Quantum Non-linear Solvers

Hao-kai Zhang

Abstract

This note summarizes two quantum algorithms for solving nonlinear differential equations, based on quantum linear solvers by different linearization strategies. Besides, we derive a new understanding on the connection between the linearity and non-linearity in quantum many-body physics using tensor network techniques, which illustrate the linearization method used by [1] in a more compact way.

1 Introduction

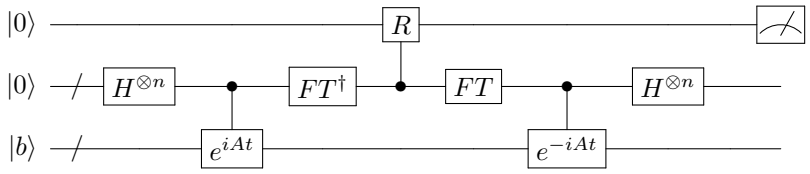
Based on the quantum phase estimation (QPE) algorithm, quantum computers have been shown to provide an exponential advantage theoretically over current classical algorithms for the solution of linear algebraic and differential equations. While many useful differential equations are non-linear, such as the Boltzmann equation describing transport phenomenon, the famed Navier-Stokes equation for fluid dynamics, the non-linear equations for plasma hydrodynamics and more. By some linearization strategies, researchers have developed algorithms [1, 2] to solve non-linear differential equations based on quantum linear solvers, which then also provide an exponential advantage over classical computers.

2 Quantum Linear Solvers

In this section, we will briefly review quantum linear solvers, including the quantum linear system algorithm and the quantum linear differential equation solver, which are subroutines of all quantum non-linear solvers discussed in the next section.

2.1 Quantum linear system algorithm

The quantum linear system algorithm (QLSA), namely the Harrow-Hassidim-Lloyd (HHL) algorithm[1], is a celebrated algorithm designed to solve linear equations of the form $A|x\rangle = |b\rangle$ using quantum computers based on the quantum phase estimation [3]. Given an oracle of controlled-e^{iAt} gate and well-prepared copies of $|b\rangle$, the HHL algorithm can be summarized as



, (1)

where FT denotes the quantum Fourier transform used in QPE and R denote a rotation gate controlled by estimated phases. If the measurement on the first qubit get $|1\rangle$, the resulting state on the last bundle of qubits is the desired $|x\rangle$. The HHL algorithm can be generalized to the non-Hermitian cases by the following trick,

$$Ax = b \to \begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix} \begin{pmatrix} 0 \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}. \tag{2}$$

The time complexity can be estimated as $O\left((\log N)s^2\kappa^2\frac{1}{\epsilon}\right)$, where N, s, κ, ϵ denote the dimensionality of x, the sparsity and condition number of A, and the tolerance error. Compared to classical algorithms such as the conjugate gradient method with time complexity $O\left(Ns\kappa\log\frac{1}{\epsilon}\right)$, it realizes exponential speed-up with respect to N.

[1] In some literature, QLSA includes not only the original HHL algorithm, but also the subsequent improved algorithms.

2.2 Quantum linear differential equation solver

General linear differential equations could be mapped to algebraic linear equations in the higher dimensions based on various discretization strategies such as the Euler methods and the Runge-Kutta methods, etc. The same is true for quantum cases.

The quantum linear differential equation solver [4] is nothing but the quantum linear system algorithm for discretized linear differential equations, with a trick to encode a time series of states. Take the Euler forward method for an example. Consider a linear differential equation of the form,

$$\frac{dx}{dt} + Ax = b(t), \tag{3}$$

where $x, b \in \mathbb{C}^d$ denote vectors in a d-dimensional complex linear space and A is a $d \times d$ constant matrix. Discretize the time variable at intervals Δt, and take $k = 0, 1, ..., T$ to be the index for the discretized time, so that x_k and b_k are the values of $x(t)$ and $b(t)$ at time label k. Given the initial condition $x_0 = b_0$, the Euler forward method discretizes Eq (3) to a series of recurrence equations,

$$x_0 = b_0,\ x_1 = x_0 - \Delta t A x_0 + \Delta t b_1,\ ...,\ x_{k+1} = x_k - \Delta t A x_k + \Delta t b_k, ... \ . \tag{4}$$

