East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.2022-315.200423

Vol. **13**, No. 3, pp. 717-739
August 2023

Time Complexity Analysis of Quantum Difference Methods for Multiscale Transport Equations

Xiaoyang He[1], Shi Jin[1,2,*] and Yue Yu[1]

[1]*School of Mathematical Sciences, Institute of Natural Sciences, MOE-LSC, Shanghai Jiao Tong University, Shanghai 200240, China.*
[2]*Shanghai Artificial Intelligence Laboratory, Shanghai, China.*

Received 12 November 2022; Accepted (in revised version) 20 April 2023.

Dedicated to Professor Tao Tang on the occasion of his 60th birthday.

Abstract. We investigate time complexities of finite difference methods for solving the multiscale transport equation by quantum algorithms. It is found that the time complexity of classical and quantum treatments of the standard explicit scheme scale is $\mathcal{O}(1/\epsilon)$, where ϵ is a small scaling parameter. On the other hand, the complexity of the even-odd parity based asymptotic-preserving (AP) schemes do not depend on ϵ. This indicates that in quantum computing, AP schemes (and probably other multiscale ones) are of great importance for solving multiscale transport and kinetic equations.

AMS subject classifications: 81Q15

Key words: Quantum difference methods, multiscale transport equations, asymptotic-preserving schemes, time complexity.

1. Introduction

Transport equations arise in many important applications, from medical imaging, astrophysics, nuclear reactor, to wave propagation in random media and semiconductor device modeling [4,5,25,26]. These equations model probability distribution of particles in a background medium, thus are defined in phase space, suffering from curse-of-dimensionality. In addition, the problem may encounter multiple temporal and spatial scales, and the numerical resolution of the small scales will further increase the computational cost tremendously. Despite of rapid development of multiscale methods, high dimensionality and multiple scales could still pose a major challenge for numerical simulations for transport, and more generally, kinetic equations by classical computers.

On the other hand, quantum computers, in various instances, have been shown to exhibit potential polynomial and even exponential advantage over the classical computers,

*Corresponding author. *Email addresses:* hexiaoyang@sjtu.edu.cn (X. He), shijin-m@sjtu.edu.cn (S. Jin), terenceyuyue@sjtu.edu.cn (Y. Yu)

if one designs adequate quantum algorithms. One of such possibilities is linear algebra problems [6,10,12]. After numerical discretizations, ordinary and partial differential equations can also be formulated as linear algebra problems thus can also use quantum linear algebra solvers to gain quantum advantages in dimension, precision, and the evolution time [2,7,9,14,16,19,22,23]. Most of these works aim at producing quantum state, after which a measurement is needed to extract classical data. In [14] though, physical observables are obtained with possible quantum advantage.

In particular, in [16], for a linear hyperbolic relaxation system with possibly stiff relaxation, it shows that a good multiscale scheme – in this case the popular in kinetic community asymptotic-preserving (AP) scheme, has shown its advantage for quantum algorithms over standard non-AP schemes. Specifically, the numerical complexity that depends on the reciprocal of the small physically scaling scales is great relaxed: the complexity of AP quantum algorithms is independent of the small scaling parameter.

In this article we study the multiscale linear transport equation

$$\epsilon \partial_t f + v \partial_x f = \frac{1}{\epsilon}\left(\frac{1}{2}\int_{-1}^{1} f\, dv' - f\right), \quad x_L < x < x_R, \quad -1 \le v \le 1, \tag{1.1}$$

where $f = f(t,x,v)$ is the probability density distribution for particles at space point $x \in \mathbb{R}$, time t, and $v \in (-1,1)$ is the cosine of the angle between the particle velocity and the x-axis. Comparing with the work in [16], here the equation is in the phase space, and one needs to also discretize the velocity (or angle) variable, and to deal with the nonlocal collision operator, hence further complicating the development of numerical approximations and the study of their time complexity for quantum algorithms. Our goal is to compare the time complexity of quantum algorithms based on an AP scheme [17] and a standard (explicit, thus not AP) scheme and show that the former has a complexity independent of the small physical scaling parameter ϵ while the latter depends on it. Hence, it demonstrates that multiscale methods still make a big difference in terms of time complexity even for quantum algorithms.

Since our aim is to compare the difference in dependence of ϵ, in this article we will only study the spatially one dimensional equation. Quantum advantages in spatial dimensions for numerical methods of partial differential equations have been well studied in other literature, see for examples [7,15,16,21].

