SHIFT RULE FOR GRADIENT DETERMINATION IN PARAMETERISED QUANTUM EVOLUTIONS

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] This disclosure is concerned with hybrid quantum/classical computing utilising parameterised quantum evolutions, and has particular relevance to determining the gradient, or partial gradient, of a cost function at a parameter value.

Description of the Related Technology

[0002] In the context of noisy intermediate-scale quantum (NISQ) computing, there is a particular interest in quantum algorithms which employ parameterised quantum evolutions, wherein a controlled quantum system is subject to a, or many, quantum evolutions parameterised by a set of real-number control values. Such parameterised quantum evolutions can be found in a variety of computing approaches: hybrid classical-quantum algorithms, such as quantum variational eigensolvers and quantum approximate optimization algorithms, and algorithms used in digital-analog quantum simulators and digital-analog quantum computation. Analog quantum simulators are based on a controllable quantum system whose effective dynamics map to a model system of interest. In digital quantum computation schemes, algorithms are based on sequential application of one and two-qubit gates. In digital-analog quantum computing, both digital and analog approaches are employed. These approaches can realise simulations of molecules and condensed matter materials, quantum machine learning and other quantum artificial intelligence, as well as some approaches to quantum-based combinatorial optimization.

[0003] In general, the approach of quantum algorithms employing such parameterised quantum evolutions is to subject the controlled quantum system to the parameterised quantum evolution for a set of control parameter values and make a measurement on the quantum system. This measurement represents in itself or determines a loss or cost function which is minimised through an optimisation process. An updated set of control parameter values is determined by a classical algorithm running on a classical computing system based on the measurements of the quantum system, and the parameterised quantum evolution executed again using these new control parameter values. Through an iterative process, the control parameter values are updated until that the quantum algorithm satisfies an optimisation condition.

[0004] One particular technique implemented by the classical algorithms running on the classical computing system is gradient descent e.g. Stochastic Gradient Descent, whereby the gradient of the cost, or loss, function with respect to a particular control parameter is used to inform the selection of new control parameter values. In the training of classical neural networks, gradient descent techniques outperform derivative-free optimisation methods, and this appears also to be the case in the realm of quantum algorithms employing parameterised quantum evolutions.

[0005] One particular way of determining the derivative is through numerical differentiation. Such a method may use, for example, a finite difference approximation, whereby the evolution is performed with a control parameter set to a first value, and then run again with the control parameter set to a second value. The difference in the value of the cost or loss function (measured as an expectation value over many measurements of the quantum system) divided by the difference between the first and second values of the control parameter provides an approximation of the gradient around this point. Due to fundamental randomness in quantum mechanical measurements, a procedure for processing measurement outcomes constitutes an estimator, whose bias (approximation error) and variance are important. For the finite difference approximation, the approximate error decreases with smaller differences in parameter control values, but the variance increases. This limits the effectiveness of this approach, as the variance of the estimator directly influences how resource-intensive a hybrid quantum-classical computation will be, with higher variance demanding more iterations in the stochastic optimization algorithm.

[0006] Another particular way of determining the derivative is to obtain an estimate for the exact (approximation error free) derivative through implementation of a parameter shift rule. Some quantum evolutions, such as the quantum gates typically implemented in parameterised quantum circuits, admit parameter shift rules which allow for the exact (up to available numerical precision) derivative at a target control parameter value to be estimated with a constant variance depending only on the gate itself. This is done by running the quantum circuit at a number of control parameter values shifted from the target control parameter value, and forming a sum of the measurement results, weighted with real-number coefficients. The precise selection of the shifted control parameter values and their real-number coefficients is specific to the quantum gate being applied.

[0007] The parameter shift rule described above supposes that the parameter θ enters into the quantum evolution in the form of a unitary $e^{\frac{i\theta A}{\hbar}}$. However, this assumption does not hold in current NISQ approaches that aim to demonstrate a *quantum advantage* over classical computers. Instead, the parameter θ enters into the quantum evolution in the form $e^{\frac{i(\theta A+B)}{\hbar}}$. This can arise, for example, in the situation when the quantum system is subject to an always-on entangling Hamiltonian (e.g., an Ising Hamiltonian linking all qubits) represented by the B-component, and a single qubit drive is switched on for a short time, represented by the θA. This concept has been proposed and is used for several qubit technologies, with several different ways in which the two Hamiltonians can be related. The B-component can also occur due to unavoidable drift in the quantum control mechanisms. The presence of the B-component in the Hamiltonian prevents the above-mentioned parameter shift rule from being applicable.

[0008] To address this, there is a known stochastic parameter shift rule (SPSR) which determines control parameter values to obtain exact derivates from an arbitrary parameterised quantum evolution. However, the known SPSR requires the parametric quantum evolution to be modified beyond simply altering the value of the control parameters. This modification of the quantum evolution requires an update of the scheduling of the control pulses, which involves either resetting control electronics (e.g. by writing a new program onto a Field Programmable Gate Array), which is a time consuming process on the order of a second and thus unsuitable for iterative schemes such as stochastic gradient descent, or requiring a large number of different programs be stored on the FPGA, one for every parameter.

[0009] Additionally, in situations where the quantum evolution $e^{\frac{i\theta A}{\hbar}}$ cannot be switched on separately (in other words, where the B-component of the Hamiltonian cannot be switched off), the SPSR method cannot be implemented without an approximation error. To reduce that approximation error, the SPSR scheme demands higher and higher drive strengths for shorter and shorter durations, which at best poses difficulties for some quantum control methods. At worst, it runs into technical limitations relating to the Fourier Uncertainty Principle, as the drive signal has to be localised both in time and in frequency: The pulses used in quantum control are, generally speaking, as short as possible already, and making them orders of magnitude shorter is problematic.

[0010] There is therefore a desire for a method of obtaining derivatives in quantum evolutions that can handle perturbed parameterised evolutions due to, for example, unavoidable drifts or arising from necessary analog control, but overcomes the approximation error and does not require fundamental re-design of the underlying quantum evolutions.

SUMMARY

[0011] According to a first aspect of the present invention, there is provided a hybrid computing system comprising a quantum computing system and a classical computing system, the quantum computing system comprising a quantum system having one or more quantum devices. The hybrid computing system is configured to estimate a derivative of a parameter-dependent physical quantity that is dependent on a first control parameter θ of a parameterised quantum evolution executed by the quantum computing system. The classical computing system determines, based on an input bounding value and phase-correction value, a subset multiset of shift-values which define an equivalent subset multiset of trial control parameter values and corresponding weighting values and summation factors such that the trial control parameter values obey a targeted probability distribution. The quantum computing system executes the parameterised quantum evolution at the trial control parameter values and measures a parameter-dependent physical quantity for each trial control parameter value. On the classical computing system, the measurement results from these measurements are combined with the weighting values and summation factors to calculate at an estimate of a phase-correction-scaled derivative of the parameter-dependent physical quantity. In this way, the derivative can be provided to a gradient-descent optimiser to optimise the control parameters of a quantum algorithm.

[0012] In examples the phase-correction value is zero, such that the hybrid system calculates the estimate of the derivative of the expectation-value function, without phase-correction scaling.

[0013] In examples hybrid computing system may be further configured to calculate the estimate of the derivative of the expectation-value function from the estimate of the phase-correction-scaled derivative.

[0014] In examples the sub-multi-set of shift values is constructed by randomly sampling the set of all shift values according to the targeted probability distribution and the weighting values are set to unity.

[0015] In examples the sub-multi-set of shift values is constructed by randomly sampling a value the set of all shift values according to the targeted probability distribution and adding the value and its additive inverse to the sub-multi-set and the weighting values are set to unity.

[0016] In examples the sub-multi-set of shift values is constructed deterministically up to a target size, and the weighting values are selected according to the target probability distribution.

[0017] In examples the sub-multi-set of shift values is constructed semi-deterministically in such a way to minimise the variance of a random variable which approximates the targeted probability distribution.

[0018] In examples the classical computing system is further configured to pass each trial control parameter values through a parameter folding function such that no trial control parameter value differs from the first control parameter value by more than a parameter shift limit value.

[0019] In examples the parameter value function has the property $\tau(x) - x \in p\mathbb{Z} = \{pk | k \in \mathbb{Z}\}$, where $p := \frac{k}{\gamma}$, where p is a real number, k is a positive integer and γ is a is a common divisor of the eigenvalues of the first parameterised quantum evolution.

[0020] In examples, the parameter value function is

$$\tau(x) = \begin{cases} -c - ((-x)\%p), \text{if } x \leq -c - p; \\ x, \quad \text{if } x \in]-c-p, +c+p[; \\ +c + (x\%p), \text{if } x \geq +c + p. \end{cases}$$

where c is a real number, $c = m \cdot p$ where m is a positive integer, and where $\%p$ stands for the remainder in $[0, p[$ when dividing by p.

[0021] In examples, the input bounding value is based on the largest and smallest eigenvalue of the first parameterised quantum evolution.

[0022] Further features and advantages of the invention will become apparent from the following description of preferred embodiments of the invention, given by way of example only, which is made with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] Figure 1 shows a schematic diagram of an exemplary hybrid quantum-classical computing system which is used to perform a parameterised quantum evolution as part of a quantum algorithm and utilise the Nyquist-shift method described herein.

[0024] Figure 2 is a flow chart showing steps executed to perform a parameterised quantum evolution as part of a quantum algorithm.

[0025] Figure 3a is a flow chart showing steps performed to optimise a parameterised quantum evolution, the parameterised quantum evolution parameterised by a set of control parameters.

[0026] Figure 3b is a flow chart showing steps performed to optimise a parameterised quantum evolution using the Nyquist-shift method disclosed herein.

[0027] Figure 4 is a flow chart showing steps performed to measure matrix values of a unitary applied during a parameterised quantum evolution

[0028] Figure 5 is a flow chart showing steps performed in implementing the Nyquist-shift method described herein.

[0029] Figure 6 is a flow chart showing steps performed in implementing a Folding variant of the Nyquist-shift method described herein.

[0030] Figure 7 is a flow chart showing steps performed in implementing a variant of the Folding-Nyquist-shift method described herein.