Written in matrix form, these equations become

$$-\begin{pmatrix} -I & 0 & 0 & \cdots & 0 & 0 \\ I - \Delta t A & -I & 0 & \cdots & 0 & 0 \\ 0 & I - \Delta t A & -I & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -I & 0 \\ 0 & 0 & 0 & \cdots & I - \Delta t A & -I \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{T-1} \\ x_T \end{pmatrix} = \begin{pmatrix} b_0 \\ \Delta t b_1 \\ \Delta t b_2 \\ \vdots \\ \Delta t b_{T-1} \\ \Delta t b_T \end{pmatrix}. \tag{5}$$

To implement the direct sum in the column vectors of Eq (5), we adjoin a time-step register $|k\rangle$ and encode the solution vector and the inhomogeneity as "quantum history states",

$$|X\rangle = \sum_k |x_k\rangle|k\rangle,\ |B\rangle = |b_0\rangle|k=0\rangle + \Delta t \sum_{k=1}^{T} |b_k\rangle|k\rangle. \tag{6}$$

Correspondingly, the matrix in Eq (5) is encoded as

$$\mathcal{M} = \sum_{k=0}^{T} I \otimes |k\rangle\langle k| - \sum_{k=0}^{T-1} (I - \Delta t A) \otimes |k+1\rangle\langle k|. \tag{7}$$

Thus, the original discretized equation Eq (5) is encoded as $\mathcal{M}|X\rangle = |B\rangle$, which is accessible to be solved using the HHL algorithm and the resulting solution $|X\rangle$ can be measured and undergo post-processing [4] to reveal its features. In practice, in order to produce $x_{k=T}$ with relatively high probability by post-selection, one could extend the integration time for identity evolution to improve the weight of x_T.

3 Quantum Non-linear Solvers

The non-linearity usually manifests itself as polynomials of order higher than one, i.e.

$$g(x) = Ax + b \ \rightarrow \ g(x) = A_n x^{\otimes n} + ... + A_2 x^{\otimes 2} + A_1 x + b. \tag{8}$$

Whereas in differential equation, equivalently, the non-linearity is more often represented by a coefficient matrix depending on state vector, i.e.

$$g(x) = Ax + b \ \rightarrow \ g(x) = A(x)x + b. \tag{9}$$

Here we focus on non-linear differential equations of the form,

$$\frac{dx}{dt} + f(x)x = b(t), \tag{10}$$

where $x \in \mathbb{C}^d$ denotes a state vector in a d-dimensional complex linear space and $f(x)$ is a $d \times d$ matrix that is an order m polynomial function of the vectors x and $x^\dagger$. Due to the linear nature of quantum circuits, it is necessary to perform linearization as a preprocessing if one aims to solve non-linear differential equations on quantum computers. We will introduce two linearization methods [1, 2] in a comprehensible way in this section. Both of the two methods transform low-dimensional non-linear equations to high-dimensional linear equations. The difference lies only in the transformation details.

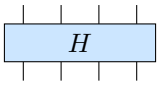
3.1 Mean-field linearization

In the context of quantum many-body physics, there is a natural bridge between linearity and non-linearity. A famous example is that the non-linear Schrödinger equation for a single system can arise as an approximation of the usual linear Schrödinger equation for multiple identical interacting copies of the original system, in the limit of large number of copies. However, there is nothing physically mysterious about this linearization method. Namely, it can be totally understood in a pure mathematical manner.

In order to avoid ambiguity, we will adopt the Penrose notation as in tensor networks. Firstly, as is known to us, an operator H in quantum mechanics is just a linear map between state vectors, which can be simply represented by a matrix under a certain basis, i.e.

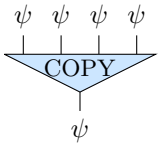
(11)

Specially, if the Hilbert space can be decomposed to a tensor product of n identical spaces $\mathcal{H}_{\text{tot}} = \bigotimes_j \mathcal{H}_j$, we would reshape the matrix to a tensor by convention, with each index corresponding to a "local" Hilbert space $\mathcal{H}_j$. Take $n = 4$ for an example.



(12)

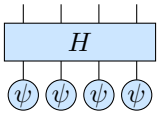
The reshaping does not change the linear nature of the map $\Psi \to H\Psi$. Things become different if there are dependencies among the components of a state vector. In the following, we will focus on a simple instance of identical product state $\Psi = \psi^{\otimes n}$. Denote this "copy" operation as $\text{COPY}_n(\psi) = \psi^{\otimes n}$ and give it a graph notation.[2]



(13)

It is important to note that the copy operation here is a non-linear map[3], which is the origin of the non-linearity in the non-linear Schrödinger equation. This property relates to the no-cloning theorem in quantum information theory, which can be simply verified by checking the definition of linear maps.

