots, n_K) = g(n, f(n, n_1, \dots, n_K)) \right\}$$

is in Class *I*.

(vii) Any function constructed by the finite application of (i)-(vi) is in *I*.

5 The elements of Class *I* are either an encoding or can be transformed into an encoding function via a finite number of steps (ii) – (vii) above. Clauses (ii) – (iv) are included because these clauses are constructed by the compiler program found in many high-level programming languages. Although the Class *I* of computable functions includes only completely specified functions, in alternate embodiments the method of the present invention can be modified to handle the larger class of discrete partially specified functions and indicator functions.

The central objective behind the method and apparatus of the present invention is to find effective and fast means of computing the discrete Class *I* functions via quantum approximations to the encoding functions.

15

B. Continualization

The method of continualization is described in detail in patent application no. 10/693,729 and is incorporated by reference. The example function provided in Figures 5 and 6 can be continualized using a step function Φ defined by:

20

$$\Phi : [0, \dots, p^{N-1}) \rightarrow [0, \dots, p^{N-1}) \quad (8)$$

$$\Phi(y) = F(x) \text{ for each } y, y \in [x, x+1), y \in \{0, \dots, p^{N-1}\}$$

Figure 7A illustrates continualizing the encoding function *F* shown in Figures 6A-B via the step function in equation (8). Continualization increases the size of the domain by including the real numbers between the discrete domain values *x* and assigns to each *y* a function value $\Phi(y)$. Figure 7A shows the graph of step-function $\Phi(y)$ coincide respect to the *x*-axis 701 and the Φ -axis 702. The function values $F(x)$ equal to $\Phi(y)$ where the discrete *x* values equal the discrete domain *y* values.

25

30 For example, the point (5, 2) 703 in Figure 7A is the point (5, 2) 605 in Figure 6B.

The continualization process is not limited to the step function given by equation (4). In alternate embodiments, other interpolation schemes can be used to construct a function $\Phi(y)$ whose values coincide with $F(x)$ such as linear interpolations, Adams-Boshworth interpolation, splines, Pade interpolation, etc.

5 Figure 4B is an illustration of a hypothetical interpolating function $\Phi(y)$, where $\Phi(y)$ equals the encoding function $F(x)$ for values of y equal to the discrete domain values x .

The interpolating function $\Phi(y)$ may be expressed for hyper-dimension manifolds as a sequence of real vectors $\{x_0, x_1, \dots\}$ defined by the
10 following recursion relation given by:

$$x_{n+1} = x_n + a_n h(x_n) \quad (9)$$

15 where each vector x_n is a k -tuple in k -dimensional set of real numbers R^k given by:

$$x_k = (x_{1,k}, x_{2,k}, \dots, x_{N,k})$$

and sequence $\{a_n\}$ is a sequence of real numbers having the properties:

20

$$\lim_{n \rightarrow \infty} a_n = 0$$

$$\sum_n a_n = \infty$$

25 and h is function mapping from R^k to R^k . In other words, a new state-vector value x_{n+1} is computed, in each iteration of a computation, from a current state-vector x_n and a discrete, vector-valued function h that depends on x_n . Figure 8A-F illustrate the continualization method for a hypothetical sequence in three-dimensions. Figure 8A shows the vectors x_{n-2} , x_{n-1} , x_n , and x_{n+1} plotted in R^3 with respect to the x_I -axis 802,

the x_2 -axis 804, and the x_3 -axis 806. Next, the sequence $(\mathbf{x}_n)_{n \geq 0}$ can be approximated by the sequence $(\mathbf{X}_n)_{n \geq 0}$ having terms satisfying the recursion relation given by:

$$\mathbf{X}_{n+1} = \mathbf{X}_n + a_n \mathbf{h}(\mathbf{X}_n) + a_n \mathbf{b}_n \quad (10)$$

where $\mathbf{b}_n$ is a vector parameter. Figure 8B shows the sequence given by the approximating recursion relation in equation (10). The sequence of vectors $(\mathbf{x}_n)_{n \geq 0}$ shown in Figure 8A have been replaced by the sequence of vectors $(\mathbf{X}_n)_{n \geq 0}$ in Figure 8B.

The time sequence $\{t_n\}$ can be defined as follows:

$$t_n = \sum_{i=0}^{n-1} a_i \quad (11)$$

where $a_i = t_{i+1} - t_i$.

Figure 8C is an illustration of the relationship between the time sequence $\{t_n\}$ and the real number sequence $\{a_n\}$ given in equations (9) and (10). Note that because each element a_n approaches zero as n approaches infinity, the real number sequence $\{a_n\}$ is the length of the decreasing time intervals $(t_{n+1} - t_n)$ on the time axis 808. For example, real number a_1 is a longer interval than a_n .

Next, two time dependent vector functions $\mathbf{X}^0(t)$ and $\mathbf{g}^0(t, a, b)$ are defined as follows:

$$\mathbf{X}^0(t_n) = \mathbf{X}_n \text{ for all } n \geq 0 \quad (12)$$

$$X^0(t) = \frac{t_{n+1}-t}{a_n} X_n + \frac{t-t_n}{a_n} X_{n+1} \text{ if } t \in (t_n, t_{n+1})$$

(13)

and

$$g^0(t_n, a, b) = \sum_{i=0}^{n-1} a_i b_i \text{ for all } n \geq 0$$

(14)

$$g^0(t, a, b) = \frac{t_{n+1}-t}{a_n} g^0(t_n, a, b) + \frac{t-t_n}{a_n} g^0(t_{n+1}, a, b) \text{ if } t \in (t_n, t_{n+1})$$

(15)

10

Equations (13) and (15) are linear functions with respect to time t . Figure 8D is an illustration of the functions defined by equation (13) and shows that segment $X^0(t)$ 814 is a vector between the vectors X_n and X_{n+1} in R^3 .

A function $\bar{X}^0$ can be defined by the following:

15

$$\bar{X}^0(t) = X_n \text{ for } t \in (t_n, t_{n+1})$$

(16)

Figure 8E is an illustration of the equation (16) plotted as a function of time t . The function values are plotted as a step function to show that $\bar{X}^0(t)$ 816 is constant on the time interval (t_n, t_{n+1}) .

20

Summing the terms in equation (10) from zero to $n-1$ gives the following expression:

$$\sum_{i=0}^{n-1} X_{i+1} = \sum_{i=0}^{n-1} X_i + \sum_{i=0}^{n-1} h(X_i) a_i + \sum_{i=0}^{n-1} a_i b_i$$

(17)

25

After some algebra and applying the definition of the integral from integral calculus, equation 17 can be given as continuous, vector-valued function of time t as follows:

$$X^0(t) = X^0(0) + \int_0^t h(\bar{X}^0(s)) ds + g^0(t, a, b) \quad (18)$$

The functions defined as:

$$X^n(t) = \begin{cases} X^0(t+t_n) & \text{if } t > -t_n \\ X_0 & \text{if } t \leq -t_n \end{cases}$$

$$g^n(t, a, b) = \begin{cases} g^0(t+t_n, a, b) - g^0(t_n, a, b) & \text{if } t > -t_n \\ -g^0(t_n, a, b) & \text{if } t \leq -t_n \end{cases}$$

can be used to rewrite equation (18) in the following form:

$$X^n(t) = \begin{cases} X^n(0) + \int_0^t h(\bar{X}^n(t_n+s)) ds + g^0(t, a, b) & \text{if } t > -t_n \\ X_0 & \text{if } t \leq -t_n \end{cases} \quad (19)$$

Equation (19) can be expressed equivalently as follows:

$$X^n(t) = \begin{cases} X^n(0) + \int_0^t h(\bar{X}^n(s)) ds + g^n(t, a, b) + e^n(t) & \text{if } t > -t_n \\ X_0 & \text{if } t \leq -t_n \end{cases}$$

where the sequence of functions $e^n(t)$ and $g^n(t, a, b)$ approach zero as n approaches infinity uniformly on finite time intervals $(t_{n+1} - t_n)$. The time dependent sequence $\{X^n(t)\}$ is bounded and equicontinuous on interval $(-\infty, \infty)$. Thus by the Arzela-

Ascoli Lemma, there is a convergent subsequence of the sequence of functions $\{X^n(t)\}$ whose limit is the vector function $X(t)$ that satisfies the following:

$$\dot{X}(t) = h(X(t))$$

5

where $\dot{X}$ is a shorthand notation for the time derivative of X .