DETAILED DESCRIPTION OF CERTAIN INVENTIVE EMBODIMENTS

[0031] Figure 1 schematically shows, by way of example, the main components of a hybrid quantum-classical computing system 100 comprising a quantum computation system 120 and classical computation system 110, the quantum computation system 120 and the classical computation system 110 cooperating to execute a hybrid classical-quantum scheme. The quantum computation system 120 includes quantum control hardware 122 and a controlled quantum system 140. The quantum control hardware 122 interacts with the controlled quantum system 140 in accordance with instructions from the classical computing system 110 so as to modify the quantum state of the controlled quantum system 140.

[0032] Typically, a series of operations are performed on the quantum system 140 in accordance with control data from the classical computing system 110 to implement a quantum algorithm 170. In the context of controlled quantum system 140 formed by qubits, these operations can be represented by a framework of quantum gates, which operate on single and multiple qubits to evolve the quantum system from a first quantum state to a second quantum state. The process of evolving the quantum state of the controlled quantum system 140 according to a series of operation, whether or not the controlled quantum system 140 is formed by qubits, is referred to as a quantum evolution 175.

[0033] The quantum evolution 175 of the quantum system 140 is parameterised by a control parameter θ, or set of control parameters, $\boldsymbol{\theta} = (\theta_a, \theta_b, \dots, \theta_n)$. These control parameters may take discrete values or continuous values, according to the choice of quantum algorithm 170. A cost, or loss, function is defined which, when minimised, encodes the solution to the problem to be solved. The cost function can be evaluated by measuring the expectation value of observables of the quantum system 140. The minimisation of the cost function involves the classical computation system iteratively selecting a value for one or more control parameters and initiating a quantum evolution 175 in accordance with the selected control parameters, receiving measurement data from the quantum computation system 120 and calculating a corresponding value of the cost function, and then selecting a new value for the one or more parameters that is expected to result in a reduction of the value of the cost function.

[0034] The classical computation system 100 uses an optimisation algorithm in order to determine the selection of the new parameter values. In general, the optimisation algorithm requires, as an input, the derivative of the cost function with respect to a control parameter, at the current control parameter value. The optimisation algorithm produces, as an output, a new value for the control parameter which will be used in the next iteration of the optimisation routine. In this example, the classical computation system 100 uses a stochastic gradient descent algorithm in order to inform what the new parameter values used in the parameterised quantum evolution 175 should be, but other gradient-based methods may be used in other examples.

[0035] In order to acquire the derivative of the cost-function described above, the hybrid quantum-classical computing system 100 utilises an analytical method which will hereafter be referred to as the Nyquist-shift method 150 or a variant thereof, Folding-Nyquist-shift method 150b or (p,c) Folding-Nyquist-shift method 150c, the specific details of which will be described later.

[0036] The quantum computation system 120 is comprised of quantum control hardware 122 which interfaces with a controlled quantum system 140. The controlled quantum system 140 is a physical system whose quantum properties are controllable in the sense that the system 140 can be prepared into a given quantum state, evolved under a parameterised quantum evolution 175 according to the quantum algorithm 170, and subsequently measured. The controlled quantum system is, in this example, a system of *n* qubits, but in other examples may be controlled quantum systems according to other computational schemes such as a continuous-variable scheme or a qutrit scheme. The controlled quantum system 140 includes a quantum subsystem 145 which is likewise controllable such that it can be entangled with a main subsystem 140a, the quantum subsystem 145 being an auxiliary system which may be used in conjunction with a primary portion of the controlled quantum system during the processing of the desired algorithm. Those skilled in the art will appreciate that there are a multitude of physical systems which are suitable for use as the controlled quantum system 140, such as platforms based on photonics, ion traps, quantum dots, nitrogen-vacancies, nuclear magnetic resonance, silicon spin qubits, superconducting qubits, etc. The present invention is not limited to a particular choice of platform for the controlled quantum system 140.

[0037] The quantum control hardware 122 comprises a state preparation module 124, a state evolution module 126, a quantum measurement module 128 and miscellaneous hardware 130. The state preparation module 124 is for preparing the controlled quantum system 140 into a given state. The state evolution module 126 is for evolving the controlled quantum system according to the quantum algorithm 170. The state evolution module 126 evolves the controlled quantum state 140 by applying quantum gates, with the gates parameterised by a set of control parameters $\boldsymbol{\theta}$. The quantum measurement module 128 can be used to make a measurement of the controlled quantum system 140. The quantum measurement module 128 measures a physical quantity, or observable, of the quantum system 140 by physically interacting with the quantum system 140 such that a measurement is made. The physical quantity measured by the quantum measurement module 128 is parameter-dependent as it depends on the parameterised quantum evolution 175 applied by the state evolution module 126.

[0038] The precise choice of hardware used to implement each module 126, 128, 130 depends on the choice of controlled quantum system 140. For example, photodetectors and polarising beam splitters may be used in a photonic platform for measurement of the controlled quantum system, whereas in a nuclear magnetic resonance platform, measurement is achieved by electronics which measure a voltage signal. The present disclosure is not limited to a particular choice of controlled quantum system and associated quantum control hardware 122, and those skilled in the art would know how to implement state preparation, evolution and measurement for a given choice of controlled quantum system 140. Similarly, it is clear that hardware used for e.g. state preparation may also be used for state evolution, and so the hardware used by each of the modules 124, 126, 128 may be shared between them.

[0039] Further, the quantum control hardware contains miscellaneous hardware 130. This hardware may include, for example, field-programmable gate arrays, timing electronics, digital-to-analog converters, cryostats, vacuum systems, etc, depending on what is required to effectively host the controlled quantum system 140 and interface the quantum control hardware with it.

[0040] The classical computation system 110 comprises a processor 102 and memory 113. The memory comprises data storage 113a and program routines 113b. The data storage 113a comprises storage for data necessary for, and generated during, the execution of the quantum algorithm 170 and the Nyquist-shift method 150. The memory 113 may be any suitable storage device known to those skilled in the art, such as a hard-drive, solid-state drive, cloud-storage, etc.

[0041] The processor 102 executes program routines 113b stored on the memory 113. The overarching scheduling of the instructions provided to the quantum control hardware 122 is provided by execution of the quantum algorithm routine 105 by the processor 102, though the quantum control hardware 122 may store and execute subroutines locally. The Nyquist-shift method 150 and variants 150b, 150c are implemented by the classical computation system 110 through execution of the Nyquist module routine 107 by the processor 102. Optimisation of the parameterised quantum evolution 175 by use of the gradients calculated by the method 150 is performed by the processor 102 executing the gradient descent routine 109.

[0042] Communications between the classical computation system 110 and the quantum computation system 120 can be realised by any known method, such as communicating via Ethernet, Local Area Network, Wi-Fi, TCP/IP etc.

[0043] The precise architecture of the classical computation system 110, including the processor 102, memory 113 and associated storage 105-119, and communication links between these elements may vary between examples. The classical computation system 110 may, for example, be distributed across one or more classical computers, and across one or more locations. Likewise, the quantum computational system 120 may be located at a separate location from the classical computation system 110 and accessed remotely.

[0044] Figure 2 illustrates the fundamental process of performing a parameterized quantum evolution dependent on a first control parameter θ_l.

[0045] In S202, the quantum system 140 is prepared into an initial state e.g. $|0^n\rangle$, but it will be appreciated that the exact choice of initial state will depend on the quantum algorithm to be implemented and the desired quantum computation to be performed, and the exact choice of initial state or method of preparation is therefore immaterial to the invention.

[0046] In S204, the quantum system 140 may undergo a first quantum evolution. It is dependent on the quantum algorithm to be implemented and the desired quantum computation to be performed as to whether this first quantum evolution is implemented or not. This first quantum evolution does not depend on the parameter θ_l, though it may depend on other parameters, and in some examples these other parameters may be optimised together with θ_l. This first quantum evolution may be used, for example, to prepare a problem-specific initial state beyond the state-preparation stage S202.

[0047] In S206, the controlled quantum system 140 is submitted to a time-dependent parameterised quantum evolution which is dependent on the first control parameter θ_l. Again, the precise form of the parameterised quantum evolution is dependent on the quantum algorithm to be implemented and the desired quantum computation to be performed. In some examples the control parameter θ_l is defined at the level of a physical (sub-logical level) variable, such as the intensity or frequency of a control laser pulse, but in other examples it is instead defined at the level of computational operations (logical level), such as a transformation in state-space. Precisely how the parameters enter the quantum evolutions is also decided by the user according to the precise form of the parameterised quantum evolution to be implemented and the desired quantum computation to be performed.

[0048] In S208, the quantum system may undergo a second quantum evolution. Like the first quantum evolution in S204, the second quantum evolution is optional, and does not depend on the parameter θ_l. In some examples this second quantum evolution can be used to prepare the state of a particular measurement at S210. Whilst the second quantum evolution, like the first quantum evolution S204, does not depend on the parameter θ_l, it may depend on other parameters, and in some examples these other parameters may later be trained in a similar manner to θ_l.

[0049] In S210, an observable of the quantum system is measured. Due to the fundamentally probabilistic nature of measurements made on a quantum system, in practice the entire process S202-S210 will be run multiple times and the measurements averaged to obtain an estimate of the expectation value / cost function. Minimising or reducing the number of measurements to be made to obtain an expectation value can reduce the overall cost of the algorithm 170 in the sense of, for example, required processing time. The exact choice of observable to be measured is again dependent on the quantum algorithm to be implemented and the desired quantum computation to be performed. In general, the measurement determines a loss or cost function. Finding the value of the control parameter θ_l (or, if there are several control parameters, a value for each of them) which minimises the cost function is the overarching objective of the parameterised quantum evolution paradigm.

[0050] The exemplary process described by the above steps S202-S210 is dependent on just one control parameter θ_l to illustrate the fundamental process of performing a parameterised quantum evolution, but in most examples of parameterised quantum evolutions there will be many such control parameters $\boldsymbol{\theta}$. Further, it is clear to those skilled in the art how to arrange the state preparation S202, evolution S204-208 and measurement S210 steps to implement their desired quantum algorithm.