Now, we want to approximate the usual linear Schrödinger equation $i\partial_t \Psi = H\Psi$ within the restricted ansatz manifold of identical product states $\{\psi^{\otimes n}\}$.[4] However, while the result of $i\partial_t \psi^{\otimes n}$ is in the tangent space of our ansatz by definition, the result of $H\psi^{\otimes n}$,



(14)

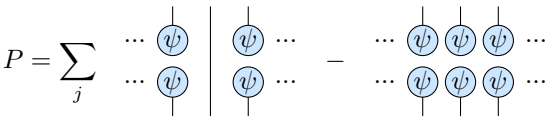
can not exactly reduce to the tangent space since the evolved state is generally an entangled state in the presence of interactions. We have to project $H\psi^{\otimes n}$ back to the tangent space of our ansatz manifold as an approximation. In fact, it can be seen as the zero entanglement limit of the time dependent variational principle (TDVP) algorithm within the framework of tensor network, from which we obtain

[2] There is a little notation abuse since conventionally we only use the Penrose notation for linear maps.

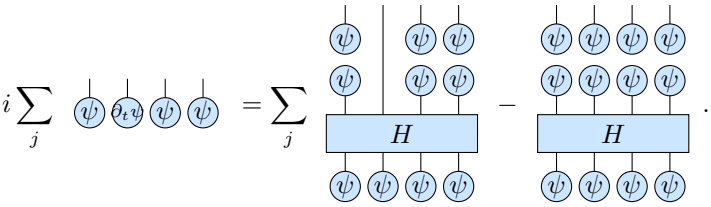
[3] The power of tensor product is reduced to the power of scalar product ψ^n if the local vector space becomes trivial, i.e. it is 1-dimensional so that the resulting global space is also 1-dimensional. In this case, the copy operation is reduced to a scalar power function where the non-linearity becomes clear.

[4] For simplicity, we suppose that the Hamiltonian H is symmetric over all local degrees of freedom.

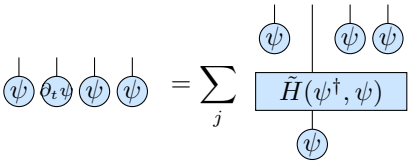
the general form of the projector P. In the special case of product state (1-dimensional virtual bond), P can be represented as

$$P = \sum_j$$  (15)

where we have implicitly taken complex conjugate for vectors with downward index wires and $\sum_j$ denotes the summation over all local degrees of freedom. Therefore, the projected Schrödinger equation $i\partial_t \Psi = PH\Psi$ is written as

$$i\sum_j$$  (16)

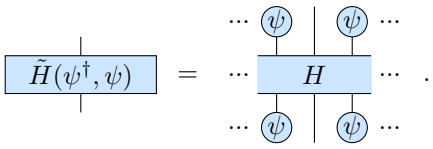
This equation can be simplified as

$$i\sum_j$$  (17)

where $\langle H \rangle$ denote the energy expectation with respect to $\psi^{\otimes n}$. Considering the identical symmetry over local degrees of freedom, we finally obtain the general form of the non-linear Schrödinger equation,

$$i\partial_t \psi = \left(\tilde{H}(\psi^\dagger, \psi) - \langle H \rangle\right) \psi, \tag{18}$$

where the "non-linear" Hamiltonian reads

 (19)

This result is closely related to the "mean-field" technique in quantum many-body physics, averaging the interacting effect simply by one-body expectation.[5] Take two-body interaction for an example.

$$\begin{aligned} \sum_j A_j A_{j+1} &= \sum_j (A_j - \langle A_j \rangle + \langle A_j \rangle)(A_{j+1} - \langle A_{j+1} \rangle + \langle A_{j+1} \rangle) \\ &\approx \sum_j A_j \langle A_{j+1} \rangle + \langle A_j \rangle A_{j+1} - \langle A_j \rangle \langle A_{j+1} \rangle. \end{aligned} \tag{20}$$

It is consistent with the form of Eq (18). Up to this point, we have demonstrated how to generally approximate a high-dimensional linear differential equation by a low-dimensional non-linear one using the mean-field technique. It is not a surprise that this procedure can be done reversely, which is named as "mean-field linearization" here [4].