Compared with the earlier work [16], where a multiscale hyperbolic relaxation system was studied, here in the time complexity analysis for transport equation defined in the phase space with a nonlocal collisional term, the analytic difficulty is to give a lower bound of the minimum singular value of the coefficient matrix. When neglecting the nonlocal term, one easily observes that the problem is reduced to the prototype problem for fixed velocity variable. Its simplicity allows one to estimate the singular values of the coefficient matrix directly, in which the proof ultimately boils down to the upper bound of the 2-norm

of the inverse matrix of $\boldsymbol{K}$ in the form of

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{I} & & & \\ -\boldsymbol{B}_1 & \boldsymbol{I} & & \\ & \ddots & \ddots & \\ & & -\boldsymbol{B}_1 & \boldsymbol{I} \end{bmatrix},$$

which satisfies $\|\boldsymbol{B}_1\| \leq 1$ under an appropriate CFL condition and leads to the expected estimate

$$\|\boldsymbol{K}^{-1}\| \leq 1 + \|\boldsymbol{B}_1\| + \|\boldsymbol{B}_1\|^2 + \cdots + \|\boldsymbol{B}_1\|^{N_t - 1} \leq N_t.$$

In contrast, the inclusion of the integral term due to the nonlocal collision operator makes the discretization fully coupled in the angular direction, so the direct manipulation of the coefficient matrix will be rather involved as opposed to the analysis for the prototype problem. For this reason, we instead characterize the singularity by using the Fourier analysis approach on the spatial variable, which enables us to derive the CFL condition quite naturally, and makes the system more convenient to perform the perturbation technique. Our analysis also relies on the special properties of a rank-one matrix composed of the weights of the numerical integration as introduced in the proof of Theorem 3.1.

2. Even-Odd Parity Based AP Scheme for Multiscale Transport Equation

In this section we will review an AP scheme for (1.1), viz the diffusive relaxation scheme, proposed in [17].

2.1. A diffusive relaxation system

The transport equation can be reformulated to the diffusive relaxation system. To this end, let us split it into two equations, each for $v > 0$

$$\epsilon \partial_t f(v) + v \partial_x f(v) = \frac{1}{\epsilon}\left(\frac{1}{2}\int_{-1}^{1} f\, dv - f(v)\right),$$
$$\epsilon \partial_t f(-v) - v \partial_x f(-v) = \frac{1}{\epsilon}\left(\frac{1}{2}\int_{-1}^{1} f\, dv - f(-v)\right).$$

Introducing the even- and odd-parities

$$r(t,v,x) = \frac{1}{2}\big[f(t,v,x) + f(t,-v,x)\big],$$
$$j(t,v,x) = \frac{1}{2\epsilon}\big[f(t,v,x) - f(t,-v,x)\big],$$

one has the following system:

$$\begin{aligned} &\partial_t r + v \partial_x j = \frac{1}{\epsilon^2}(\rho - r), \\ &\partial_t j + \frac{v}{\epsilon^2}\partial_x r = -\frac{1}{\epsilon^2} j, \end{aligned} \tag{2.1}$$

where

$$\rho(t,x)=\int_0^1 r\,dv.$$

The idea of [18] is to rewrite (2.1) as the following diffusive relaxation system:

$$\begin{aligned}&\partial_t r+v\partial_x j=-\frac{1}{\epsilon^2}(r-\rho),\\&\partial_t j+\phi v\partial_x r=-\frac{1}{\epsilon^2}\left(j+(1-\epsilon^2\phi)v\partial_x r\right),\end{aligned}\tag{2.2}$$

where $\phi=\phi(\epsilon)$ satisfies $0\le\phi\le 1/\epsilon^2$. The requirement of ϕ guarantees that $\phi(\epsilon)$ and $1-\epsilon^2\phi(\epsilon)$ are positive, making the problem uniformly stable when ϵ is small. A simple choice is $\phi(\epsilon)=\min\{1,1/\epsilon\}$. In what follows, we take $\phi=1$ since we are mainly concerned with the case of $\epsilon\ll 1$.

2.2. A diffusive relaxation scheme

Ref. [18] presented a natural splitting of (2.2) which consists of combining the relaxation step

$$\begin{aligned}&\partial_t r=-\frac{1}{\epsilon^2}(r-\rho),\\&\partial_t j=-\frac{1}{\epsilon^2}\left(j+(1-\epsilon^2)v\partial_x r\right)\end{aligned}\tag{2.3}$$

with the transport step

$$\begin{aligned}&\partial_t r+v\partial_x j=0,\\&\partial_t j+v\partial_x r=0.\end{aligned}\tag{2.4}$$

Now we use some discretization methods to deal with the two steps.

2.2.1. Relaxation step

To have a good stability property, considering the implicit discretization for the relaxation term (2.3), one can obtain

$$\begin{aligned}&\frac{r^*-r^n}{\tau}=-\frac{1}{\epsilon^2}(r^*-\rho^*),\\&\frac{j^*-j^n}{\tau}=-\frac{1}{\epsilon^2}\left(j^*+(1-\epsilon^2)v\partial_x r^*\right).\end{aligned}\tag{2.5}$$

We remark that the above system can be implemented explicitly on a classical computer since ρ is preserved, i.e., $\rho^*=\rho^n$, and hence

$$\begin{aligned}&\frac{r^*-r^n}{\tau}=-\frac{1}{\epsilon^2}(r^*-\rho^n),\\&\frac{j^*-j^n}{\tau}=-\frac{1}{\epsilon^2}\left(j^*+(1-\epsilon^2)v\partial_x r^*\right),\end{aligned}$$

where the integral given by ρ will be approximated by the Gaussian quadrature rule

$$\rho(t,x)=\int_0^1 r(t,v,x)\mathrm{d}v \approx \sum_{k=1}^{N} w_k r(t,v_k,x)$$

with (v_k,w_k) being the Gaussian quadrature points and weights on $[0,1]$.