[0051] Figure 3A illustrates the generalised process of how the control parameters $\boldsymbol{\theta}$ associated with the quantum algorithm 170 are optimised by system 100, or in other words, trained. In this example there are a set of control parameters under consideration, e.g. $\boldsymbol{\theta} = (\theta_a, \theta_b, \dots, \theta_n)$ rather than just a single control parameter as in the example provided in Figure 2. The objective of the optimisation process is to find the value of each control parameter that minimises the cost function $C(\theta)$.

[0052] In S302, the set of control parameters $\boldsymbol{\theta}$ is initialised at respective initial values. The precise choice of respective initial values may be determined in multiple ways known to the skilled person. For example, they may simply be generated at random, or may represent educated guesses based on knowledge of the problem to be solved.

[0053] In S304, one of the set of control parameters $\boldsymbol{\theta}$ is selected e.g. θ_a. In S302, this is provided with some initial value. The exact choice of which one of the set of control parameters is selected would be according to a routine based on the precise optimisation algorithm to be implemented.

[0054] In S306, a subset multiset, hereafter referred to a sub-multi-set of trial control parameter values are determined for the selected control parameter θ_a currently under consideration. That is, the sub-multi-set of trial control parameter values is a subset of all possible trial control parameter values, and a multiset in that trial control parameter values can be repeated. The sub-multi-set of trial control parameter values are determined by the Nyquist-shift method 150 or variants 150b,c. The trial control parameter values are values for θ_a at which the parameterised quantum evolution 175 will be performed and measured, such that a derivative of the cost function with respect to θ_a can be calculated.

[0055] In S308, the parameterised quantum evolution is performed for each value of the sub-multi-set of trial control parameter values. This process is substantially the same as that described in Figure 2 S202-S210. The controlled quantum system 140 is prepared, subject to a quantum evolution parameterised by the trial control parameter value and measured to obtain a measurement result which is an expectation value of the cost function. For each trial control parameter value, this process may be repeated to obtain the expectation value to a required degree of accuracy. A set of measurement results are generated, wherein each measurement result corresponds to a trial control parameter value. This determines the cost function at multiple values of the control parameter θ_a currently under consideration.

[0056] In S310, the measurement results are processed to obtain a new control parameter θ_a value. As described for S306, multiple known methods exist for determining what the new control parameter θ_a value should be. The new control parameter value is decided according to a stochastic optimisation method.

[0057] Steps S312, S314 and S316 enable the optimisation process across the full set of control parameters $\boldsymbol{\theta}$. S304-S310 are repeated to cycle through all the control parameters, for each control parameter θ_x perform parameterised quantum evolutions at determined trial control parameter values, and make corresponding measurements. At S312 steps S304-S310 are repeated until all the derivatives of the cost function with respect to each control parameter $\theta_a, \theta_b, \ldots, \theta_n$ have been calculated. All the control parameter values are then updated at step S314 according to the implemented optimising algorithm. Steps S304 to S316 are then repeated, iteratively selecting new control parameter values and trial control parameter values until the cost function $C(\boldsymbol{\theta})$ has been minimised. The person skilled in the art will recognise that there are several different ways to performing steps S312-S16 based on gradient descent methods, and the precise manner in which the parameters are optimised by the gradient descent method will vary from implementation to implementation in a way which is immaterial to the application of the method 150.

[0058] At S318, the optimisation is completed as a set of optimised parameter values which minimise the cost function $C(\boldsymbol{\theta})$ have been found. Depending on the precise choice of quantum algorithm, the solution to the problem may be encoded in the parameter values or in the measurement results, and so either may be output at this step.

[0059] Steps S302-306 and S310, S314, S318 takes place on the processor 102 of the classical computing system 110 through executing the relevant program routines stored in the memory 113b.

[0060] Step S308 takes place on the quantum computation system 120, but the precise scheduling may be guided by the quantum algorithm routine 105. In this example, the state preparation module 124 prepares the controlled quantum system 140 into an initial quantum state, state evolution module 126 submits the controlled quantum system 140 to the parameterised quantum evolution according to the trial control parameter, and the quantum measurement module 128 measures the controlled quantum system 140 to generate the measurement result.

[0061] Figure 3B illustrates the process of optimising the parameterised quantum evolution described in Figure 3A, specifically using the Nyquist-shift method 150. Steps 352 and 354 are substantially the same as steps 302 and 304. Step S356, in which the Nyquist-shift method is implemented, effectively replaces the steps S306 – 310. Steps 312 – 316 are described here by step S358, which covers any optimisation of the parameter values which occurs by known derivative-based methods e.g. stochastic gradient descent.

[0062] Having described the general paradigm of parameterised quantum evolutions and the optimisation process into which the presently disclosed Nyquist-shift method is implemented, the details of the Nyquist-shift method 150 necessary for implementation will now be presented. In this disclosure, we assume units normalised so that the Planck constant $h = 1$, which implies that $\hbar = 1/2\pi$. Any person skilled in the art can perform the necessary scaling to use the disclosure for other units, for example other values of h, in particular, for the common normalization $\hbar = 1$ (i.e., $h = 2\pi$).

[0063] The method 150 provides a way to calculate the derivative of the cost function of a parameterised quantum evolution dependent on a first control parameter, at a first value of the first control parameter and with respect to the first control parameter, as part of a hybrid quantum-classical scheme implementing a parameterised quantum evolution. It does this by determining the values for the trial control parameter values and coefficients for each of the trial control parameter values. Performing the parametrised quantum evolution on the controlled quantum system at these trial control parameter values and combining with their associated coefficients allows for the estimation of an exact derivative. This derivative can then be used by a known gradient descent optimiser to optimise the parameterised quantum evolution.

[0064] The method 150 is a parameter shift rule. One particular known parameter-shift rule states that for a parameterised quantum evolution which can be written as a unit-time evolution $e^{-i\theta H/\hbar}$, where H is the Hamiltonian, the gradient of the cost function $C(\boldsymbol{\theta})$ with respect to a parameter θ_l can be estimated exactly (without bias) by evaluating the parameterised quantum evolution at a number of values of the cost parameter θ_l. However, in reality the parameterised evolution may be perturbed by e.g. environmental or quantum control influences. This results in a unit-time evolution $e^{i(\theta A+B)/\hbar}$ with $H \approx -(\theta A + B)$, where the θA term represents the targeted quantum evolution and the B term represents a perturbation due to e.g. quantum control. This disrupts the known parameter-shift rule and introduces an unacceptable bias into the measurement of the gradient. The method 150, however is robust to such perturbations, and so can be used to obtain the exact derivative of a unit-time evolution of the form $e^{i(\theta A+B)\hbar}$, for arbitrarily large B.

[0065] Underpinning the method 150 is the fact that the exact derivative of the expectation-value function with respect to the first control parameter θ_0 at a value of the first control parameter can be calculated according to the equation:

$$f'(\theta_0) = \sum_{\substack{s=(n-\frac{1}{2})/2K \\ n\in\mathbb{Z}}} \frac{2\pi K(-1)^{2Ks+\frac{1}{2}}}{(2\pi Ks)^2} f(\theta_0 - s),$$

where s is a parameter shift value drawn from a set parameterised by K, where K relates to the Fourier spectrum of the expectation-value function. Importantly, the sequence

½ (fractions without numerators) defines a probability distribution on the set $S_K = \{(n - {}^1/_2)/2K \mid n \in \mathbb{Z}\}$. For $K = 1/2$ and $\theta_0 = 1/2$ this equality can be motivated as taking the derivative of the Nyquist-Shannon-Whittaker interpolation formula; other values of θ_0 can then be understood by translation, and other values of K by dilation.

[0066] Firstly, implementation of the method 150 is subject to conditions of the nature of the parameterised quantum evolution to which the controlled quantum system 140 is subject. The present method 150 is applicable providing, firstly, that the parameterised quantum evolution can be approximated as a unitary evolution $e^{-i(\theta A+B)/\hbar}$.

[0067] Secondly, the bounds for the Fourier spectrum of the expectation-value function must be found in order to realize the method 150. The expectation-value function is the function which maps the parameter value, θ, to the expectation value of the measurement and can be written mathematically as:

$$f(\theta) = \operatorname{tr}\left(M_0 e^{i(\theta A+B)/\hbar} \rho e^{-i(\theta A+B)/\hbar}\right),$$

where ρ denotes the density operator representing the state of the quantum system after step S204, and M denotes the measurement operator for the measurement S210 in the picture defined by the evolution S208. For example, if step S208 effects a state change that can be described by a Kraus operator $\sigma \longmapsto \sum_j E_j \sigma E_j^\dagger$ then $M = \sum_j E_j^\dagger M_0 E_j$, where M_0 represents the operator measured in step S210. Matching the adopted normalisation $h = 1$, the Fourier transform

$$\hat{f}(\xi) \coloneqq \int e^{-2\pi i \xi \theta} f(\theta) d\theta$$

is used, though the person skilled in the art can perform the necessary scaling for their preferred variant of the Fourier transform.

[0068] The way in which the bounds of the Fourier spectrum of the expectation-value function are found will vary depending on the precise implementation of the method 150. For example, they may be calculated manually *a priori* and hard-wired into the implementation, but in other examples a method which extracts them from the data about the operator A may be used. Such data may stem from a calibration process for the control of the quantum system. Here we describe some methods on how the bounds can be found.

[0069] An interval containing the Fourier spectrum can be derived from bounds on the spectrum of A. That is to say, if $\lambda_{\min}(A) \leq \lambda \leq \lambda_{\max}(A)$ holds for all elements λ in the spectrum of A, then the Fourier spectrum of the expectation-value function is contained in the interval $[-K, +K]$ where $K \coloneqq \lambda_{\max}(A) - \lambda_{\min}(A)$, a result which can be proven using the Lie-Trotter product formula. Hence, the method 150 is applicable if upper and lower bounds on the spectrum of A are known *a-priori* or can be extracted from data about the operator A.

[0070] In some examples of the disclosed method 150, additional knowledge about the operators A and B, used to approximate the evolution in S206, the initial state S202, the first quantum evolution S204, the second quantum evolution S208 and the measurement S210 can be combined to find smaller intervals containing the Fourier spectrum of the expectation-value function.