The sequence of functions $\{g^n(t, a, b)\}$ are also bounded and equicontinuous on interval $(-\infty, \infty)$. Again, applying by the Arezla-Ascoli Lemma, there exists a convergent subsequence of functions $\{g^n(t, a, b)\}$ whose limit is the vector function $g(t, a, b)$. Thus the original discrete sequence given by equation (9) can be modeled by the first-order, time-dependent, differential equation given by:

10

$$\dot{x}(t) = h(x(t)) - \dot{g}(t, a, b)$$

(20)

15

Figure 8F is an illustration of the differential equation (20) as a tangent vector to the vector solution functions $x(t)$.

C. Variational Model

20

The variational model is used to formulate a Lagrangian for the first-order, time-dependent, differential equation (20) in step 210 in Figure 2. The procedure begins with the variational problem formulated in terms of the Lagrangian given by:

$$L(x, \dot{x}, t)$$

25

First, the Lagrangian is assumed to have to the following properties:

(1) The Lagrangian is homogeneous of degree 1 in $\dot{x}$. In other words,

$$L(x, \lambda \dot{x}, t) = \lambda L(x, \dot{x}, t) \text{ for a real valued } \lambda \geq 0.$$

(2) The Lagrangian is positive definite in $\dot{x}$. In other words,

$$(g_{ij}(x, \dot{x}, t)) = \left(\frac{1}{2} \frac{\partial^2 (L^2(x, \dot{x}, t))}{\partial \dot{x}_j \partial \dot{x}_i} \right)$$

5

Properties (1) and (2) allow for the definition of a metric ground form ds on the carrier manifold such that

$$ds^2 = \sum_{ij} g_{ij}(x, \dot{x}, t) dx_i dx_j$$

10

(21)

A geodesic is the path that represents the shortest path between any two points when the path is restricted to a particular surface. The geodesics subject to the metric ground in equation (21) are the extremum of the corresponding parametric calculus of variations problem which is to find a curve on the manifold which minimizes the following integral:

15

$$\int_{t_0}^{t_1} L(x(t), \dot{x}(t), t) dt$$

(22)

satisfying the endpoint conditions. The geodesics are solutions to the Euler-Lagrange equation given by:

$$\frac{d}{ds} \frac{\partial L(x, \dot{x}, t)}{\partial \dot{x}_i} - \frac{\partial L(x, \dot{x}, t)}{\partial x_i} = 0$$

(23)

25

Figure 9 is an illustration of a hypothetical geodesic solution $x(t)$ 904 of equation (23) on the three-dimensional surface carrier manifold 902. The path followed by the geodesic solution $x(t)$ 904 is the minimum path between the starting point $x(t_0)$ 906,

which represents the data 104, program 106, and uninitialized variables 108 in step 102 in Figure 1, and the ending point $\mathbf{x}(t_f)$ 908 represents the data 104, program 106, and instantiated variables 114 in step 112 in Figure 1.

Second, the vector function $\mathbf{x}(t)$ defines a trajectory that has an integral within
 5 an ε of its minimum and is referred to as the ε -optimal control.

Third, the Lagrangian $L(x, u, t)$ is replaced by $L^*(x, u, t)$ which is the convexification of the Lagrangian L with respect to u . Then an existence theorem for the relaxed controls can be applied to $L^*(x, u, t)$ to ensure that the optimal control problem has a relaxed solution. The relaxed solution can be approximated to obtain
 10 an ε -optimal control for the original problem.

Fourth, for any pre-specified ε , the ε -optimal control can be computed for the original problem and implement ε -optimal control as a finite-state, physically-realizable, control automaton. The actual control law issued by the finite-control automaton at state x is a chattering control. The chattering is among approximations
 15 to some local extrema of $L(x, u)$.

The desired evolution of solution $\mathbf{x}(t)$ to the first-order, time-dependent, differential equation (20) is assumed to by define a path along a geodesic of the carrier manifold. In other words, there is a desired Lagrangian $L(\mathbf{x}, \dot{\mathbf{x}}, t)$ which induces a metric on the carrier manifold as described in equation (21) by the evolution
 20 of a function $\mathbf{x}(t)$ satisfying equation (20). The covariant partial derivatives on the carrier manifold are defined as follows:

$$\frac{\partial_c}{\partial_c \mathbf{x}} \text{ and } \frac{\partial}{\partial_c \dot{\mathbf{x}}}$$

25 There exists a covariant partial derivative with respect to time t , however, time is an invariant, therefore the covariant partial derivative with respect to time t is just:

$$\frac{\partial_c}{\partial_c t} = \frac{\partial}{\partial t}$$

Therefore, the transformation from local coordinates to coordinates on the carrier manifold is given by:

$$\frac{\partial x_j}{\partial x_i} = \frac{\partial x_j}{\partial x_i} + \sum_l \Gamma_{l,i}^j x_l$$

5

where $\Gamma_{l,i}^j$ are the Christoffel coefficient arising from the Levi-Civita connection associated with the metric in equation (21). The Christoffel coefficients may be computed explicitly via the formula:

$$\Gamma_{l,i}^j = \frac{1}{2} \sum_k g^{j,k} \left(\frac{\partial}{\partial x_i} g_{k,l} + \frac{\partial}{\partial x_l} g_{k,i} - \frac{\partial}{\partial x_k} g_{l,i} \right)$$

10

where $(g^{i,j})$ is the inverse of the metric ground form matrix given by:

$$(g_{ij}(\mathbf{x}, \dot{\mathbf{x}})) = \left(\frac{1}{2} \frac{\partial^2 (L^2(\mathbf{x}, \dot{\mathbf{x}}))}{\partial \dot{x}_j \partial \dot{x}_i} \right)$$

15

For convenience, the covariant partial derivatives The derivation begins by taking the derivative with respect to time t of both sides of equation (20) to give the following expression:

20

$$\ddot{\mathbf{x}} = \frac{\partial_c h}{\partial_c \mathbf{x}} \dot{\mathbf{x}} - \ddot{\mathbf{g}}(t, \mathbf{a}, \mathbf{b})$$

(24)

where $\dot{\mathbf{x}} = [\dot{x}_i]$ and $\ddot{\mathbf{x}} = [\ddot{x}_i]$ are column vectors and

25

$$\frac{\partial_c \mathbf{h}}{\partial_c \mathbf{x}} = \left[\frac{\partial_c h_i}{\partial_c Y_j} \right]$$

is a $k \times k$ matrix. The Lagrangian L is a function mapping defined as follows:

$$L: TM \times R \rightarrow R$$

5

where TM is the tangent bundle and the Lagrangian $L(\mathbf{x}, \dot{\mathbf{x}}, t)$ satisfies the Euler-Lagrange equation given by:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{x}}} \right) - \frac{\partial L}{\partial \mathbf{x}} = 0$$

10

(25)

where $\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{x}}} \right)$ and $\frac{\partial L}{\partial \mathbf{x}}$ are column vectors

Expanding the time derivative in equation (25) gives the following:

15

$$\frac{\partial L}{\partial \mathbf{x}} \dot{\mathbf{x}} + \frac{\partial^2 L}{\partial \dot{\mathbf{x}} \partial \dot{\mathbf{x}}} \ddot{\mathbf{x}} + \frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{\mathbf{x}}} \right) - \frac{\partial L}{\partial \mathbf{x}} = 0$$

(26)

20

The Lagrangian $L(\mathbf{x}, \dot{\mathbf{x}}, t)$ is assumed to be three times differentiable, and the mixed partial derivatives of $L(\mathbf{x}, \dot{\mathbf{x}}, t)$ are assumed to be continuous so that order of differentiation does not make any difference for mixed partial derivatives of $L(\mathbf{x}, \dot{\mathbf{x}}, t)$ of degree 2 or 3. Thus taking the partial derivative of equation (26) with respect to $\dot{x}_k$ of the i th component gives the following:

25

(27)

$$\sum_j \frac{\partial^3 L}{\partial \dot{x}_k \partial \dot{x}_i \partial \dot{x}_j} \dot{x}_j + \frac{\partial^2 L}{\partial \dot{x}_k \partial \dot{x}_i} + \sum_j \frac{\partial^3 L}{\partial \dot{x}_k \partial \dot{x}_i \partial \dot{x}_j} \left(\sum_l \frac{\partial h_j}{\partial \mathbf{x}} \dot{x}_l - \ddot{g}_j(t, \mathbf{a}, \mathbf{b}) \right)$$

$$+\sum_j \frac{\partial L}{\partial_c \dot{x}_j} \frac{\partial \mathbf{h}}{\partial \mathbf{x}} - \frac{\partial_c L}{\partial_c \mathbf{x}} + \frac{\partial}{\partial t} \left(\frac{\partial_c^2 L}{\partial_c \dot{x}_k \partial_c \dot{x}_i} \right) - \frac{\partial_c^2 L}{\partial_c \dot{x}_k \partial_c \dot{x}_i} = 0$$

Equation (27) can be rewritten as:

$$5 \quad \frac{d}{dt} \left(\frac{\partial_c^2 L}{\partial_c \dot{x}_k \partial_c \dot{x}_i} \right) + \sum_j \frac{\partial_c^2 L}{\partial_c x_j \partial_c \dot{x}_i} \frac{\partial \mathbf{h}}{\partial_c x_k} + \frac{\partial_c^2 L}{\partial_c x_k \partial_c \dot{x}_i} - \frac{\partial_c^2 L}{\partial_c \dot{x}_k \partial_c \dot{x}_i} = 0$$

(28)

Thus, equation (28) can be rewritten in matrix form as follows:

$$10 \quad \frac{d}{dt} \left(\frac{\partial_c^2 L}{\partial_c \dot{\mathbf{x}} \partial_c \dot{\mathbf{x}}} \right) + \frac{\partial_c^2 L}{\partial_c \mathbf{x} \partial_c \dot{\mathbf{x}}} - \frac{\partial_c^2 L}{\partial_c \dot{\mathbf{x}} \partial_c \mathbf{x}} = - \left(\frac{\partial_c^2 L}{\partial_c \mathbf{x} \partial_c \dot{\mathbf{x}}} \right) \frac{\partial \mathbf{h}}{\partial_c \mathbf{x}}$$

(29)

Taking the transpose of both sides of equation (29) and using the partial differential property given by:

15

$$\left(\frac{\partial_c^2 L}{\partial_c \mathbf{x} \partial_c \dot{\mathbf{x}}} \right) = \left(\frac{\partial_c^2 L}{\partial_c \dot{\mathbf{x}} \partial_c \mathbf{x}} \right)^T$$

gives the following:

$$20 \quad \frac{dQ}{dt} = -\frac{1}{2} \left(\frac{\partial_c \mathbf{h}}{\partial_c \mathbf{x}} \right)^T Q - \frac{1}{2} Q \left(\frac{\partial_c \mathbf{h}}{\partial_c \mathbf{x}} \right)$$

where

$$Q = \frac{\partial_c^2 L}{\partial_c \dot{\mathbf{x}} \partial_c \dot{\mathbf{x}}}$$

(30)

25

Equation (30) can be explicitly solved for by integrating along the path of the characteristic, second-order, partial-differential equation given by:

$$\ddot{\mathbf{x}}(t) = \frac{\partial_c \mathbf{h}}{\partial_c \mathbf{x}} \mathbf{x}(t) - \ddot{\mathbf{g}}(t, \mathbf{a}, \mathbf{b})$$

5

D. Quantization

In this section the development turns to an alternative statement of the structure and formulation known as the “quantum, canonical, Hamiltonian operator” as first introduced in step 212 in Figure 2. The usefulness of the Hamiltonian viewpoint lies in providing a framework for theoretical extensions in many areas of physics and provides much of the language with which present day quantum mechanics is constructed.

In the Lagrangian formulation given above, the system is characterized by the time-dependent, vector-valued function $\mathbf{x}(t)$ of k independent degrees of freedom and is a problem with k independent variables x_k , and k time derivatives $\dot{x}_k$. The Hamiltonian is fundamentally different from the Lagrangian, because the Hamiltonian is formulated in terms of the position coordinates x_k and the associated momenta p_k (*Classical Mechanics 2nd Edition*, Herbert Goldstein, Addison-Wesley Publishing Co., New York, 1980). The Lagrangian formulation is transformed into a Hamiltonian formulation by first taking the total time derivative of the Lagrangian $L(\mathbf{x}, \dot{\mathbf{x}}, t)$ to get:

$$\frac{dL}{dt} = \sum_k \frac{\partial L}{\partial x_k} \dot{x}_k + \sum_k \frac{\partial L}{\partial \dot{x}_k} \ddot{x}_k + \frac{\partial L}{\partial t} \quad (31)$$

25

Next, the Euler-Lagrangian equation (25) is substituted into equation (31) to give the following equation:

$$\frac{d}{dt} \left(\sum_k \dot{x}_k \frac{\partial L}{\partial \dot{x}_k} - L \right) = - \frac{\partial L}{\partial t}$$

(32)

The quantity in the brackets is referred to as the “energy function” and is denoted by:

5

$$\hat{h}(\mathbf{x}, \dot{\mathbf{x}}, t) = \sum_k \dot{x}_k \frac{\partial L}{\partial \dot{x}_k} - L(\mathbf{x}, \dot{\mathbf{x}}, t)$$

(33)

The energy function (33) can be reformulated in terms of the linear conjugate momenta p_k which are associated with the coordinates x_k and are determined by the equation:

$$p_k = \frac{\partial L}{\partial \dot{x}_k}$$

(34)

The conversion from the Lagrangian formulation to the Hamiltonian formulation is treated strictly as a mathematical problem by substituting equation (34) into equation (33). Thus, the energy function given by equation (33) can now be represented as a Hamiltonian function of the coordinates x_k and conjugate momenta p_k given by:

20

$$H(\mathbf{x}, \mathbf{p}, t) = \sum_k p_k \cdot \dot{x}_k - L(\mathbf{x}, \dot{\mathbf{x}}, t)$$

(35)

Equation (35) is written in this manner to stress the fact that the Hamiltonian is always considered as a function of the coordinates $(\mathbf{x}, \mathbf{p}, t)$, whereas the Lagrangian is a function of the coordinates $(\mathbf{x}, \dot{\mathbf{x}}, t)$.

25

Using the classical mechanical expression for the Lagrangian given by:

$$L(\mathbf{x}, \dot{\mathbf{x}}, t) = \sum_k \frac{1}{2} m_k \dot{x}_k^2 - V$$

where V is a potential energy function of $\mathbf{x}$ or both $\mathbf{x}$ and time t , equation (35) can be
 5 rewritten as a function of the canonical variables $(\mathbf{x}, \mathbf{p})$ to give the classical
 mechanical Hamiltonian:

$$H(\mathbf{x}, \mathbf{p}, t) = \sum_k \frac{p_k^2}{2m_k} + V = E$$

(36)

10

If the potential energy is free of the time t , the system described by the Hamiltonian in
 equation (36) is said to be conservative, which means that no energy is being added to
 or removed from the system.

The Hamiltonian expressed in equation (36) best characterizes dynamical
 15 systems that behave in a continuous manner. However, the Hamiltonian as
 formulated in equation (36) is not adequate for characterizing the quantum
 mechanical method and apparatus of the present invention, because the present
 invention relies on the behavior of atoms and molecules, which exists in a certain
 number of stationary or quantum states.