We start from the general form of non-linear equations in Eq (10). The $d \times d$ matrix $f(x)$ is generally an inhomogeneous polynomial of order m with respect to the vector x, while the mean-field technique in Eq (19) only works for homogeneous cases.[6] This contradiction is easily reconciled by increasing the dimensionality of x by 1 with an additional constant component, i.e. $x \to \tilde{x} = (1, x)$.[7]

[5] In principle, tracing out environment generally leads to mixed states. But people still use pure states approximately.
[6] The original literature neglects the energy expectation term $\langle H \rangle$.
[7] For simplicity of notation, we still use x instead of $\tilde{x}$ below.

As thus, we can construct a linear "Hamiltonian" F satisfying $f(x) = x^{\dagger\otimes m} F x^{\otimes m}$ by use of Eq (19).[8] By solving the corresponding linear differential equation, one can approximately obtain the solution of the original non-linear one. The explicit construction as in Eq (5) is

$$\begin{aligned} |B\rangle^{(n)} &= |b_0\rangle^{\otimes n}|k=0\rangle + \Delta t \sum_{k=1}^{T} |b_k\rangle^{\otimes n}|k\rangle, \\ \mathcal{M}^{(n)} &= \sum_{k=0}^{T} I \otimes |k\rangle\langle k| - \sum_{k=0}^{T-1} (I - \Delta t F) \otimes |k+1\rangle\langle k|. \end{aligned} \tag{21}$$

The solution to the equation $\mathcal{M}^{(n)}|X\rangle^{(n)} = |B\rangle^{(n)}$ approximately equals to

$$|X\rangle^{(n)} \approx \sum_k |x_k\rangle^{\otimes n}|k\rangle, \tag{22}$$

where $|x_k\rangle$ is the Euler forward solution to Eq (10) at time step k, i.e.

$$|x_k\rangle = (I - \Delta t f(x_{k-1}))|x_{k-1}\rangle + \Delta t |b_k\rangle. \tag{23}$$

The approximation error is estimated as $O(|E|^2 t \Delta t m^2/n)$, where $|E|^2$ is the average modulus squared for the eigenvalues of f(x) over the integration time t and n denotes the number of copies.

3.2 Carleman linearization

As a traditional method from pure mathematical considerations, the Carleman linearization can also be used to design a quantum algorithm to solve non-linear differential equations [2]. The main idea is to introduce powers of the variables as additional components of new variables. For instance, consider a scalar quadratic ordinary differential equation (ODE),

$$\frac{du}{dt} = au^2 + bu + c. \tag{24}$$

The power series of u satisfy

$$\begin{aligned} \frac{d}{dt}u^2 &= 2u(au^2 + bu + c), \\ \frac{d}{dt}u^3 &= 3u^2(au^2 + bu + c), \ \dots . \end{aligned} \tag{25}$$

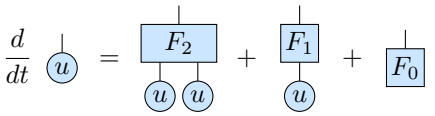
Thus, we can transform a non-linear differential equation into an infinite-dimensional linear one with respect to the vector $(u, u^2, u^3, ...)$, i.e.

$$\frac{d}{dt}\begin{pmatrix} u \\ u^2 \\ u^3 \\ \vdots \end{pmatrix} = \begin{pmatrix} b & a & 0 & 0 & \cdots \\ 2c & 2b & 2a & 0 & \cdots \\ 0 & 3c & 3b & 3a & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} u \\ u^2 \\ u^3 \\ \vdots \end{pmatrix} + \begin{pmatrix} c \\ 0 \\ 0 \\ \vdots \end{pmatrix}. \tag{26}$$

Truncation on the dimensionality leads to an approximation as an operational finite-dimensional linear ODE. Similarly, if u is not a scalar but a vector $u = (u_1, ..., u_n)^T \in \mathbb{R}^n$ satisfying a quadratic ODE,

$$\frac{du}{dt} = F_2 u^{\otimes 2} + F_1 u + F_0(t), \tag{27}$$

graphically,



(28)

[8] This is a notation full of ambiguity at the first sight. But it is actually the same with the right hand side in Eq (19).