[0071] One way to achieve this is by relying on complex-valued measurements. Figure 4 displays a particular implementation of the process illustrated by Figure 2 on a system 100, the particular implementation allowing for the measurement of matrix elements $u(\theta) \coloneqq \langle\phi|e^{-(\theta A+B)/\hbar}|\psi\rangle$. This particular implementation is just one example of a method which a person skilled in the art would know in order to readily check whether the parameterised quantum evolution is suitable.

[0072] In step S402, a main quantum subsystem 140a of the controlled quantum system 140 is prepared into a definite state $|\psi_0\rangle$.

[0073] In step S404, an ancillary quantum subsystem 145 of the controlled quantum system 140 is prepared into a definite state $|+\rangle$ that is a superposition of orthogonal states $|0\rangle$ and $|1\rangle$.

[0074] In step S406, the quantum subsystem 140a is submitted to a quantum evolution effecting a unitary evolution U_1. The combined effect of S402 and S406 is to bring the quantum system into the state $|\psi\rangle$, i.e., $|\psi\rangle = U_1|\psi_0\rangle$, see above.

[0075] In step S408, the ancillary quantum subsystem 145 and the quantum subsystem 140a are subjected to a controlled parameterised quantum evolution which effects a unitary transformation $|0\rangle\langle 0| \otimes \mathrm{Id} + |1\rangle\langle 1| \otimes e^{-i(\theta A+B)\hbar}$.

[0076] In step S410, the ancillary quantum subsystem 145 and the quantum subsystem 140a are subjected to a controlled quantum evolution which effects a unitary transformation $|0\rangle\langle 0| \otimes U_2 + |1\rangle\langle 1| \otimes \mathrm{Id}$. The unitary U_2 is chosen in such a way that it would result in the state $|\phi\rangle$, if applied to $|\psi\rangle$, i.e., $|\phi\rangle = U_2|\psi\rangle$.

[0077] In step S412, a measurement is performed on the ancillary quantum subsystem 145, resulting in measuring a first Pauli observable $\sigma := X = |1\rangle\langle 0| + |0\rangle\langle 1|$ to produce a measurement result $F_{\sigma=X}$

[0078] At step S414, steps S404-S410 are repeated such that a second Pauli observable can be measured.

[0079] In step S416, the system having been re-prepared at step S414, a measurement is performed on the ancillary quantum subsystem 145, resulting in measuring a second Pauli observable $\sigma := Y = i|1\rangle\langle 0| - i|0\rangle\langle 1|$ to produce a measurement result $F_{\sigma=Y}$

[0080] At step S418, a complex number $F_{\sigma=X} + iF_{\sigma=Y}$ is returned. The measurement result that is returned in step S418 is a combination of two real-valued measurements in the steps S412 and S416, resulting, effectively, in measuring the non-Hermitian observable $M = X + iY$.

[0081] In Fig. 5, this situation is covered by the possibility to use a phase correction factor: If the Fourier spectrum of the expectation-value function is contained in an interval $[\varphi - K, \varphi + K]$, then the method 150 is applicable, by multiplication of the measurement results by $e^{-i\theta\varphi/\hbar}$. The result of the method will estimate the derivative, at the parameter value θ_0, of the function

$$\theta \longmapsto e^{-\frac{i\theta\varphi}{\hbar}} \cdot f(\theta)$$

which contains the phase correction factor $e^{-i\theta\varphi/\hbar}$ as an artefact. The artefact can be removed in the usual ways, e.g., by obtaining, through measurements, an estimate of the expectation value $f(\theta)$ at θ_0 and using the equation

$$\frac{d}{d\theta}\left(e^{-i\theta\varphi} f(\theta)\right)_{\theta:=\theta_0} = e^{-i\theta\varphi/\hbar} \cdot f'(\theta_0) - i\theta_0\varphi/\hbar \cdot e^{-\frac{i\theta\varphi}{\hbar}} \cdot f(\theta_0).$$

[0082] Returning to Figure 4, the Fourier spectrum of the complex-valued expectation-value function is contained in the interval $[\lambda_{\min}(A), \lambda_{\max}(A)]$, and hence choosing, for example $\varphi := (\lambda_{\max}(A) + \lambda_{\min}(A))/2$, and $(\lambda_{\max}(A) - \lambda_{\min}(A))/2$ will work.

[0083] That the Fourier spectrum of the complex-valued expectation-value function is contained in the interval $[\lambda_{\min}(A), \lambda_{\max}(A)]$ in the case of Figure 4 can be seen using standard mathematical manipulations, taking as a starting point that the operator-valued function $\theta \longmapsto e^{i(\theta A+B)/\hbar}$ has a Fourier spectrum that is contained in the interval $[\lambda_{\min}(A), \lambda_{\max}(A)]$; this fact can be proven using the Lie-Trotter product formula.

[0084] There is some freedom in the choice of K and it is a tunable parameter of the method 150. The width $2K$ of the interval influences the efficiency of the method: The variance of the estimator for the derivative increases when K is increased, and the magnitudes of the parameter shifts decrease when K is increased. In some examples, such as the second and third exemplary methods below, the magnitudes of the parameter shifts or parameter values may stay the same, but an approximation error may decrease with increasing K.

[0085] The method will continue to work if the Fourier spectrum extends slightly beyond the interval $[\varphi - K, \varphi + K]$ but an approximation error will be introduced depending on the magnitude of these deviations.

[0086] Figure 5 illustrates the steps required to implement the method 150.

[0087] In step S502, a real number K is determined and a phase factor φ are chosen such that the Fourier expansion spectrum of the expectation value function is contained in $[\varphi - K, \varphi + K]$, as described earlier. These values may be determined by an implementor and input to the classical computing system, or may be determined by the classical computing system by, for example, use of an algorithm or a look-up table.

[0088] In step S504, a sub-multi-set of shift values is produced by constructing a finite sub-multi-set S of the set of real numbers $\left\{\frac{(n-\frac{1}{2})}{2K} \middle| n \in \mathbb{Z}\right\}$.

[0089] In step S506, a sub-multi-set of trial control parameter values is produced by taking the current value of the selected parameter, denoted here as θ_0, and calculating trial parameter value $\theta = \theta_0 - s$ for each shift value $s \in S$.

[0090] In step S508 a sub-multi-set of weighting values q_s is produced, where the weighting values correct deviations of the frequency statistics of S from the target probability distribution $P(s) = \frac{1}{(2\pi K s)^2}$ on the set $\left\{ \frac{(n-\frac{1}{2})}{2K} \middle| n \in \mathbb{Z} \right\}$.

[0091] In step S510, the parameterised quantum evolution is performed as described in Figure 2 steps S202-S210 for each of the trial control parameter values. The measurement result F_s is stored, for example in the measurement result storage 119.

[0092] In step S512, it is determined whether a phase correction factor is required. It may not be required when $\varphi = 0$, or if $\varphi \approx 0$ in the case of some physically irrelevant error is allowed.

[0093] If no such phase correction factor is required, in S520a the derivative is obtained by calculating the sum

$$\sum 2\pi K(-1)^{2Ks+1/2} F_s q_s$$

where the measurement results F_s are weighted by their corresponding weighting value q_s and a summation factor $2\pi K(-1)^{2Ks+1/2}$.

[0094] If such a phase correction factor is required, in S514 for each shift value $s \in S$ the measurement result F_s is multiplied by phase correction factor $e^{-i(\theta_0 - s)\varphi/\hbar}$. The derivative of the cost function *scaled by the phase factor*, is then obtained in S520b by calculating the sum

$$\sum 2\pi K(-1)^{2Ks+1/2} F_s q_s.$$

[0095] Steps S504 – 508 are carried out on the processor 102 of the classical computation system 110, before communicating the trial parameter values at S510 to the quantum computation system 120. The quantum control hardware 122 then utilises the state preparation module 124, the state evolution module 126 and the quantum measurement module 128 to process the controlled quantum system 140 as described above. The measurement results resulting from S510 are communicated back from the quantum computation system 120 to the classical computation system 110. Steps S512, S514 and S520a/b are carried out on the processor 102 of the classical computation system 110.

[0096] In Step S502, the choice of the value of K allows to tune the standard deviation of the method 150 versus the magnitudes of the shifts in the θ-values produced in S504 and S506. Decreased magnitudes of shifts lead indirectly to decreased magnitudes of the θ-values. In some examples, it is desirable to keep the magnitudes of θ-values low because the magnitude of the θ-values correlate with intensities of control pulses, and higher intensity of control pulses can sometimes cause noise in the quantum computation. It is important to highlight that K does not necessarily need to be based on the largest and smallest eigenvalues, and values within the interval may be more or as appropriate depending on the system and/or algorithm the method is employed in.

[0097] In Step S504, the sub-multi-set S can be constructed randomly or deterministically. Providing an appropriate set of weighting values q_s which correct the frequency statistics of S to the targeted probability distribution $P(s) = \frac{1}{(2\pi K s)^2}$ is constructed, the skilled person will be able to think of several methods for constructing S.

[0098] As a first exemplary method of constructing S, a positive integer N is chosen which decides the variance of the estimated gradient by $O\left(1/\sqrt{N}\right)$. A real number s is sampled N times from the targeted probability distribution on the set $\left\{\frac{\left(n-\frac{1}{2}\right)}{2K} \middle| n \in \mathbb{Z}\right\}$, and added to S. This allows the weighting value $q_s = 1$ for each $s \in S$.

[0099] As a variant of the first exemplary method, pairs are sampled by sampling N times from the probability distribution $P(s) = \frac{2}{(2\pi K s)^2}$ on the set

$\left\{\frac{(n-\frac{1}{2})}{2K}\middle| n \in \mathbb{Z}, n \geq 1\right\}$ and *s* and -*s* are added to S. This allows the weighting value $q_s = 1, q_{-s} = 1$ for each $s \in S$.

[0100] As a second exemplary method of constructing S, a positive integer *N* is chosen which decides the bias (approximation error) of the estimator. The set S is instead deterministically constructed by choosing the numbers $s = (n - 1/2)/2K$ where *n* ranges from $-N + 1, \ldots, +N$. The weighting values q_s are set to $q_s = \frac{1}{(2\pi Ks)^2}$ for each $s \in S$. This results in a biased estimator with approximation error $O(1/N)$.