20 The Hamiltonian in equation (36) can be modified to characterize the quantum
 behavior exhibited by atoms and molecules by substituting the differential operators:

$$E \rightarrow i\hbar \frac{\partial}{\partial t}$$

25 and

$$p_k \rightarrow \frac{\hbar}{i} \frac{\partial}{\partial x_k}$$

into equation (36) (*Quantum Mechanics*, Albert Messiah, Elsevier Science Publishers, The Netherlands, 1961). The resulting equation is the quantum, canonical, Hamiltonian operator given by:

$$H\left(\mathbf{x}, \frac{\hbar}{i} \frac{\partial}{\partial \mathbf{x}}, \frac{\partial}{\partial t}\right) = \sum_k -\frac{\hbar^2}{2m_k} \frac{\partial^2}{\partial x_k^2} + V = i\hbar \frac{\partial}{\partial t} \quad (37)$$

that characterizes the behavior of a quantum mechanical system such as an atom or molecule. Equation (37) is called an “operator” because the kinetic energy and total energy have respectively been replaced by the differential operators given by:

$$\frac{\partial^2}{\partial x_k^2} \text{ and } \frac{\partial}{\partial t}.$$

The Hamiltonian in equation (37) is a second-order, differential, wave equation. In general, the solutions to the Hamiltonian given in equation (37) are complex-valued, wave functions that depend on spatial coordinates, time, and integer parameters. The wave function solutions of the Hamiltonian are given by:

$$\Psi_n(\mathbf{x}, t) = |\Psi_n(t)\rangle \quad (38)$$

where the integer parameter n is referred to as the “quantum number.” In general, there are an infinite number of solutions $|\Psi_n(t)\rangle$, referred to as “eigenfunctions,” “eigenstates,” or “states,” that contain all the information that can be determined about the state of the system such as position, momentum, and energy. The set of solutions $\{|\Psi_n(t)\rangle\}$ given by equation (38) form a basis for a Hilbert space, and therefore, given any two eigenfunctions $|\Psi_n(t)\rangle$ in the Hilbert space $\{|\Psi_n(t)\rangle\}$:

$$\langle \Psi_i(t) | \Psi_j(t) \rangle = \int \Psi_i^*(x,t) \Psi_j(x,t) dx = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

(39)

where Ψ_i^* is the complex conjugate of Ψ_i .

- 5 The Hamiltonian given by equation (38) can also be rewritten as an eigenvalue problem in the form:

$$H|\Psi_n(t)\rangle = \left(\sum_k -\frac{\hbar^2}{2m_k} \frac{\partial^2}{\partial x_k^2} + V \right) |\Psi_n(t)\rangle = \lambda_n |\Psi_n(t)\rangle$$

(40)

10

where λ_n is the energy of the system in the state $|\Psi_n(t)\rangle$ and is referred to as an “eigenvalue.” Prior to a measurement of the quantum mechanical system characterized by the Hamiltonian, the system is assumed to be represented by a linear combination of state functions $|\Psi_n(t)\rangle$ as follows:

15

$$|\Omega(t)\rangle = \sum_i c_i |\Psi_i(t)\rangle$$

(41)

- 20 where $|\Omega(t)\rangle$ is the state function for the entire system and c_i are complex-valued coefficients. When a measurement of the system is made, the system is observed in only one of the eigenstates $|\Psi_i(t)\rangle$ in equation (41), where the probability of observing the system in the state $|\Psi_i(t)\rangle$ is the square of the coefficient, $|c_i|^2$.

- 25 The potential energy function V in equation (40) characterizes the particular physics of the system under observation such as the rotational and/or vibrational motion of a molecule, or the electronic states of an atom or molecule. For example, substituting a potential V characterizing the vibrational behavior of a diatomic

molecule, referred to as V_{vib} , into the Hamiltonian, gives the vibrational eigenstates $|\Psi_n(t)\rangle$ and the accompanying vibrational energy eigenvalues λ_n .

Figures 10A-C illustrate the Morse potential and spectrum for a hypothetical diatomic molecule. The vibrational potential of a diatomic molecule may be approximated by the Morse potential given by:

$$V_{vib} = D_e (1 - \exp(-\kappa r))^2$$

where D_e is the energy required to completely disassociate the diatomic molecule, κ is the curvature at the potential minimum, and r is bond length between atoms 1002 and 1006 shown in Figure 10A. The Morse potential characterizes the interaction between the atoms 1004 and 1006.

Figure 10B is an illustration of the Morse potential and the quantized vibrational energy levels. The Hamiltonian representing the vibrational behavior of a hypothetical diatomic molecule is given by:

$$H|\chi_n\rangle = \left(\sum_k -\frac{\hbar^2}{2m_k} \frac{\partial^2}{\partial x_k^2} + V_{vib} \right) |\chi_n\rangle = \lambda_n |\chi_n\rangle$$

where $\{|\chi_n\rangle\}$ is the set of vibrational states of the diatomic molecule, and $\{\lambda_n\}$ is the set of corresponding eigenvalues. The potential energy V_{vib} is plotted as a function of the inter-atomic bond distance r of atoms 1004 and 1006 in Figure 10A, where the energy is represented by the vertical axis, and the inter-atomic bond distance r is represented by the r -axis in Figure 10B. The lowest vibrational state is $|\chi_0\rangle$ with the accompanying energy eigenvalue λ_0 is referred to as the “ground state.”

In Figure 10B, the light sources 1012 and 1014 emit photons with energy given by:

$$h\nu_{i,j} = \lambda_{i,j}$$

where $\nu_{i,j}$ is the light frequency, and

$$\lambda_{i,j} = \lambda_j - \lambda_i$$

The hypothetical diatomic molecule absorbs a photon from source 1012 having frequency $\nu_{0,1}$, and the diatomic molecule experiences an increase in vibrational energy by an amount $\lambda_{1,2}$, as indicated by edge 1016. The transition from the ground state to a higher energy state is referred to as “excitation,” and the molecule is said to exist in an “excited state.” The hypothetical diatomic molecule now has an energy eigenvalue λ_1 and the state of the molecule has transitioned from the ground state $|\chi_0\rangle$ to the excited state $|\chi_1\rangle$. Edge 1018 shows a second transition from the state $|\chi_1\rangle$ to the excited state $|\chi_2\rangle$ resulting from absorption of a photon with frequency $\nu_{1,2}$ emitted from light source 1014. In this hypothetical example, the excited states are generated using light of a particular frequency. It should be noted that atoms and molecules may also experience excitation to higher quantum states by application of an electrical field.

Atoms and molecules also emit photons to the surroundings as they transition from an excited state to a lower energy state. The process of transitioning to a lower energy state is referred to as “relaxation” or “thermal relaxation.” Edge 1020 shows the relaxation of the hypothetical diatomic molecule from the excited state $|\chi_2\rangle$ with energy eigenvalue λ_2 to the state $|\chi_1\rangle$ with energy eigenvalue λ_1 . The transition from the state $|\chi_2\rangle$ to the state $|\chi_1\rangle$ results in the emission of a photon with the frequency $\nu_{1,2}$. Edge 1022 shows a transition from the state $|\chi_1\rangle$ to the ground state $|\chi_0\rangle$ and the emission of a photon with the frequency $\nu_{0,1}$.

The emitted photon can be observed by a detector to generate a spectrum of the different observed frequencies of light for an ensemble of diatomic molecules. Figure 10C is an illustration of the ensemble average of the spectrum of gas phase hypothetical diatomic molecules as described in relation to Figures 10A-B. The horizontal axis 1024 is the observed quantized vibrational frequencies for the ensemble and the vertical axis 1026 is the intensity associated with each vibrational

frequency. The spectrum of the hypothetical molecule reveals a tapering off in the intensity with higher vibrational frequencies. The intensity is a measure of the fraction of diatomic molecules in a particular vibrational state. Emission line 1028 has the largest intensity δ_0 and can be used to approximate the fraction of hypothetical
 5 diatomic molecules in the state $|\chi_0\rangle$. Therefore, according to equation (41) the state function describing the ensemble of diatomic molecules can be given by:

$$|\Omega\rangle = \sum_i c_i |\chi_i\rangle \cong \sum_i \sqrt{\delta_i} |\chi_i\rangle$$

where $|c_i|^2 \cong \delta_i$

10

E. Realization

Realization is the process of transforming the canonical, quantum, Hamiltonian operator in equation (40) into a physical operation of the hardware to solve for the function $\Phi(x)$. The function $\Phi(x)$ may be expanded in terms of an
 15 orthogonal basis set of p -valued Chrestenton functions spanning the Hilbert space γ_p given by:

$$B_p = \left\{ \phi_k(x) \middle| \phi_k(x) : [0, p^{N-1}] \rightarrow [0, p^{N-1}], k = 0, \dots, p^{N-1} \right\}$$