The spatial mesh of x is defined as $x_0 < x_1 < \cdots < x_{N_x} < x_{N_x+1}$ (where x_0 and x_{N_x+1} are boundary points), and the discrete time are $t_0 < t_1 < \cdots < t_{N_t}$. Let u_{km} be the approximation to $u(v_k,x_m)$. Then the discrete scheme of (2.5) is

$$\begin{aligned} r_{km}^* &= \frac{1}{1+\tau/\epsilon^2}\left(r_{km}^n+\frac{\tau}{\epsilon^2}\rho_m^n\right),\\ j_{km}^* &= \frac{1}{1+\tau/\epsilon^2}\left(j_{km}^n-\frac{\tau}{\epsilon^2}(1-\epsilon^2)v_k\frac{r_{k,m+1}^*-r_{k,m-1}^*}{2h}\right), \end{aligned} \tag{2.6}$$

where $k=1,\ldots,N$ and $m=1,\ldots,N_x$. For fixed v_k, we define $\boldsymbol{u}_k=[u_{k1},u_{k2},\cdots,u_{k,N_x}]^T$ and write (2.6) in vector form as

$$\begin{aligned} \boldsymbol{r}_k^* &= \frac{1}{1+\tau/\epsilon^2}\left(\boldsymbol{r}_k^n+\frac{\tau}{\epsilon^2}(w_1\boldsymbol{r}_1^n+\cdots+w_N\boldsymbol{r}_N^n)\right),\\ \boldsymbol{j}_k^* &= \frac{1}{1+\tau/\epsilon^2}\left(\boldsymbol{j}_k^n-\frac{\tau}{2h\epsilon^2}(1-\epsilon^2)v_k\boldsymbol{M}_h\boldsymbol{r}_k^*-\frac{\tau}{2h\epsilon^2}(1-\epsilon^2)v_k\widetilde{\boldsymbol{b}}_k^*\right), \end{aligned} \qquad k=1,2,\ldots,N,$$

where

$$\boldsymbol{M}_h=\begin{bmatrix} 0 & 1 & & & \\ -1 & 0 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 0 & 1\\ & & & -1 & 0 \end{bmatrix}_{N_x\times N_x}, \quad \widetilde{\boldsymbol{b}}_k(t)=\begin{bmatrix} -r_{k0}(t)\\ 0\\ \vdots\\ 0\\ r_{k,N_x+1}(t) \end{bmatrix}.$$

Let $\boldsymbol{r}=[\boldsymbol{r}_1;\boldsymbol{r}_2;\cdots;\boldsymbol{r}_N]$, where ";" indicates the straightening of $\{\boldsymbol{r}^i\}_{i\geq 1}$ into a column vector. Then,

$$\begin{aligned} \boldsymbol{r}^* &= \frac{1}{1+\tau/\epsilon^2}\left(\boldsymbol{I}+\frac{\tau}{\epsilon^2}W\otimes\boldsymbol{I}\right)\boldsymbol{r}^n,\\ \boldsymbol{j}^* &= \frac{1}{1+\tau/\epsilon^2}\left(\boldsymbol{j}^n-\frac{\tau}{2h\epsilon^2}(1-\epsilon^2)\boldsymbol{V}\otimes\boldsymbol{M}_h\boldsymbol{r}^*-\frac{\tau}{2h\epsilon^2}(1-\epsilon^2)\widetilde{\boldsymbol{b}}_v^*\right), \end{aligned}$$

where $\widetilde{\boldsymbol{b}}_v^*=[v_1\widetilde{\boldsymbol{b}}_1;\cdots;v_N\widetilde{\boldsymbol{b}}_N]$, $\boldsymbol{V}=\mathrm{diag}([v_1,v_2,\cdots,v_N])$, and

$$W=\begin{bmatrix} w_1 & w_2 & \cdots & w_N\\ w_1 & w_2 & \cdots & w_N\\ \vdots & \vdots & \ddots & \vdots\\ w_1 & w_2 & \cdots & w_N \end{bmatrix}_{N\times N}.$$

DOI https://doi.org/10.4208/eajam.2022-315.200423 | Generated on 2024-10-16 08:29:06

2.2.2. Transport step

For the transport step (2.4), by introducing the Riemann invariants $U = r + j$ and $V = r - j$, one can obtain

$$\begin{aligned} \partial_t U + v\partial_x U = 0, \\ \partial_t V - v\partial_x V = 0. \end{aligned}$$