where $u^{\otimes 2} = (u_1^2, u_1u_2, ...u_1u_n, u_2u_1, ..., u_nu_{n-1}, u_n^2)^T \in \mathbb{R}^{n^2}$, and $F_2 \in \mathbb{R}^{n^2\times n}, F_1 \in \mathbb{R}^{n\times n}$ are time-independent matrices, the truncated Carleman linearization leads to a tri-diagonal block structure,

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\hat{y}_1\\ \hat{y}_2\\ \hat{y}_3\\ \vdots\\ \hat{y}_{N-1}\\ \hat{y}_N\end{pmatrix} = \begin{pmatrix} A_1^1 & A_2^1 & 0 & \cdots & 0 & 0\\ A_1^2 & A_2^2 & A_3^2 & \cdots & 0 & 0\\ 0 & A_2^3 & A_3^3 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & A_{N-1}^{N-1} & A_N^{N-1}\\ 0 & 0 & 0 & \cdots & A_{N-1}^N & A_N^N\end{pmatrix}\begin{pmatrix}\hat{y}_1\\ \hat{y}_2\\ \hat{y}_3\\ \vdots\\ \hat{y}_{N-1}\\ \hat{y}_N\end{pmatrix} + \begin{pmatrix}F_0(t)\\ 0\\ 0\\ \vdots\\ 0\\ 0\end{pmatrix}, \tag{29}$$

where $\hat{y}_j = u^{\otimes j} \in \mathbb{R}^{n^j}$ and $A_{j+1}^j \in \mathbb{R}^{n^j\times n^{j+1}}, A_j^j \in \mathbb{R}^{n^j\times n^j}, A_{j-1}^j \in \mathbb{R}^{n^j\times n^{j-1}}$. Since

$$d\left(u\otimes u\otimes ...\otimes u\right) = \left(du\otimes u\otimes ...\otimes u\right) + \left(u\otimes du\otimes ...\otimes u\right) + ..., \tag{30}$$

the matrices $A_{j+1}^j, A_j^j, A_{j-1}^j$ satisfy

$$\begin{aligned} A_{j+1}^j &= F_2\otimes I^{\otimes j-1} + I\otimes F_2\otimes I^{\otimes j-2} + \cdots + I^{\otimes j-1}\otimes F_2,\\ A_j^j &= F_1\otimes I^{\otimes j-1} + I\otimes F_1\otimes I^{\otimes j-2} + \cdots + I^{\otimes j-1}\otimes F_1,\\ A_{j-1}^j &= F_0\otimes I^{\otimes j-1} + I\otimes F_0\otimes I^{\otimes j-2} + \cdots + I^{\otimes j-1}\otimes F_0. \end{aligned} \tag{31}$$

The total dimensionality of Eq (29) is $\Delta = \sum_{j=1}^N n^j = O(n^N)$. Thus, we can apply the quantum linear differential equation solver for the linearized equation Eq (29) and produce $\hat{y}_1 = u$ by post-selection. The approximation error is estimated as $O(te^{-N})$, where the ratio of non-linearity R defined in [2] is assumed to satisfy $R < 1$ so that the error decays exponentially with the truncated dimension N.

4 Summary

To sum up, both of the quantum non-linear solvers use a linear system over multiple copies to induce the single-system non-linearity and take the quantum linear differential equation solver as a subroutine. The mean-field linearization [1] has a quantum physical background and is more natural and concise to implement on practical devices, but its approximation error scales with the number of copies inverse-linearly. While the Carleman linearization [2] is shown to have exponential small error scaling with the number of copies under certain conditions when applied to quadratic non-linear differential equations, but demand an additional post-selection on the different number of copies.

References

[1] Seth Lloyd, Giacomo De Palma, Can Gokler, Bobak Kiani, Zi-Wen Liu, Milad Marvian, Felix Tennie, and Tim Palmer. Quantum algorithm for nonlinear differential equations. pages 1–17, nov 2020.

[2] Jin-Peng Liu, Herman Øie Kolden, Hari K. Krovi, Nuno F. Loureiro, Konstantina Trivisa, and Andrew M. Childs. Efficient quantum algorithm for dissipative nonlinear differential equations. *Proc. Natl. Acad. Sci.*, 118(35):e2026805118, aug 2021.

[3] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum Algorithm for Linear Systems of Equations. *Phys. Rev. Lett.*, 103(15):150502, oct 2009.

[4] Bobak T. Kiani, Giacomo De Palma, Dirk Englund, William Kaminsky, Milad Marvian, and Seth Lloyd. Quantum advantage for differential equation analysis. oct 2020.