20 where the general form of a Chrestenton basis function is given by:

$$\phi^j(x) = \exp\left(\frac{2\pi}{p} i \sum_{s=0}^{N-1} j_{N-1-s} \cdot x_s\right)$$

with

25

$$x = \sum_{s=0}^{N-1} x_s \cdot p^{N-1-s}$$

and

$$j = \sum_{s=0}^{N-1} j_s \cdot p^{N-1-s}$$

5

Therefore, the interpolating functions $\Phi(x)$, written in terms of the orthogonal basis set B_p is given by:

$$\Phi(x) = \sum_{k \in [0, p^{N-1}]} \beta_k \phi_k(x) \quad (42)$$

10

where

$$\beta_k = \frac{\int_0^{p^N} \phi_k^*(x) \Phi(x) dx}{\int_0^{p^N} \phi_k^*(x) \phi_k(x) dx} \quad (43)$$

15

The set of coefficients $\{\beta_k\}$ is referred to as the spectrum of the function in equation (42), where the number of terms in the spectrum $\{\beta_k\}$ ranges from about 10^5 to 10^{12} . If the spectrum $\{\beta_k\}$ is known, then the function $\Phi(x)$ can be used to determine the encoding function $F(x)$ as described above in relation to steps 216 and 206 in Figure 2. The encoding function $F(x)$ can then be used to determine the original function $f(x)$ as described above in relation to steps 204 and 202 in Figure 2. The objective of the present invention is to determine a spectrum that approximates the spectrum $\{\beta_k\}$.

25

The hardware of the system is characterized by the hardware Hamiltonian operator referred to as " H_0 ." The realization step consists of constructing the

excitation-field Hamiltonian H_f , referred to as “ H_f ,” so that the hardware, when interacting with the excitation field H_f , satisfies the following condition:

$$5 \quad |H_0 + H_f - H| \leq \varepsilon \quad (44)$$

where $|\cdot|$ denotes an operator norm, and ε is an a parameter chosen to satisfy the precision requirements for the computation.

In general, the eigenvalues of the composite Hamiltonian $H_0 + H_f$ approximate
 10 the spectrum $\{\beta_k\}$ of the function given by equation (42). Let $\{\lambda_k | k=1, \dots, N\}$ be the subset of the eigenvalues of $H_0 + H_f$ with the corresponding eigenstates $\{|\Psi_k(t)\rangle | k=1, \dots, N\}$ so that if

$$15 \quad (H_0 + H_f)|\Psi_k(t)\rangle = \lambda_k |\Psi_k(t)\rangle \quad (45)$$

then

$$|\lambda_k - \beta_k| \leq \frac{\varepsilon}{N}$$

(46)

20 and

$$\int_0^\infty |\phi_k(x) - |\Psi_k(x)\rangle| dx \leq \varepsilon$$

(47)

25 where ϕ_k is a Chrestenton basis function. Therefore, the interpolating function $\Phi(x)$ given in equation (42) can be approximated as follows:

$$\Phi(x) \cong \sum_k \sqrt{\lambda_k} \phi_k(x) \quad (48)$$

Thus, a discrete function can be approximated by exciting the hardware system H_0 , according to the excitation-field Hamiltonian H_f to obtain the set of coefficients for equation (48). The encoding function can then be constructed from equation (48) which in turn can be used to determine the Class I functions, as described above in relation to reversing steps 206 – 202 in Figure 2.

II. The Functional Architecture for Quantum Computing

A. Function Input

The function input 302 in Figure 3 symbolically determines the excitation-field Hamiltonian H_f via the steps 202 - 214 described above in relation to Figure 2. The process of symbolically determining the excitation-field Hamiltonian H_f is referred to as the “compilation process.”

B. Excitation Generator

The excitation generator 303 in Figure 3 receives the symbolic excitation-field Hamiltonian H_f from the function input 302 and converts the symbolic description of H_f into a physical implementation to be carried out by the quantum processor. The excitation generator addresses each node of the quantum processor according to the desired state behavior given by:

$$i\hbar \frac{\partial}{\partial t} |\Psi_j(t)\rangle = (H_0 + H_f) |\Psi_j(t)\rangle$$

where $|\Psi_j(t)\rangle$ is the state function of the j th node in the quantum processor.

C. Quantum Processor

Figure 11 is an illustration of a quantum processor 304 in Figure 3 having a two-dimensional array of nodes representing one of many possible embodiments of the present invention. The nodes 1102 are chemical polymers linked to adjacent nodes in the lattice via quantum lateral bonds 904 and quantum forward bonds 1106. The boundaries 1106 and 1108 provide an insulating barrier to prevent the electric field from leaking out to the surrounding components of the system. The quantum processor also contains an excitation field 1112 that implements the excitation-field H_f . Located at opposite ends of the quantum processor are reflective plates 1114 and 1116 that reflect the excitation field back onto the lattice of nodes 1102.

The lattice of nodes is not limited to the rectilinear nodal arrangement shown in Figure 11. In alternate embodiments, the nodes may be arranged to provide a higher density of nodes, as, for example, by offsetting the nodes in adjacent rows to produce a more closely packed arrangement of nodes. In alternate embodiments, the nodes may be arranged in any number of different three-dimensional lattice structures, as, for example, the simple cubic, body-centered cubic, face centered cubic, primitive face centered cubic, simple hexagonal, hexagonal closest packed, rhombohedral etc. Figure 12 shows a simple cubic lattice of equally spaced nodes 1202, reflective plates 1204 and 1206, and insulating boundaries 1208 and 1210 that represents one of many possible embodiments of the present invention.

The Hamiltonian realizing the quantum processor prior to receiving the excitation-field H_f from the excitation generator is given by:

$$H = H_0$$

where H_0 is the hardware Hamiltonian. Prior to application of the excitation field, the lattice nodes are in the ground vibrational state. After the quantum processor receives the excitation-field H_f from the excitation generator, the quantum processor is characterized by the Hamiltonian given by:

$$H = H_0 + H_f$$

Figure 13 shows a hypothetical excitation-field H_f applied to the hypothetical quantum processor shown in Figure 11. The excitation-field H_f originates at the excitation field 1302 and sweeps across the quantum processor in the direction 1304. The excitation field wave front 1306 passes over nodes in their ground state exciting
 5 each node to a higher vibrational and/or electronic state. The nodes in the ground state are identified by the open circles 1308, and the nodes in an excited state are identified by the solid circles 1310. The rotational quantum states of the nodes are not accessed because each node is bound in the lattice and cannot freely rotate.

Figure 14 is a diagram of the two-level automaton for a single node 1402. The
 10 lower automaton 1404 provides a node's view of the quantum processor consisting of the transition selector 1406, the state transition 1408, and current state 1410. The upper automaton 1412 represents the quantum computing paradigm as described above by the function input 302 in Figure 3 and consists of the probability state 1414 and the probability transition 1416. The input selector function 1418 is an interface
 15 function, carried out by the excitation generator 303 in Figure 3, that models the interaction of the node 1402 with the excitation field 1418, the neighboring states 1420 of the other nodes in the lattice, probability states 1414, and the current state 1410.

20 1. The Lower Level Automaton

In the lower automaton 1404, the state transition 1408 characterizes the programmable discrete spectrum of the node and is controlled by the mixing state probability density transition computed in the upper automaton 1412. The transition selector 1406 in the lower automaton 1404 executes the commands issued by the
 25 upper automaton 1412. Each node 1402 has a set of pure states given by:

$$S = \left\{ \left| \Psi_{j_k}(t) \right\rangle \middle| k \in N_p, N_p \text{ is finite} \right\}$$

(50)

30 where $\left| \Psi_{j_k}(t) \right\rangle$ are the pure states of the node, j is the index of the j th node in the quantum processor, k is the quantum number, and N_p is number of available quantum

states. Each node in the lattice begins in the ground state $|\Psi_{j_1}(t)\rangle$ and is excited to a higher states $|\Psi_{j_k}(t)\rangle$ by the excitation-field H_f , as described above in relation to Figure 11. After the node has been excited to a higher state by the excitation-field H_f , the node spontaneously undergoes relaxation to a lower energy state in S . After a period of time determined by the control and scheduling system 301 in Figure 3, the lattice of nodes are subsequently re-excited by another excitation-field H_f generated by excitation generator 303. The time duration between the excitation fields is referred to as the “update time” and is denoted by Δ . The update time Δ of the node is determined to be larger than 10 times the maximum relaxation time of any of the state transitions in the lower automaton.