Applying the upwind scheme to the spatial derivative gives

$$\begin{aligned} r_{km}^{n+1} &= (1-\lambda v_k) r_{km}^* + \frac{\lambda v_k}{2}\big(r_{k,m+1}^* + r_{k,m-1}^*\big) - \frac{\lambda v_k}{2}\big(j_{k,m+1}^* - j_{k,m-1}^*\big), \\ j_{km}^{n+1} &= (1-\lambda v_k) j_{km}^* + \frac{\lambda v_k}{2}\big(j_{k,m+1}^* + j_{k,m-1}^*\big) - \frac{\lambda v_k}{2}\big(r_{k,m+1}^* - r_{k,m-1}^*\big), \end{aligned} \tag{2.7}$$

where $\lambda = \tau/h$. Similarly, for the fixed k, one has

$$\begin{aligned} \boldsymbol{r}_k^{n+1} &= \left(\boldsymbol{I} + \frac{\lambda v_k}{2}\boldsymbol{L}_h\right)\boldsymbol{r}_k^* - \frac{\lambda v_k}{2}\boldsymbol{M}_h \boldsymbol{j}_k^* + \frac{\lambda v_k}{2}\left(\boldsymbol{b}_k^* - \widetilde{\boldsymbol{c}}_k^*\right), \\ \boldsymbol{j}_k^{n+1} &= \left(\boldsymbol{I} + \frac{\lambda v_k}{2}\boldsymbol{L}_h\right)\boldsymbol{j}_k^* - \frac{\lambda v_k}{2}\boldsymbol{M}_h \boldsymbol{r}_k^* + \frac{\lambda v_k}{2}\left(\boldsymbol{c}_k^* - \widetilde{\boldsymbol{b}}_k^*\right), \end{aligned} \qquad k = 1, 2, \ldots, N,$$

where

$$\boldsymbol{L}_h = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & & 1 \\ & & & 1 & -2 \end{bmatrix}, \quad \boldsymbol{b}_k = \begin{bmatrix} r_{k0} \\ 0 \\ \vdots \\ 0 \\ r_{k,N_x+1} \end{bmatrix}, \quad \boldsymbol{c}_k = \begin{bmatrix} j_{k0} \\ 0 \\ \vdots \\ 0 \\ j_{k,N_x+1} \end{bmatrix}, \quad \widetilde{\boldsymbol{c}}_k = \begin{bmatrix} -j_{k0} \\ 0 \\ \vdots \\ 0 \\ j_{k,N_x+1} \end{bmatrix}.$$

We rewrite the scheme in matrix form

$$\begin{aligned} \boldsymbol{r}^{n+1} &= \left(\boldsymbol{I} + \frac{\lambda}{2}\boldsymbol{V} \otimes \boldsymbol{L}_h\right)\boldsymbol{r}^* - \frac{\lambda}{2}\boldsymbol{V} \otimes \boldsymbol{M}_h \boldsymbol{j}^* + \frac{\lambda}{2}\boldsymbol{f}_v^*, \\ \boldsymbol{j}^{n+1} &= \left(\boldsymbol{I} + \frac{\lambda}{2}\boldsymbol{V} \otimes \boldsymbol{L}_h\right)\boldsymbol{j}^* - \frac{\lambda}{2}\boldsymbol{V} \otimes \boldsymbol{M}_h \boldsymbol{r}^* + \frac{\lambda}{2}\boldsymbol{g}_v^*, \end{aligned}$$

where

$$\boldsymbol{f}_v^* = \begin{bmatrix} v_1(\boldsymbol{b}_1^* - \widetilde{\boldsymbol{c}}_1^*) \\ \vdots \\ v_N(\boldsymbol{b}_N^* - \widetilde{\boldsymbol{c}}_N^*) \end{bmatrix}, \quad \boldsymbol{g}_v^* = \begin{bmatrix} v_1(\boldsymbol{c}_1^* - \widetilde{\boldsymbol{b}}_1^*) \\ \vdots \\ v_N(\boldsymbol{c}_N^* - \widetilde{\boldsymbol{b}}_N^*) \end{bmatrix}.$$