[0101] As a third exemplary method for constructing S, a positive integer T and a positive integer $N \gg 2T$ are chosen. S is constructed deterministically or semi-deterministically to have *N* elements, which are distributed among the real numbers

$$\left\{\frac{(n-\frac{1}{2})}{2K}\middle| n \in \mathbb{Z}, n \geq 1\right\} \cap [-(T-1/2)/2K, +(T-1/2)/2K]$$

by taking *s* N_s times, in such a way as to minimise the variance of the resulting random variable

$$\sum_{\substack{s \in \left\{\left(n-\frac{1}{2}\right)/2K \middle| n \in \mathbb{Z}, n \geq 1\right\} \cap \\ [-(T-1/2)/2K, +(T-1/2)/2K]}} \frac{(-1)^{2Ks+\frac{1}{2}}}{\pi(2Ks)^2} \mathrm{Avg}_{N_s}(F_s)$$

where $\mathrm{Avg}_{N_s}(F_s)$ denotes the average of *N* independent copies (i.e. sample repetitions) of the random variable *X*. The number of times the real number *s* is chosen, N_s, is proportional to $2K|s|$ by rounding either deterministically or randomly – for example, for choosing randomly between two numbers from $\{\lfloor x \rfloor, \lceil x \rceil\}$, the smaller number $\lfloor x \rfloor$ is chosen with probability $\lceil x \rceil - x$ and the larger number $\lceil x \rceil$ is chosen with probability $x - \lfloor x \rfloor$. The weighting values q_s for approximately correcting the frequency statistics can be set to $q_s = \frac{1}{(2\pi Ks)^2}$ for all $s \in S$. Another possibility is to correct the individual rounding for each s. In this method, the variance of the resulting estimator is $O(1/N)$, the approximation error (bias) is $O(1/T)$, and, on average, the magnitudes of the shifts that are used $2\log(T) + O_K(1)$.

[0102] It is clear to the person skilled in the art that the choice of N to determine the size of S depends on the desired level of accuracy for the gradient, as the variance of the estimated gradient is $O\left(1/\sqrt{N}\right)$ using the disclosed method. For example, early on in the optimisation process a small value of *N* may be chosen, such as $N = 10$. As a local optimum gradient is reached, the value of the gradient being small, N is increased as it becomes worthwhile to execute many repetitions to obtain a good estimate of the gradient, such as $N = 1000$. This choice may be made by the implementor and input to the classical computing device, or determined by an algorithm such as the stochastic gradient descent algorithm, for example.

[0103] It can be advantageous to determine tuneable parameters *K, N,* and where applicable *T*, in such a way as to balance the limitations of the quantum device, such as a preference for small magnitudes of shifts, with the bias and variance of the estimator depending on, for example, the intended use of the resulting derivative estimator.

[0104] Crucially, the disclosed method 150 does not require an adjustment of the parameterised quantum evolution beyond modification of the parameters.

[0105] As described regarding the choice of the value of K in S502, it can be desirable to restrict the magnitude of θ-values. This may correspond to control parameter values which are, for example, physically more straightforward to implement or less susceptible to noise and/or other sources of error. One way to address this is through using a variation of the Nyquist-shift method 150 which utilises a folding function.

[0106] A p-folding function, or parameter-folding function, for a given positive real number p, is a partial or total function $\tau : \mathbb{R} \to \mathbb{R}$ with the property that for all $s \in S_K$, we have $\tau(s) \cong s \bmod p\mathbb{Z}$. The idea of the folding function is that if γ is an approximate common divisor of the eigenvalues of A, and 1/γ divides p (e.g. $p = {}^{1}/_{\gamma}, p = {}^{2}/_{\gamma}, \ldots.$) then, for large θ-values, *p* is an approximate period of expectation values of the measurements as a function of θ. One way of using the folding function τ is, therefore, to map large θ-values to smaller ones whilst not disturbing the approximately p-periodic contributions to the derivative. This can be used to ensure that no trial control parameter value differs from the first control parameter value by more than a parameter shift limit, the parameter shift limit representing the largest acceptable magnitude of parameter shift to the user or algorithm.

[0107] Figure 6 illustrates the steps of a folding-Nyquist-shift method 150b.

[0108] In step S602, a positive real number p is decided on, which is an integer multiple of a reciprocal of the known common divisor γ of the eigenvalues of A. In other words, $p := \frac{k}{\gamma}$ where k is a positive integer.

[0109] In step S604, a p-folding function is chosen. A p-folding function is a function that satisfies $\tau(x) - x \in p\mathbb{Z} = \{pk|k \in \mathbb{Z}\}$. An example is to let τ compute the unique number in $[-\frac{p}{2}, +\frac{p}{2}[$ that is congruent to θ mod p:

$$\tau(x) := \text{ unique element of } (x + p\mathbb{Z}) \cap [-\frac{p}{2}, +\frac{p}{2}[$$

[0110] The person skilled in the art will be able to think of a variety of functions which achieve the goal of reducing the magnitudes of the θ-value function. Through selection of the folding function τ a particular balance of small magnitude θ-values and a small approximation error may be struck in accordance with the requirements of the system or algorithm in which the method is implemented.

[0111] In step S606, the steps of S502-S504 are carried out, choosing a real number K and a phase correction factor φ and producing a sub-multi-set of shift values S.

[0112] In step S608, a sub-multi-set of trial control parameter values are produced by taking the current value of the selected control parameter θ_0 and calculating $\theta = \tau(\theta_0 - s)$ for each shift value $s \in S$, implanting the folding function to ensure the values of θ remain within chosen bounds.

[0113] In step S610, steps S508-510 are carried out. This produces the sub-multi-set of weighting values and, for each trial control parameter value, performs the parameterised quantum evolution steps to produce a set of measurement values F.

[0114] Step 612 is identical to S512, and simply involves determining whether a phase correction factor is required.

[0115] In step S614, where a phase correction factor is required, in a similar manner to that described in S514 for each shift value $s \in S$ the measurement result F_s is multiplied by $e^{-2\pi i\tau(\theta_0 - s)\varphi}$, differing from S514 in that the folding function appears in the exponent.

[0116] Steps S620a and S520b are analogous to S520a and 520b.

[0117] Figure 7 depicts a specific variant of the process described by Figure 6 by using one folding function in particular, known as a *(p,c)*-folding function. This introduces a further tunable parameter *c* which a person skilled in the art may tune to adapt for a particular use. The value *c* also influences both the approximation error (like *K*) and the magnitude of the θ-values.

[0118] In step S702, a positive real number $c = m \cdot p$ is chosen, where m is a positive integer.

[0119] In step S704, the process as described by S602 is carried out.

[0120] In step S706, the specific *(p,c)*-folding function is implemented. This function is defined as

$$\tau(x) = \begin{cases} -c - \big((-x)\%p\big), \text{ if } x \leq -c - p; \\ x, \quad \text{ if } x \in]-c-p, +c+p[; \\ +c + (x\%p), \text{ if } x \geq +c + p. \end{cases}$$

where $\%p$ stands for the remainder in $[0, p[$ when dividing by p. By use of this function, *p* determines a range where the approximation error is particularly low, while *c* determines the magnitude of the approximation error.

[0121] In step S708, the remaining steps S606-620a/b are carried out.

[0122] The above embodiments are to be understood as illustrative examples of the invention. Further embodiments of the invention are envisaged. For example, the method has been described in the context of quantum computations, but it may also be applied in the context of characterisation or control of quantum systems. It is to be understood that any feature described in relation to any one embodiment may be used alone, or in combination with other features described, and may also be used in combination with one or more features of any other of the embodiments, or any combination of any other of the embodiments. Furthermore, equivalents and modifications not described above may also be employed without departing from the scope of the invention, which is defined in the accompanying claims.

ABSTRACT

A hybrid computing system configured to estimate a derivative of a parameter-dependent physical quantity that is dependent on a first control parameter θ of a parameterised quantum evolution executed by the quantum computing system. The classical computing system determines, based on an input bounding value and phase-correction value, a subset multiset of shift-values which define an equivalent subset multiset of trial control parameter values and corresponding weighting values and summation factors such that the trial control parameter values obey a targeted probability distribution. The quantum computing system executes the parameterised quantum evolution at the trial control parameter values and measures a parameter-dependent physical quantity for each trial control parameter value. On the classical computing system, the measurement results from these measurements are combined with the weighting values and summation factors to calculate at an estimate of a phase-correction-scaled derivative of the parameter-dependent physical quantity.