Each node in the lattice may transition between one or more states in S during the update time Δ and referred to as “chattering.” The chattering effect is described as a combination of the pure states in S and is given as follows:

$$|\Psi_j(\tau)\rangle = \begin{cases} |\Psi_{j_1}(\tau)\rangle & \tau \in I_{j_1}(t) \\ |\Psi_{j_2}(\tau)\rangle & \tau \in I_{j_2}(t) \\ \vdots & \\ |\Psi_{j_{N_p-1}}(\tau)\rangle & \tau \in I_{j_{N_p-1}}(t) \\ |\Psi_{j_{N_p}}(\tau)\rangle & \tau \in I_{j_{N_p}}(t) \end{cases} \quad (51)$$

where

$$I_{j_k}(t) = \left[t + \sum_{l=1}^{k-1} \Delta_{j_l}(t), t + \sum_{l=1}^k \Delta_{j_l}(t) \right) \quad (52)$$

are the semi-open intervals of the update time interval $[t, t + \Delta)$ and give the duration that node j spends in the pure state $|\Psi_{j_k}(\tau)\rangle$. The time interval $\Delta_{j_k}(t)$ is a function

of the characteristic excitation or relaxation times associated with the state. See *Fields and Nodes: Field Theory and Dispersion Relations*, by K. Nishijima, Benjamin, Inc., NY, 1969.

Figure 15 illustrates the relationship between the chattering combination in equation (51) and the semi-open time intervals in equation (52). The pure states $|\Psi_{j_k}(t)\rangle$ in the chattering combination (51) are arranged vertically in order of increasing energy. The semi-open intervals $I_{j_k}(t)$ appearing in the chattering combination (51) and defined in equation (52) are displayed graphically below the pure states $|\Psi_{j_k}(t)\rangle$. Each semi-open interval $I_{j_k}(t)$ is of length Δ_{j_k} and represents the amount of time the j th node remains in the pure state $|\Psi_{j_k}(t)\rangle$ during the update time interval $[t, t + \Delta)$. For example, the j th node occupies the pure state $|\Psi_{j_2}(t)\rangle$ for the length of time Δ_{j_2} in the interval $I_{j_2}(t)$. The sum of the time intervals Δ_{j_k} can be expressed as follows:

$$\sum_k \Delta_{j_k}(t) = \Delta \quad (53)$$

The chattering coefficients can then be defined as:

$$\alpha_{j_k}(t) = \frac{\Delta_{j_k}(t)}{\Delta} \quad (54)$$

which allows equation (53) to be written as:

$$\sum_k \alpha_{j_k}(t) = 1 \quad (55)$$

A chattering node may spontaneously transition to a lower state in S giving off radiation to the surroundings, or the node may absorb radiation emitted by other nodes in the system causing the node to transition to higher energy states in S . Figure 16 illustrates chattering for a single hypothetical node using three pure states in the chattering combination (51) arranged in order of increasing along the energy-axis 1602. First, assume the node begins in state $|\Psi_{j_k}(\tau)\rangle$ 1604 with energy λ_{j_k} 1606. The node in the state $|\Psi_{j_k}(\tau)\rangle$ 1604 spontaneously transitions to the lower state $|\Psi_{j_{k-1}}(\tau)\rangle$ 1608 with energy $\lambda_{j_{k-1}}$ 1610 emitting energy $h\nu_{j_{k-1},j_k}$ to the surroundings, as indicated by edge 1612. Second, the node absorbs energy $h\nu_{j_{k-1},j_k}$ from the surroundings, as indicated by edge 1614, the node transitions back to the state $|\Psi_{j_k}(\tau)\rangle$ 1602 with energy λ_{j_k} 164. Third, the node absorbs energy with frequency $h\nu_{j_k,j_{k+1}}$ from the lattice surroundings, and transitions to the state $|\Psi_{j_{k+1}}(\tau)\rangle$ 1616 with energy $\lambda_{j_{k+1}}$ 1618, as indicated by edge 1620.

For each update time interval $[t, t + \Delta)$, the transition selector 1406 receives a command from the upper automaton 1412 in Figure 14, which is comprised of an ordered tuple of chattering coefficients $\{\alpha_{j_k}(t) | j_k \in N_p\}$. The Transition Selector 1406 then computes the chattering combination of mixed states $|\Psi_{j_k}(t)\rangle$ according to equation (51).

20

2. The Upper Level Automaton

In general, there is not enough information to determine a specific state function that characterizes the lattice or any of the lattice nodes. Therefore, the best description of the computation is a probabilistic description. In the quantum formalism, the description is referred to as the "probability density description." See *Probabilistic Model of Quantum Relation*, V. F. Weisskopf and E. Wigner, Physik 63, pp. 54-62, 1930.

25

Consider again the set of pure states in the lower automaton given by equation (54). Let p_j be the probability that the j th node is in the pure state $|\Psi_j(t)\rangle$. Then the probability density operator is defined by:

$$\rho(t) = \sum_j p_j |\Psi_j(t)\rangle \langle \Psi_j(t)| \quad (56)$$

and can be used to determine the probability of a particular quantum state. Differentiating equation (56) and substituting equation (40) gives the differential operator:

$$i\hbar \frac{\partial}{\partial t} \rho = H \cdot \rho - \rho \cdot H \quad (57)$$

The operator on the right-hand side of equation (57) is termed the “commutator” and can be written as $[H, \rho]$. Thus equation (57) becomes:

$$i\hbar \frac{\partial}{\partial t} \rho = [H, \rho] \quad (58)$$

Equation (58) characterizes the computation carried out by the upper automaton 1412 in Figure 14. The results obtained by the upper automaton 1412 is the probability of the node being in the state $|\Psi_j(\tau)\rangle$.

Given the density operator in equation (56) the ensemble average for the lattice can be determined by the trace of the matrix $\rho \cdot C$ and is given by the expression:

$$\langle C \rangle = \text{trace}(\rho \cdot C) \quad (59)$$

where C can be any quantum mechanical operator. For example, the vibrational
 5 Hamiltonian characterizing the vibrational motion of the nodes is substituted for the
 operator C to give the ensemble average of the vibration states. The ensemble
 average calculated for an operator C is carried out by the transducer 305 in Figure 3.

Notice that the state function $|\Psi_j(t)\rangle$ is given as a chattering combination in
 equation (51) and can equivalently be written as a linear combination of the states in S
 10 as follows:

$$|\Psi_j(t)\rangle = \sum_k \alpha_{j_k}(t) |\Psi_{j_k}(t)\rangle \quad (60)$$

15 where the coefficients $\alpha_{j_k}(t)$ are the chattering coefficients given by equation (54).
 The equivalence between the linear combination in equation (60) and the chattering
 combination in equation (51) is established in the following version of the Chattering
 Lemma:

Let $S = \{|\Psi_{j_k}(t)\rangle | k \in N_p, N_p \text{ is finite}\}$ and let $\hat{S}$ be the set of chattering
 20 combinations of S . Let a real positive number ε be given. There exists state
 functions $|\Theta_j\rangle \in \hat{S}$, defined for each tuple given by:

$$\left\{ \alpha_1, \dots, \alpha_{N_p} \mid \alpha_l \geq 0, \sum_{l=0}^{N_p} \alpha_l = 1 \right\} \quad (61)$$

25 such that

$$\max_t \left| \int_0^t \left| \left| \Theta_j(\tau) \right\rangle - \sum_{k=1}^{N_p} \alpha_{j_k} \left| \Psi_{j_k}(\tau) \right\rangle \right| d\tau \right| < \varepsilon$$

(62)

for all $\{\alpha_1, \dots, \alpha_{N_p}\}$.

5 The Chattering Lemma says that every chattering combination of the form of equation (51) on a set S can be realized as a linear combination of elements from S with an arbitrarily small error. Under strong continuity assumptions of the state functions, the converse of the lemma is also true. By choosing the boundary conditions in the lattice appropriately, the continuity assumption is not limiting.

10 Although the linear combination given in equation (60) can be used to represent the state function $|\Psi_j(t)\rangle$ of the j th node, formulating the state function in the form of the chattering combination in equation (51) allows for formulating the sequence of excitation steps.