3. Time Complexity Analysis of AP Scheme

3.1. Quantum difference method

Let $\boldsymbol{A}=(\lambda/2)\boldsymbol{V}\otimes\boldsymbol{M}_h$, $\boldsymbol{B}=\boldsymbol{I}+(\lambda/2)\boldsymbol{V}\otimes\boldsymbol{L}_h$ and $\gamma=\tau/\epsilon^2$. Substituting $\boldsymbol{r}^*$ and $\boldsymbol{j}^*$ into $\boldsymbol{r}^{n+1}$, we obtain

$$\begin{aligned}
\boldsymbol{r}^{n+1}&=\boldsymbol{B}\boldsymbol{r}^*-\boldsymbol{A}\boldsymbol{j}^*+\frac{\lambda}{2}\boldsymbol{f}_v^*\\
&=\boldsymbol{B}\boldsymbol{r}^*-\frac{1}{1+\gamma}\boldsymbol{A}\left(\boldsymbol{j}^n-\frac{(1-\epsilon^2)}{\epsilon^2}\boldsymbol{A}\boldsymbol{r}^*-\frac{\lambda(1-\epsilon^2)}{2\epsilon^2}\widetilde{\boldsymbol{b}}_v^*\right)+\frac{\lambda}{2}\boldsymbol{f}_v^*\\
&=\left(\boldsymbol{B}+\frac{1}{1+\gamma}\frac{(1-\epsilon^2)}{\epsilon^2}\boldsymbol{A}^2\right)\boldsymbol{r}^*-\frac{1}{1+\gamma}\boldsymbol{A}\boldsymbol{j}^n+\frac{1}{1+\gamma}\frac{\lambda(1-\epsilon^2)}{2\epsilon^2}\boldsymbol{A}\widetilde{\boldsymbol{b}}_v^*+\frac{\lambda}{2}\boldsymbol{f}_v^*\\
&=\frac{1}{1+\gamma}\left(\boldsymbol{B}+\frac{1}{1+\gamma}\frac{(1-\epsilon^2)}{\epsilon^2}\boldsymbol{A}^2\right)(\boldsymbol{I}+\gamma\boldsymbol{W}\otimes\boldsymbol{I})\boldsymbol{r}^n-\frac{1}{1+\gamma}\boldsymbol{A}\boldsymbol{j}^n\\
&\quad+\frac{1}{1+\gamma}\frac{\lambda(1-\epsilon^2)}{2\epsilon^2}\boldsymbol{A}\widetilde{\boldsymbol{b}}_v^*+\frac{\lambda}{2}\boldsymbol{f}_v^*\\
&=\frac{1}{1+\gamma}\left(\boldsymbol{B}+\frac{1-\epsilon^2}{\tau+\epsilon^2}\boldsymbol{A}^2\right)(\boldsymbol{I}+\gamma\boldsymbol{W}\otimes\boldsymbol{I})\boldsymbol{r}^n-\frac{1}{1+\gamma}\boldsymbol{A}\boldsymbol{j}^n+\frac{\lambda(1-\epsilon^2)}{2(\tau+\epsilon^2)}\boldsymbol{A}\widetilde{\boldsymbol{b}}_v^*+\frac{\lambda}{2}\boldsymbol{f}_v^*\\
&=:\boldsymbol{B}_1\boldsymbol{r}^n-\boldsymbol{A}_1\boldsymbol{j}^n+\widetilde{\boldsymbol{f}}^{n+1},
\end{aligned}$$

where

$$\boldsymbol{B}_1=\frac{1}{1+\gamma}\left(\boldsymbol{B}+\frac{1-\epsilon^2}{\tau+\epsilon^2}\boldsymbol{A}^2\right)(\boldsymbol{I}+\gamma\boldsymbol{W}\otimes\boldsymbol{I}),\quad \boldsymbol{A}_1=\frac{1}{1+\gamma}\boldsymbol{A}.$$

A similar calculation gives

$$\begin{aligned}
\boldsymbol{j}^{n+1}&=\boldsymbol{B}\boldsymbol{j}^*-\boldsymbol{A}\boldsymbol{r}^*+\frac{\lambda}{2}\boldsymbol{g}_v^*\\
&=\frac{1}{1+\gamma}\boldsymbol{B}\left(\boldsymbol{j}^n-\frac{(1-\epsilon^2)}{\epsilon^2}\boldsymbol{A}\boldsymbol{r}^*-\frac{\lambda(1-\epsilon^2)}{2\epsilon^2}\widetilde{\boldsymbol{b}}_v^*\right)-\boldsymbol{A}\boldsymbol{r}^*+\frac{\lambda}{2}\boldsymbol{g}_v^*\\
&=\frac{1}{1+\gamma}\boldsymbol{B}\boldsymbol{j}^n-\left(\frac{1}{1+\gamma}\frac{1-\epsilon^2}{\epsilon^2}\boldsymbol{B}\boldsymbol{A}+\boldsymbol{A}\right)\boldsymbol{r}^*-\frac{1}{1+\gamma}\frac{\lambda(1-\epsilon^2)}{2\epsilon^2}\boldsymbol{B}\widetilde{\boldsymbol{b}}_v^*+\frac{\lambda}{2}\boldsymbol{g}_v^*\\
&=\frac{1}{1+\gamma}\boldsymbol{B}\boldsymbol{j}^n-\frac{1}{1+\gamma}\left(\frac{1-\epsilon^2}{\tau+\epsilon^2}\boldsymbol{B}\boldsymbol{A}+\boldsymbol{A}\right)(\boldsymbol{I}+\gamma\boldsymbol{W}\otimes\boldsymbol{I})\boldsymbol{r}^n-\frac{\lambda(1-\epsilon^2)}{2(\tau+\epsilon^2)}\boldsymbol{B}\widetilde{\boldsymbol{b}}_v^*+\frac{\lambda}{2}\boldsymbol{g}_v^*\\
&=:\boldsymbol{A}_2\boldsymbol{j}^n-\boldsymbol{B}_2\boldsymbol{r}^n+\widetilde{\boldsymbol{g}}^{n+1},
\end{aligned}$$