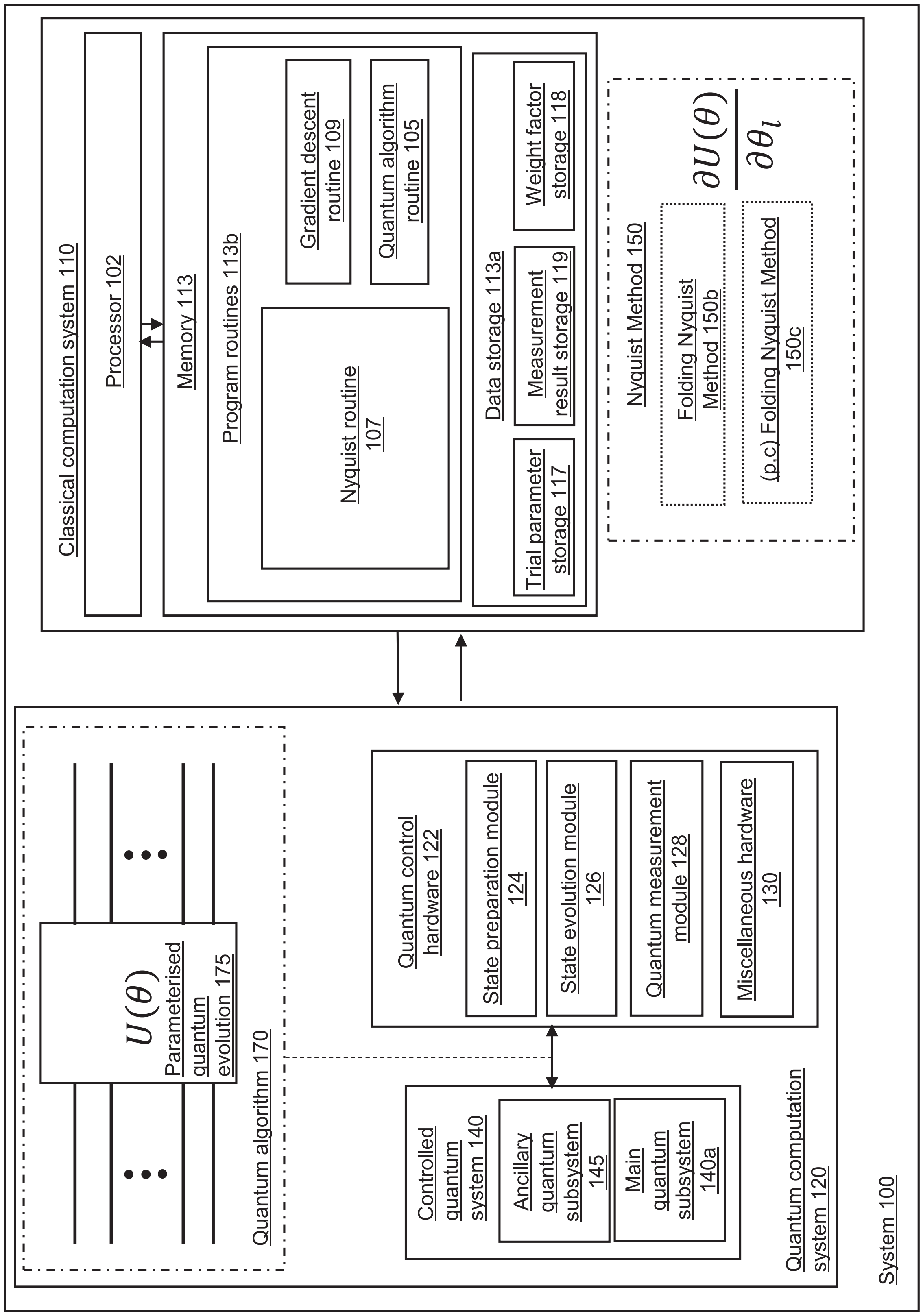

Classical computation system 110
Processor 102
Memory 113
Program routines 113b
Gradient descent routine 109
Quantum algorithm routine 105
Nyquist routine 107
Data storage 113a
Weight factor storage 118
Measurement result storage 119
Trial parameter storage 117
Nyquist Method 150
$\frac{\partial U(\theta)}{\partial \theta_l}$
Folding Nyquist Method 150b
(p.c) Folding Nyquist Method 150c
$U(\theta)$
Parameterised quantum evolution 175
Quantum algorithm 170
Quantum control hardware 122
State preparation module 124
State evolution module 126
Quantum measurement module 128
Miscellaneous hardware 130
Controlled quantum system 140
Ancillary quantum subsystem 145
Main quantum subsystem 140a
Quantum computation system 120
System 100

Figure 1

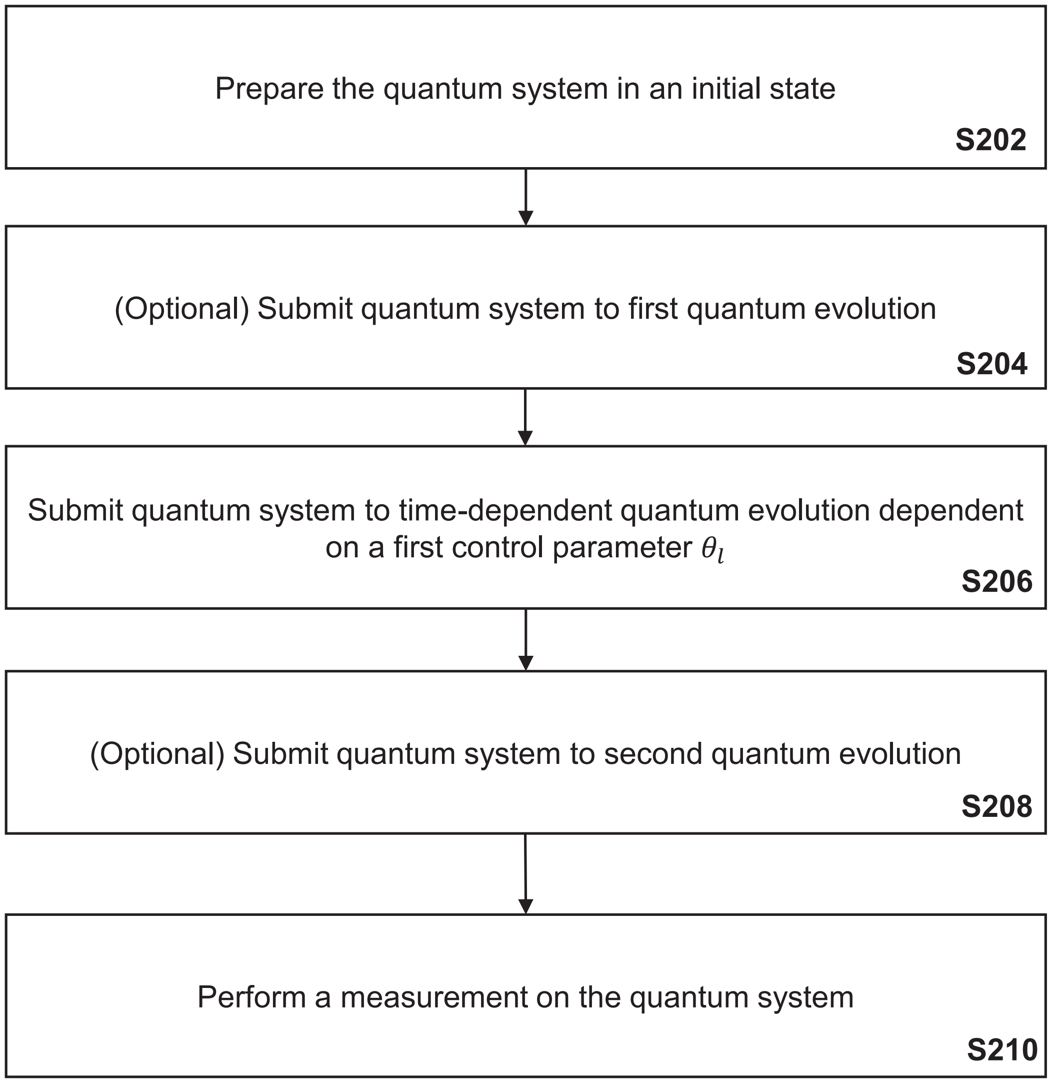
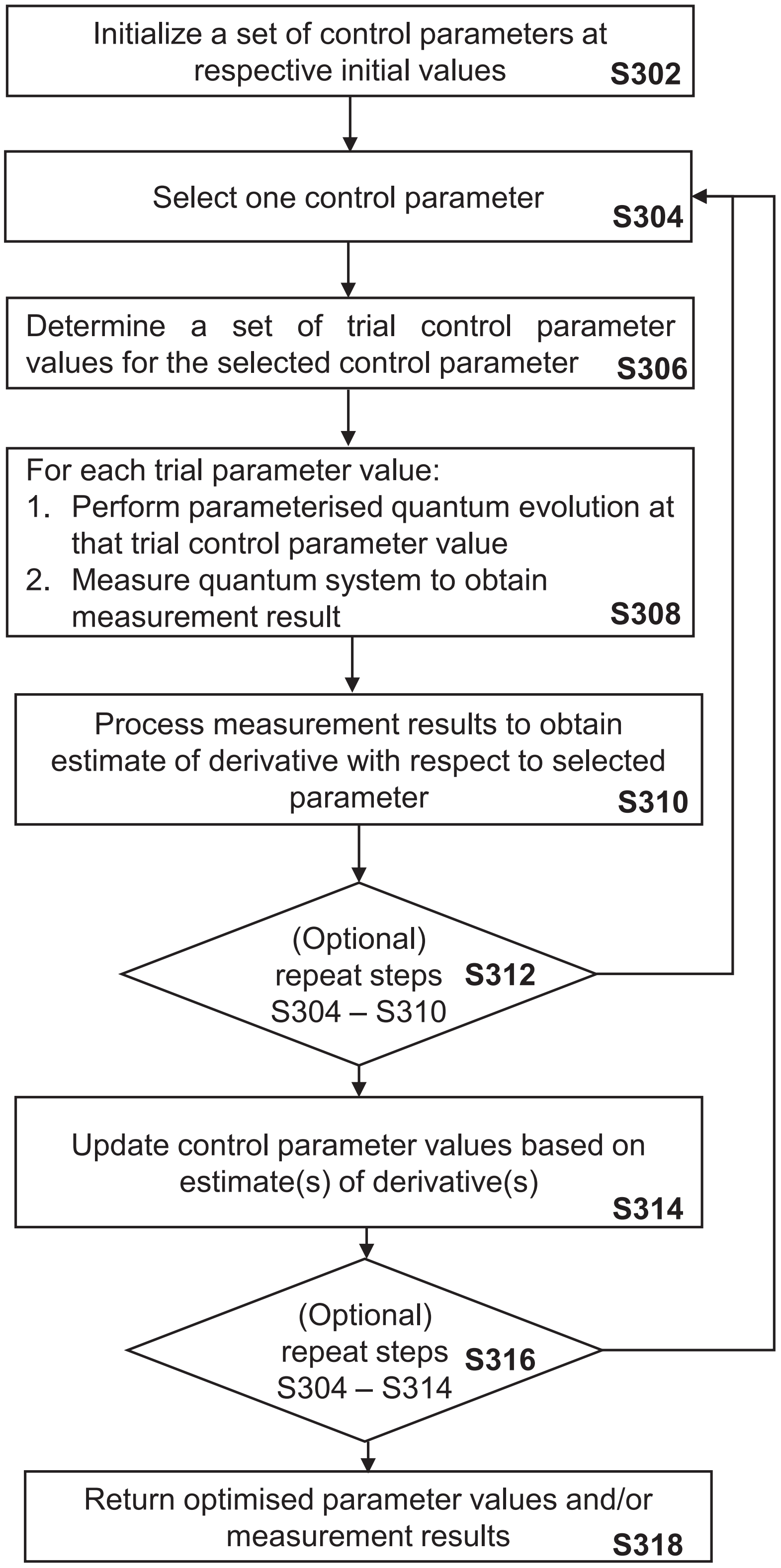