15 3. The Input Selector Function

The Input Selector Function 1418 in Figure 14 utilizes the information received from the current state 1410 of the node, the probability state 1414, models the interaction of a node in the lattice with the excitation field 1418 and with the electric fields generated by neighboring nodes 1420 to generate H_f . Figure 17 shows examples of the kinds of lattice node interactions. For the sake of simplicity, only nearest neighbor interactions are shown in Figure 17, however, the present invention can accommodate other node interactions. Oval 1702 identifies a dipole moment interaction resulting from a node 1704 interacting with a nearest neighboring node 1706. Oval 1708 identifies a tripole moment interaction created by a node 1710 interacting with two neighboring nodes 1712 and 1714. Oval 1716 identifies a quadrupole interaction between a node 1718 neighboring nodes 1720, 1722, 1724, and 1726.

In order to distinguish the excitation field from the electric fields generated by other nodes in the lattice, the excitation-field Hamiltonian H_f can be written as the sum of two terms as follows:

30

$$H_f = H_e + H_l \quad (63)$$

5 where H_e is the interaction of the node with the excitation field, and H_l is the interaction of the node with other nodes in the lattice. The interaction Hamiltonian H_l for a given node can be written as:

$$H_l |\Psi_j(t)\rangle = g_\Psi(H_l^i, \dots, H_l^k) |\Psi_j(t)\rangle \quad (64)$$

10 where for the j th node, H_l^j is the interaction Hamiltonian of the node j with “neighboring” nodes i_j , and g_Ψ is referred to as the “neighboring function” at the current state of the node $|\Psi_j(t)\rangle$. Figure 18 illustrates the notation for a hypothetical
15 interaction Hamiltonian H_l given by:

$$g_\Psi(H_l^{1j}, H_l^{2j}, H_l^{3j})$$

Edges 1801-1803 identify the interaction Hamiltonians H_l^{1j} , H_l^{2j} , and H_l^{3j} between
20 the node j and the neighboring nodes 1_j 1804, 2_j 1805, and 3_j 1806, respectively. The neighborhood function g_Ψ represents the locus structure of interaction of the node j in the lattice at the current state $|\Psi_j(t)\rangle$.

The structure of interaction Hamiltonian H_l can be determined by constructing the Lagrangian associated with the lattice and applying the canonical quantization
25 procedure discussed above in subsection I. D. The elements of carrying out this task are well known but the details depend on the physical characteristics of the lattice. The approach consists of defining potentials to characterize the lattice node interactions. For example, a dipole interaction as shown by oval 1702 in Figure 17 may be characterized by a potential of the form:

$$V(x, y, t) = d_{x-y}(t) \frac{1}{r_{ij}} \quad (65)$$

5 where $d_{x-y}(t)$ is a relaxation function and r_{ij} is the distance between nodes i and j .

For purposes of analysis and to determine a detailed formulation of the realization step described above in section I.F, assume that a finite set of primitive excitation Hamiltonians is given by:

$$10 \quad T = \{H_e^i | i = 1, \dots, N_e\} \quad (66)$$

where N_e is the number of excited states available to the node j . The primitive excitation Hamiltonians given in equation (66) can be realized and implemented by
 15 chattering among the elements in the set T over the update time Δ . Chattering among the elements of T is given by the chattering equation:

$$H_e |\Psi(\tau)\rangle = \begin{cases} H_e^{i_1} |\Psi(\tau)\rangle & \tau \in I_{i_1}(t) \\ H_e^{i_2} |\Psi(\tau)\rangle & \tau \in I_{i_2}(t) \\ \vdots & \\ H_e^{i_{N_e-1}} |\Psi(\tau)\rangle & \tau \in I_{i_{N_e-1}}(t) \\ H_e^{i_{N_e}} |\Psi(\tau)\rangle & \tau \in I_{i_{N_e}}(t) \end{cases} \quad (67)$$

20

where the sets $I_{i_k}(t)$ are defined by equation (52) above and $|\Psi(\tau)\rangle$ is a state function. The probabilistic resonance chattering equation (67) is central to the implementation of the quantum processor. The idea is to induce a probability distribution ρ on the states of the nodes in the lattice so that the realization criterion is
 25 satisfied.

Figures 19 and 20 show how a hypothetical quantum processor can be used to obtain an approximation to the function Φ . Figures 19A-C illustrate a single hypothetical run of the excitation-field Hamiltonian H_f used to obtain a hypothetical spectrum from the quantum processor. In Figure 19A, all of the nodes of the quantum processor 1902 begin in the ground state and can be characterized by the hardware Hamiltonian H_0 . The detector 1904 converts the incident radiation into an electrical current which is used to count the number of photons. In Figure 19B, the excitation-field H_f impinged upon the quantum processor 1802 via the excitation generator 303 in Figure. The excited nodes emit light radiation in the form of photons that strike the detector 1904, as indicated by radiating lines 1906, 1908, and 1910. Figure 19C shows a plot of the frequencies along the horizontal axis 1914 and the corresponding intensities along the vertical axis 1916 for the single application of the excitation-field Hamiltonian H_f . The radiating lines 1906, 1908, and 1910 are associated with nodes that emit radiation of the same frequency ν_k and are included in determining the intensity value λ_k . The resulting spectrum intensities $\{\lambda_1, \lambda_2, \dots\}$ shown in Figure 19C are stored in the coherent memory 306 in Figure 3.

Figure 20 shows example hypothetical spectra and the average spectrum after n hypothetical runs of the excitation-field H_f . The n spectra indicated by spectrum 2002, 2004, and 2006 are used to compute an average spectrum 2008. Approximately 10^{12} runs are used to determine the average spectrum 2008 where each run takes about .2 nanoseconds to complete. Rather than store the spectrum for each run, a running average of the intensities, referred to as $\{\bar{\lambda}_1, \bar{\lambda}_2, \dots\}$, is stored in the coherent memory 306 in Figure 3. The average spectrum can be used to approximate the function $\Phi(x)$ in equation (42) by the following:

25

$$\Phi(x) = \sum_{k \in [0, p^{N-1}]} \sqrt{\bar{\lambda}_k} \phi_k(x) \quad (68)$$

D. The Read-Out Period

The read-out period, referred to as " ζ ," is the amount of time the transducer devotes to reading the light radiation emitted from the quantum processor in step 410 in Figure 4. If the transducer collects singles from the quantum processor for too long, there is a danger of *de coherence* due to emissions caused by nodal interaction characterized by the interaction Hamiltonian H_I . For example, a computation carried out in the lattice of the quantum processor begins with a propagating probabilistic wave train as described above in relation to Figure 13. Figure 21 is an illustration of a forward propagating wave train 2102 and a backward propagating wave train 2104. After the initial excitation-field H_f impinges upon the lattice, the forward propagating wave train 2102 sweeps across the lattice of nodes, where some of the energy stored in the excitation field is converted into nodal excitation energy. The nodes respond after excitation by relaxing to lower energy states by releasing light radiation $h\nu$ 2108. The excitation field sloshes back and forth in the form of forward and backward moving wave trains 2102 and 2104, respectively, until the entire excitation field H_f has been consumed and converted to light energy via nodal excitation. During this time period radiation continues to be emitted as a result of nodal interaction characterized by the interaction Hamiltonian H_I of different energies $h\nu'$ 2110, which can distort the intensity-versus-frequency spectrum. Therefore, in order to limit the amount of light radiation emitted as a result of nodal interaction from contributing significantly to the intensity-versus-frequency spectrum, the readout period ζ for the transducer is considerably shorter than the amount time it takes for the excitation field to dissipate.

Figure 22 is a control-flow diagram depicting determination of the transducer read-out period ζ representing one of many possible embodiments of the present invention. In step 2202, the quantum processor lattice is at an initial probabilistic state characterized by the hardware Hamiltonian H_0 . In step 2204, the programmed excitation-field H_f determined by the function input 302 and realized by the excitation generator 303 in Figure 3 impinges upon the quantum processor lattice of nodes. In step 2206, after a read-out period ζ has elapsed, the transducer 305 in Figure 3 transmits the spectrum intensities and corresponding frequencies to the coherent memory 306 in Figure 3. In step 2208, if the resulting equation (68) does not approximate the encoding function within parameters set by the operator, then in step

2210, the read-out period ζ is extended to a longer read-out period ζ_l and the computation is started again from the initial probabilistic state. In step 2208, if the resulting average spectrum intensities $\{\bar{\lambda}_1, \bar{\lambda}_2, \dots\}$ are satisfactory, then the computation is complete and an approximation to the encoding function has been
 5 determined.