where

$$\boldsymbol{B}_2=\frac{1}{1+\gamma}\left(\boldsymbol{A}+\frac{1-\epsilon^2}{\tau+\epsilon^2}\boldsymbol{B}\boldsymbol{A}\right)(\boldsymbol{I}+\gamma\boldsymbol{W}\otimes\boldsymbol{I}),\quad \boldsymbol{A}_2=\frac{1}{1+\gamma}\boldsymbol{B}.$$

Introducing the notations

$$\boldsymbol{S}_1=\left[\boldsymbol{r}^1;\boldsymbol{r}^2;\cdots;\boldsymbol{r}^{N_t}\right],\quad \boldsymbol{S}_2=\left[\boldsymbol{j}^1;\boldsymbol{j}^2;\cdots;\boldsymbol{j}^{N_t}\right],\quad \boldsymbol{S}=[\boldsymbol{S}_1;\boldsymbol{S}_2],$$

one obtains the linear system

$$\boldsymbol{L}\boldsymbol{S}=\boldsymbol{F},$$

where $\boldsymbol{L}=(\boldsymbol{L}_{ij})_{2\times 2}$ and $\boldsymbol{F}=[\boldsymbol{F}_1;\boldsymbol{F}_2]$, with

$$\boldsymbol{L}_{11}=\begin{bmatrix} \boldsymbol{I} & & & \\ -\boldsymbol{B}_1 & \boldsymbol{I} & & \\ & \ddots & \ddots & \\ & & -\boldsymbol{B}_1 & \boldsymbol{I} \end{bmatrix},\quad \boldsymbol{L}_{12}=\begin{bmatrix} \boldsymbol{O} & & & \\ \boldsymbol{A}_1 & \boldsymbol{O} & & \\ & \ddots & \ddots & \\ & & \boldsymbol{A}_1 & \boldsymbol{O} \end{bmatrix},\quad \boldsymbol{F}_1=\begin{bmatrix} \widetilde{\boldsymbol{f}}^1+\boldsymbol{B}_1\boldsymbol{r}^0-\boldsymbol{A}_1\boldsymbol{j}^0 \\ \widetilde{\boldsymbol{f}}^2 \\ \vdots \\ \widetilde{\boldsymbol{f}}^{N_t} \end{bmatrix},$$

$$\boldsymbol{L}_{21}=\begin{bmatrix} \boldsymbol{O} & & & \\ \boldsymbol{B}_2 & \boldsymbol{O} & & \\ & \ddots & \ddots & \\ & & \boldsymbol{B}_2 & \boldsymbol{O} \end{bmatrix},\quad \boldsymbol{L}_{22}=\begin{bmatrix} \boldsymbol{I} & & & \\ -\boldsymbol{A}_2 & \boldsymbol{I} & & \\ & \ddots & \ddots & \\ & & -\boldsymbol{A}_2 & \boldsymbol{I} \end{bmatrix},\quad \boldsymbol{F}_2=\begin{bmatrix} \widetilde{\boldsymbol{g}}^1-\boldsymbol{B}_2\boldsymbol{r}^0+\boldsymbol{A}_2\boldsymbol{j}^0 \\ \widetilde{\boldsymbol{g}}^2 \\ \vdots \\ \widetilde{\boldsymbol{g}}^{N_t} \end{bmatrix}.$$

For fixed step sizes τ and h, when $\epsilon\to 0$, one has

$$\frac{1}{1+\gamma}\to 0,\quad \frac{\gamma}{1+\gamma}\to 1,$$

and hence

$$\boldsymbol{B}_2=\left(\boldsymbol{A}+\frac{1-\epsilon^2}{\tau+\epsilon^2}\boldsymbol{B}\boldsymbol{A}\right)\frac{1}{1+\gamma}(\boldsymbol{I}+\gamma\boldsymbol{W}\otimes\boldsymbol{I})\to\left(\boldsymbol{A}+\frac{1}{\tau}\boldsymbol{B}\boldsymbol{A}\right)\boldsymbol{W}\otimes\boldsymbol{I}.$$

Considering the amplification factor τ^{-1}, as in [16] we reformulate the linear system as

$$\begin{bmatrix} \boldsymbol{L}_{11} & \tau^{-1}\boldsymbol{L}_{12} \\ \tau\boldsymbol{L}_{21} & \boldsymbol{L}_{22} \end{bmatrix}\begin{bmatrix} \tau^{-1}\boldsymbol{S}_1 \\ \boldsymbol{S}_2 \end{bmatrix}=\begin{bmatrix} \tau^{-1}\boldsymbol{F}_1 \\ \boldsymbol{F}_2 \end{bmatrix},\tag{3.1}$$

where $\tilde{\boldsymbol{S}}_1=\tau^{-1}\boldsymbol{S}_1$ and $\tilde{\boldsymbol{S}}_2=\boldsymbol{S}_2$. This means we consider a linear system with new variables $\tilde{r}=\tau^{-1}r$ and $\tilde{j}=j$.