Prepare the quantum system in an initial state
S202
(Optional) Submit quantum system to first quantum evolution
S204
Submit quantum system to time-dependent quantum evolution dependent on a first control parameter θ_l
S206
(Optional) Submit quantum system to second quantum evolution
S208
Perform a measurement on the quantum system
S210

Figure 2



Initialize a set of control parameters at respective initial values
S302
Select one control parameter
S304
Determine a set of trial control parameter values for the selected control parameter
S306
For each trial parameter value:
1. Perform parameterised quantum evolution at that trial control parameter value
2. Measure quantum system to obtain measurement result
S308
Process measurement results to obtain estimate of derivative with respect to selected parameter
S310
(Optional) repeat steps S304 – S310
S312
Update control parameter values based on estimate(s) of derivative(s)
S314
(Optional) repeat steps S304 – S314
S316
Return optimised parameter values and/or measurement results
S318

Figure 3a

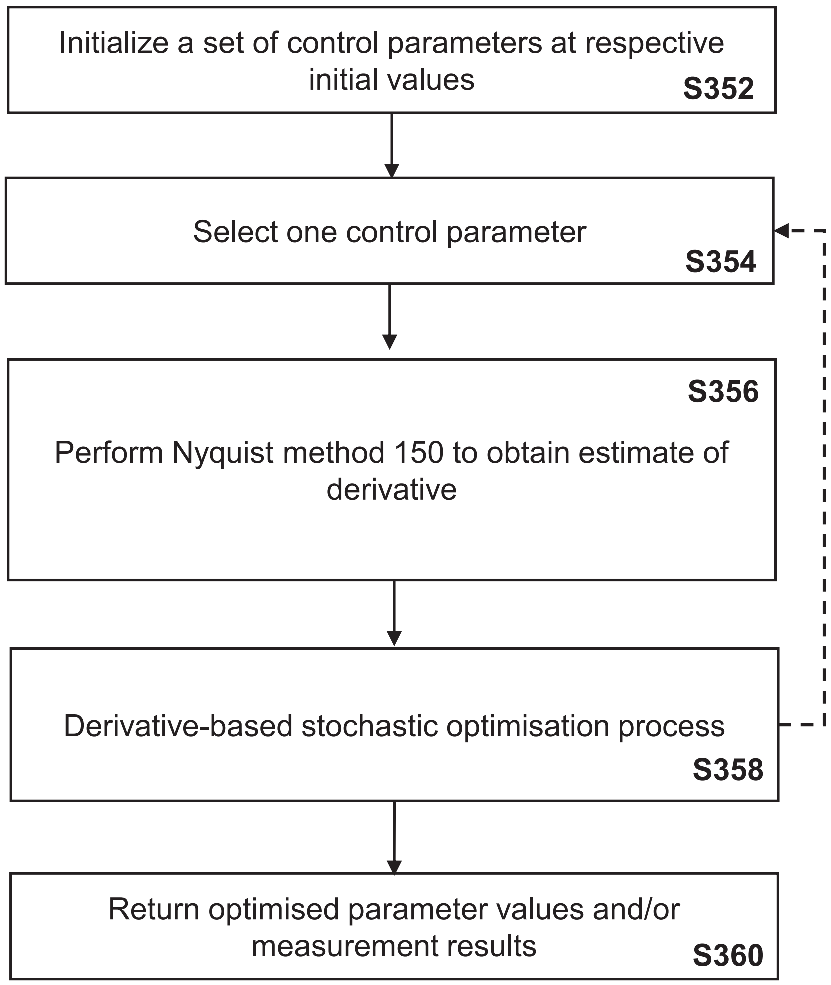
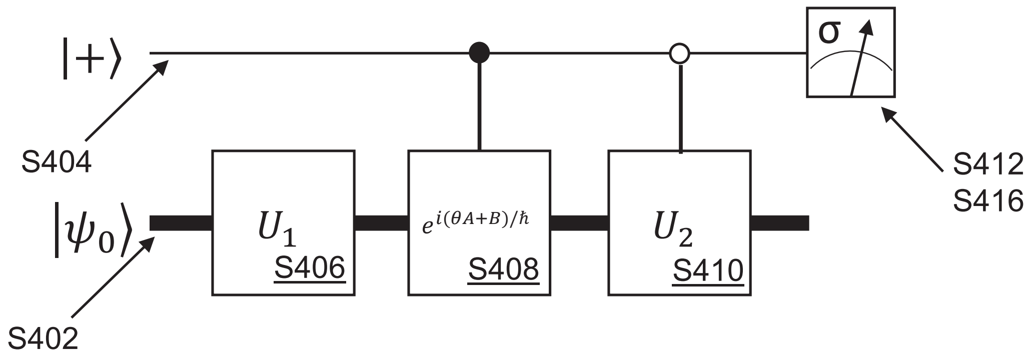

Initialize a set of control parameters at respective initial values
S352
Select one control parameter
S354
S356
Perform Nyquist method 150 to obtain estimate of derivative
Derivative-based stochastic optimisation process
S358
Return optimised parameter values and/or measurement results
S360

Figure 3b



Prepare a quantum subsystem in a definite state $|\psi_0\rangle$ **S402**

$\downarrow$

Prepare an ancillary quantum subsystem in a definite state $|+\rangle$ that is a superposition of two orthogonal states $|0\rangle$ and $|1\rangle$ **S404**

$\downarrow$

Submit the quantum subsystem to a quantum evolution effecting a unitary transformation U_1 **S406**

$\downarrow$

Submit the ancillary quantum subsystem and the quantum subsystem to a controlled quantum evolution effecting a unitary transformation $|0\rangle\langle 0| \otimes Id + |1\rangle\langle 1| \otimes e^{i(\theta A+B)/\hbar}$ **S408**

$\downarrow$

Submit the ancillary quantum subsystem and the quantum subsystem to a controlled quantum evolution effecting a unitary transformation $|0\rangle\langle 0| \otimes U_2 + |1\rangle\langle 1| \otimes Id$ **S410**

$\downarrow$

Perform a measurement on the ancillary quantum subsystem resulting in measuring a Pauli observable $\sigma \coloneqq X = |1\rangle\langle 0| + |0\rangle\langle 1|$ and denote the measurement result by $F_{\sigma=X}$ **S412**

$\downarrow$

Repeat steps S402-S410 **S414**

$\downarrow$

Perform a measurement on the ancillary quantum subsystem resulting in measuring a Pauli observable $\sigma \coloneqq Y = i|1\rangle\langle 0| - i|0\rangle\langle 1|$ and denote the measurement result by $F_{\sigma=Y}$ **S416**