The read-out period ζ cannot be extended arbitrarily because thermal relaxation mechanisms inherent in a quantum mechanical system will eventually induce *de coherence*. In order to demonstrate this, consider the discrete spectrum $\{E_i\}$ of the hardware Hamiltonian H_0 given by the eigenvalue equation:

10

$$H_0 |\Psi_i(t)\rangle = E_i |\Psi_i(t)\rangle$$

(69)

where $|\Psi_i(t)\rangle$ are the corresponding eigenfunctions. Using equation (58), the
 15 transition probability matrix elements ρ_{ij} satisfies the expression:

$$i\hbar \frac{\partial}{\partial t} \rho_{ij}(t) = (E_i - E_j) \cdot \rho_{ij} + [H_f, \rho]_{ij}$$

(70)

20 where the matrix elements are given by:

$$\rho_{ij} = \langle \Psi_i(t) | \rho | \Psi_j(t) \rangle = \int_0^\infty \Psi_i^*(x, t) \rho \Psi_j(x, t) dx$$

(71)

25 where $\Psi_i^*(x, t)$ is the complex conjugate of the function $\Psi_j(x, t)$. If the system is in the state $|\Psi_j(t)\rangle$ at time t equal to zero, then the presence of the relaxation Hamiltonian H_l given by equation (54) causes the corresponding probability density

matrix elements ρ_{ij} to decay exponentially with time. See *Fields and Nodes: Field Theory and Dispersion Relations*, K. Nishijima, Benjamin, Inc., N.Y., 1969. For small values of t , the diagonal matrix elements ρ_{jj} are the largest terms in the matrix representation of the probability density operator. Assuming no excitation and using
 5 equation (57), the term ρ_{ij} satisfies the following equation:

$$i\hbar \frac{\partial}{\partial t} \rho_{ii} = \sum_k \left((H_I)_{jk} \rho_{jk} - \rho_{jk} (H_I)_{kj} \right) \quad (72)$$

10 where

$$i\hbar \frac{\partial}{\partial t} \rho_{ji} \cong (E_j - E_i) \cdot \rho_{ji} - \rho_{ij} \cdot (H_I)_{ji} \quad (73)$$

for i not equal to j . The solution of equations (72) and (73) is given by:
 15

$$\rho_{ij}(t) = e^{-t/\xi} \quad (74)$$

where ξ is the relaxation time that depends on the physical characteristics of the
 20 quantum processor lattice. The relaxation time is the time required for the wave train to achieve *de coherence*. Therefore, the effect of the inter-node excitation Hamiltonian H_I is to randomize the state transitions in each node corrupting the computation described above. Therefore, the read-out period ζ must be chosen so that this effect is acceptable. In other words, the read-out period is selected so that

25

$$\zeta \ll \xi \quad (75)$$

The upper bound ξ is used in the design specification of the quantum processor and in the computability analysis.

Although the present invention has been presented in terms of a particular embodiment, it is not intended that the invention be limited to this embodiment.

- 5 Modifications within the spirit of the invention will be apparent to those skilled in the art.

The foregoing description, for purposes of explanation, used specific nomenclature to provide a thorough understanding of the invention. However, it will be apparent to one skilled in the art that the specific details are not required in order to practice the invention. The foregoing descriptions of specific embodiments of the present invention are presented for purposes of illustration and description; they are not intended to be exhaustive or to limit the invention to the precise forms disclosed, obviously many modifications and variations are possible in view of the above techniques. The embodiments were chosen and described in order to best explain the principles of the invention and its practical applications and to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalents:

10

15

CLAIMS

What is claimed is:

1. A method for computing a value comprising:
 - 5 encoding a program of computable functions to describe computation of the value to be computed;
 - continualizing the encoded program;
 - expressing the continualized, encoded program as a differential operator;
 - instantiating the differential operator in a physical medium; and
 - 10 extracting from the physical medium a solution for the continualized, encoded program.
2. The method of claim 1 wherein the encoding a program of computable functions further includes:
 - 15 for each point $(x_0, x_1, \dots, x_{N-1})$ in the domain $S_1 \times S_2 \times \dots \times S_N$ of computable functions a mapping given by:

$$F: [0, 1, \dots, p^{N-1}] \rightarrow [0, 1, \dots, p]$$

20 where

$$x = \sum_{s=0}^{N-1} x_s \cdot p^{N-1-s}$$

and p is a natural number defined by:

25

$$p = \max_{i,d} \left\{ \{N_i | i = 1, \dots, N\}, N_d \right\} + 1$$

and

$$F_i(x) = \begin{cases} f_i(x_0, \dots, x_{N-1}) & \text{if defined} \\ 0 & \text{otherwise} \end{cases}$$

3. The method of claim 2 wherein continualizing the encoded program further includes determining an interpolating function $\Phi(x)$ of the encoded program $F(x)$.

5

4. The method of claim 3 wherein continualizing the encoded program further includes parameterizing the interpolating function $\Phi(x)$, by the recursion relation $x_{n+1} = x_n + a_n h(x_n)$, where $\{x_0, x_1, \dots\}$ is a sequence of real vectors.

10 5. The method of claim 4 wherein the iteration $x_{n+1} = x_n + a_n h(x_n)$ is transformed into a first-order, time-dependent differential equation $\dot{x}(t) = h(x(t)) - \dot{g}(t, a, b)$ where the solution vector $x(t)$ defines a path on a carrier manifold.

15 6. The method of claim 5 wherein the first-order, time-dependent differential equation $\dot{x}(t) = h(x(t)) - \dot{g}(t, a, b)$ is used to formulate a Lagrangian $L(x, \dot{x}, t)$.

7. The method of claim 6 wherein the Lagrangian $L(x, \dot{x}, t)$ is converted to the classical Hamiltonian given by:

20

$$H(x, p, t) = \sum_k \frac{p_k^2}{2m_k} + V = E$$

where V is the potential energy function of x or both x and t , and E is the energy of the system

25

8. The method of claim 7 wherein expressing the continualized, encoded function as a differential operator further includes converting the classical Hamiltonian $H(x, p, t)$ into a quantum, canonical, Hamiltonian operator given by:

$$5 \quad H\left(x, \frac{\hbar}{i} \frac{\partial}{\partial x}, \frac{\partial}{\partial t}\right) = \sum_k -\frac{\hbar^2}{2m_k} \frac{\partial^2}{\partial x_k^2} + V = i\hbar \frac{\partial}{\partial t}$$

where $\frac{\partial^2}{\partial x_k^2}$ and $\frac{\partial}{\partial t}$ are the coordinate and time differential operators.

9. The method of claim 8 wherein instantiating the differential operator in a physical medium further includes realizing the quantum, canonical, Hamiltonian operator.

10. The method of claim 9 wherein instantiating the differential operator in a physical medium further includes impinging a quantum processor.

11. The method of claim 10 wherein instantiating the differential operator in a physical medium further includes collecting the light radiation emitted from the quantum processor.

12. The method of claim 11 wherein collecting the emitted light radiation further includes converting the emitted light radiation into a coherent spectrum of intensities versus vibrational frequencies.

13. The method of claim 12 wherein instantiating the differential operator in a physical medium further includes storing a running average of the spectrum of vibrational intensities given by:

$$\{\bar{\lambda}_1, \bar{\lambda}_2, \dots\}$$

14. The method of claim 13 wherein extracting from the physical medium a solution for the continualized, encoded program further includes constructing a polynomial approximation of the function $\Phi(x)$ given by:

$$\Phi(x) = \sum_{k \in [0, p^{N-1}]} \sqrt{\lambda_k} \phi_k(x)$$

15. The method of claim 14 wherein extracting from the physical medium a solution further includes determining the encoded program $F(x)$ from the polynomial approximation of the function $\Phi(x)$.

16. The method of claim 15 wherein extracting from the physical medium a solution further includes determining computable functions from the encoded program $F(x)$.

17. A quantum processor comprising:
- one or more nodes;
 - one or more bonds connecting the nodes to form a lattice;
 - one or more reflector plates; and
 - one or more insulating boundaries.