Remark 3.1. For the output of quantum algorithms, we are not interested in the quantum state but the induced physical observables, defined as

$$\langle O\rangle:=\int_{-1}^{1}G(v)f(v)\mathrm{d}v,$$

which yields, for examples, the density and flux by taking for $G(v)=1,v$ respectively. One easily finds that

$$\langle O\rangle=\int_{0}^{1}G(v)r(v)\mathrm{d}v.$$

This means that we only need to project the final quantum state onto $|\boldsymbol{r}\rangle$ when computing the above physical observables. This is referred to as the level set encoding, which could have quantum advantages even if one computes the physical observables (at least if one is only interested in solution in a submanifold of the computational domain). See [14].

On the other hand, we can reformulate the resulting linear system such that the query complexity in obtaining either $[\boldsymbol{S}_1, \boldsymbol{S}_2]$, $[\tau^{-1}\boldsymbol{S}_1, \boldsymbol{S}_2]$ or $[\boldsymbol{S}_1, \tau^{-1}\boldsymbol{S}_2]$ are the same. For details, please refer to the discussion below [16, Theorem 5.2].

3.2. Input model

In this article we will apply the optimal quantum linear systems algorithm (QLSA) proposed in [6] to solve the resulting linear system. Of course the specific quantum linear algebra solver is not unique. One could also use, for example, the block encoding algorithms [1]. To this end, let us first state the quantum linear systems problem.

Definition 3.1 (Quantum Linear Systems Problem (QLSP) [6]). *Consider the linear system $Ax = b$, where A is an $N \times N$ Hermitian matrix, $x = [x_0, \cdots, x_{N-1}]^T$, and $b = [b_0, \cdots, b_{N-1}]^T$. Assume x and b can be encoded as*

$$|x\rangle = \frac{1}{N_x}\sum_{i=0}^{N-1} x_i|i\rangle, \quad |b\rangle = \frac{1}{N_b}\sum_{i=0}^{N-1} b_i|i\rangle,$$

where $N_x = (x_0^2 + \cdots + x_{N-1}^2)^{1/2}$ and $N_b = (b_0^2 + \cdots + b_{N-1}^2)^{1/2}$ are normalization constants. Then the goal of QLSA is the following. When the access to A and $U_{initial}$ ($U_{initial}$ is a unitary operation such that $U_{initial}|0\rangle = |b\rangle$) are given, one aims to prepare a quantum state $|x'\rangle$ that is η-close to $|x\rangle$, i.e., $\||x'\rangle - |x\rangle\| \le \eta$.

In the above definition, the access to the matrix A refers to query the elements of A. The time complexity is usually measured by the number of calls of oracles for positions of nonzero entries of sparse matrices.

Definition 3.2 (Input Model for the Matrix [3, 14]). *Sparse access to a Hermitian matrix A is a 4-tuple $(s, \|A\|_{\max}, O_A, O_F)$, where s is the sparsity of A, and $\|A\|_{\max} = \max_{i,j}|A_{ij}|$. O_A and O_F are black boxes of unitary operations. O_A can access the matrix elements A_{ij} such that*

$$O_A|j\rangle|k\rangle|z\rangle = |j\rangle|k\rangle|z \oplus A_{jk}\rangle$$

for any $j, k \in \{1, 2, \ldots, N\} := [N]$, where the third register holds a bit string representing of A_{jk}. O_F allows to perform the map

$$O_F|j\rangle|l\rangle = |j\rangle|F(j,l)\rangle$$

for any $j \in [N]$ and $l \in [s]$, where the function F outputs the column index of the l-th non-zero elements in row j.

With these definitions, query complexity refers to the number of times oracles O_A, O_F and $U_{initial}$ are applied throughout the whole algorithm. Because the resulting coefficient matrices for partial differential equations are usually sparse, query complexity is often used to measure the time complexity of quantum algorithms. It should be noted, however, that query complexity of some algorithms is not measured by the number of sparse unitary operations O_A, O_F and $U_{initial}$, but is in terms of calls to a block encoding of the coefficient matrix, defined as follows.