$\downarrow$

Return the complex number $F_{\sigma=X} + i\, F_{\sigma=Y}$ **S418**

Figure 4

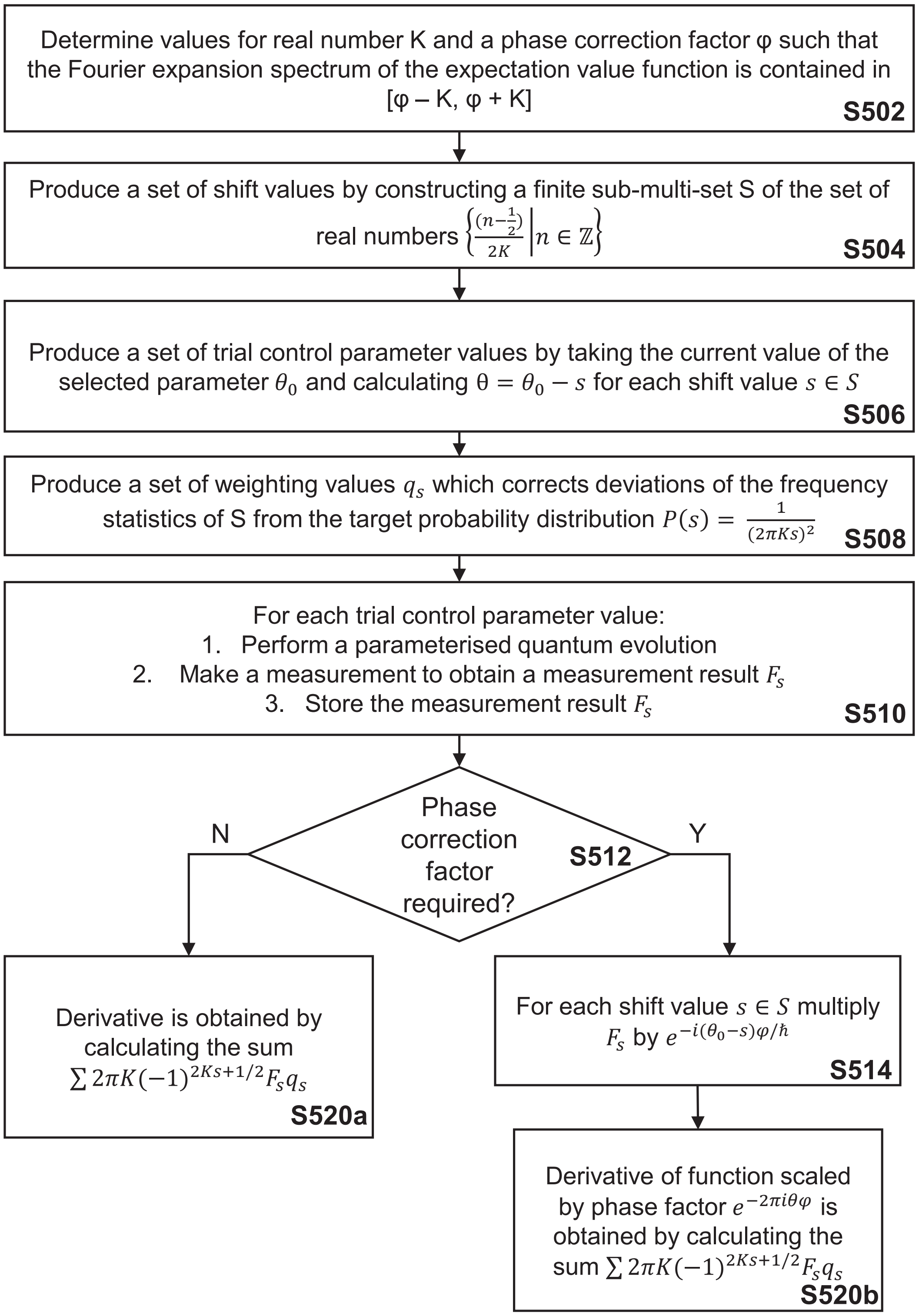
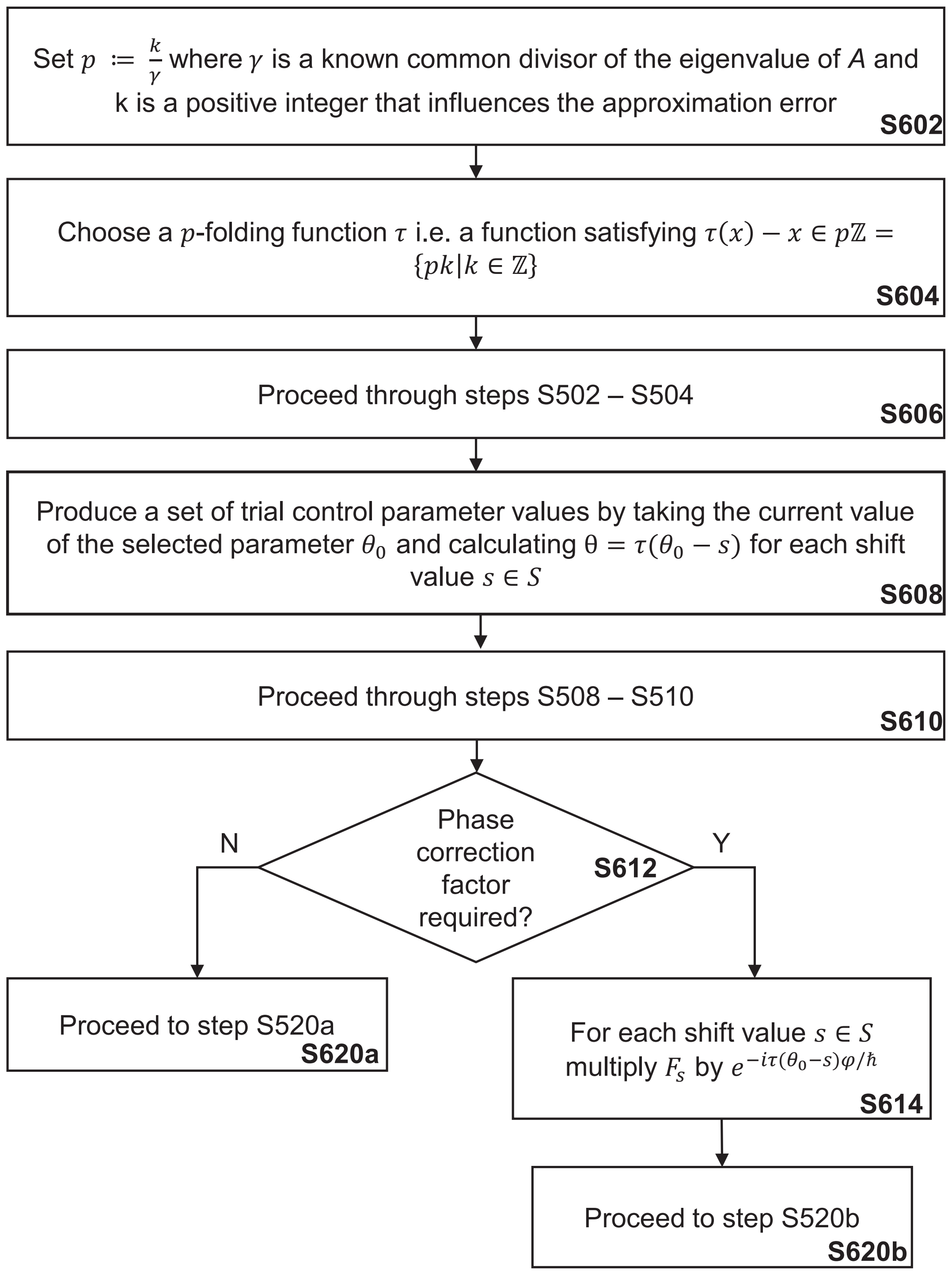

Determine values for real number K and a phase correction factor φ such that the Fourier expansion spectrum of the expectation value function is contained in [φ – K, φ + K]
S502
Produce a set of shift values by constructing a finite sub-multi-set S of the set of real numbers $\left\{\frac{(n-\frac{1}{2})}{2K}\middle| n \in \mathbb{Z}\right\}$
S504
Produce a set of trial control parameter values by taking the current value of the selected parameter θ_0 and calculating $\theta = \theta_0 - s$ for each shift value $s \in S$
S506
Produce a set of weighting values q_s which corrects deviations of the frequency statistics of S from the target probability distribution $P(s) = \frac{1}{(2\pi K s)^2}$
S508
For each trial control parameter value:
1. Perform a parameterised quantum evolution
2. Make a measurement to obtain a measurement result F_s
3. Store the measurement result F_s
S510
Phase correction factor required?
S512
N
Y
Derivative is obtained by calculating the sum $\sum 2\pi K(-1)^{2Ks+1/2} F_s q_s$
S520a
For each shift value $s \in S$ multiply F_s by $e^{-i(\theta_0-s)\varphi/\hbar}$
S514
Derivative of function scaled by phase factor $e^{-2\pi i\theta\varphi}$ is obtained by calculating the sum $\sum 2\pi K(-1)^{2Ks+1/2} F_s q_s$
S520b

Figure 5



Set $p := \frac{k}{\gamma}$ where γ is a known common divisor of the eigenvalue of A and k is a positive integer that influences the approximation error S602

Choose a p-folding function τ i.e. a function satisfying $\tau(x) - x \in p\mathbb{Z} = \{pk|k \in \mathbb{Z}\}$ S604

Proceed through steps S502 – S504 S606

Produce a set of trial control parameter values by taking the current value of the selected parameter θ_0 and calculating $\theta = \tau(\theta_0 - s)$ for each shift value $s \in S$ S608

Proceed through steps S508 – S510 S610

Phase correction factor required? S612

Proceed to step S520a S620a

For each shift value $s \in S$ multiply F_s by $e^{-i\tau(\theta_0 - s)\varphi/\hbar}$ S614

Proceed to step S520b S620b

Figure 6

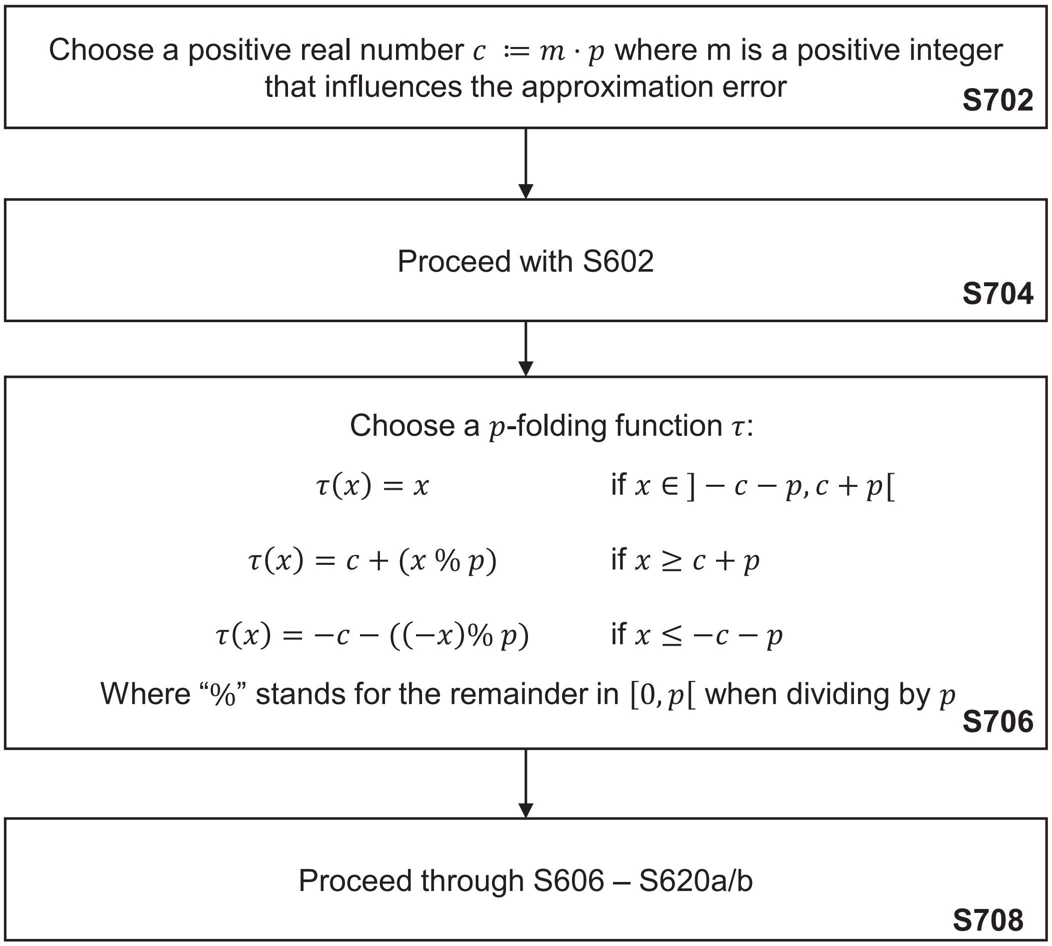

Choose a positive real number $c \coloneqq m \cdot p$ where m is a positive integer that influences the approximation error
S702
Proceed with S602
S704
Choose a p-folding function τ:
$\tau(x) = x$ if $x \in] - c - p, c + p[$
$\tau(x) = c + (x \,\%\, p)$ if $x \geq c + p$
$\tau(x) = -c - ((-x)\% \,p)$ if $x \leq -c - p$
Where "%" stands for the remainder in $[0, p[$ when dividing by p
S706
Proceed through S606 – S620a/b
S708

Figure 7