Definition 3.3 (Block Access to the Matrix [14, 20]). *Let A be a m-qubit Hermitian matrix, $\delta_A > 0$ and n_A is a positive integer. An $(m+n_A)$-qubit unitary matrix U_A is a $(\alpha_A, n_A, \delta_A)$-block encoding of A if*

$$\|A - \alpha_A \langle 0^{n_A}| U_A |0^{n_A}\rangle\| \leq \delta_A.$$

Block access to A is then the 4-tuple $(\alpha_A, n_A, \delta_A, U_A)$ where U_A is the unitary black-box block-encoding of A.

In some cases, the block access to A can be constructed from the sparse access to A. If standard methods are used to construct the block access from the sparse access, there will be a multiplicative factor s in the query complexity of block encoding. We also refer the reader to [3, 11, 14, 16, 24, 28] for detailed discussions.

We assume $\|A\|_{\max} < 1$ throughout the discussion, unless otherwise stated, since the complexity can have a contribution proportional to $\|A\|_{\max}$. Otherwise we can simply replace it by the re-scaled matrix A/α for some $\alpha > \|A\|_{\max}$, where $\|A\|_{\max} = \max_{ij} |A_{ij}|$.

3.3. Time complexity of the AP scheme

In this article we apply the optimal quantum linear systems algorithm (QLSA) proposed in [8] to solve the resulting linear system. The query complexity with respect to the sparse access to matrices can be written as

$$Q = \mathcal{O}\left(s\kappa \log\left(\frac{1}{\delta}\right)\right), \tag{3.2}$$

where s is the sparsity of the coefficient matrix, κ is the condition number, and δ is the error bound.

Theorem 3.1. *Suppose that the time step τ and the space step h satisfy $\tau/h^2 \leq 1/(1+h)$.*

1. *For sufficiently small ϵ, the singular value of the coefficient matrix in* (3.1) *satisfies*

$$\sigma_{\min} \gtrsim \frac{1}{N^{1/2} N_t} - \alpha(\epsilon), \quad \sigma_{\max} \lesssim N^{1/2} + \alpha(\epsilon),$$

where

$$\alpha(\epsilon) = \frac{\epsilon^2}{\tau + \epsilon^2}\left(N^{1/2}\tau + N^{1/2} + \tau + \frac{1}{\tau}\right) + \frac{\epsilon^2(1-\epsilon^2)}{(\epsilon^2+\tau)^2}(1+\tau) + \frac{\epsilon^2(\epsilon^2 + 2\tau + \tau^2)}{\tau(\epsilon^2+\tau)^2} N^{1/2}\left(1 + \frac{1}{\tau}\right),$$

which tends to zero as $\epsilon \to 0$.

2. *The sparsity and the condition number of the coefficient matrix satisfy* $s = \mathcal{O}(N)$ *and* $\kappa = \mathcal{O}(NN_t)$.

3. *The time complexities of the classical treatment and the quantum treatment for solving* (3.1) *are*

$$C = \mathcal{O}\left(N^2 N_t N_x\right), \quad Q = \mathcal{O}\left(N^2 N_t \log(N_x)\right).$$

Given the error bound δ, *we set* $h = \mathcal{O}(\delta)$. *If* $N_t = \mathcal{O}(N_x^2)$, *then*

$$C = \mathcal{O}\left(N^2 N_x^3\right) = \mathcal{O}\left(N^2 \delta^{-3}\right), \quad Q = \mathcal{O}\left(N^2 N_x^2 \log(N_x)\right) = \mathcal{O}\left(N^2 \delta^{-2} \log \delta^{-1}\right).$$

Proof. Since the problem is linear, one can apply the discrete Fourier transform to characterize the singular values of the coefficient matrix. In the following, we only consider the discrete Fourier transform for the spatial variables.

1. Introduce the following expressions:

$$r_{km}^n = \hat{r}_k^n \mathrm{e}^{\mathrm{i}m\xi h}, \quad j_{km}^n = \hat{j}_k^n \mathrm{e}^{\mathrm{i}m\xi h}, \quad r_{km}^* = \hat{r}_k^* \mathrm{e}^{\mathrm{i}m\xi h}, \quad j_{km}^* = \hat{j}_k^* \mathrm{e}^{\mathrm{i}m\xi h},$$

where ξ represents the frequency variable and $\mathrm{i} = \sqrt{-1}$. Plugging them in (2.6) and (2.7), one obtains

$$\hat{r}_k^* = \frac{1}{1+\gamma}\left(\hat{r}_k^n + \gamma \sum_{k'=1}^{N} w_{k'} \hat{r}_{k'}^n\right),$$
$$\hat{j}_k^* = \frac{1}{1+\gamma}\left(\hat{j}_k^n - \frac{1-\epsilon^2}{\epsilon^2}\mathrm{i}\lambda \sin(\xi h) v_k \hat{r}_k^*\right)$$

and

$$\hat{r}_k^{n+1} = (1-\lambda v_k)\hat{r}_k^* + \lambda \cos(\xi h) v_k \hat{r}_k^* - \mathrm{i}\lambda \sin(\xi h) v_k \hat{j}_k^*,$$
$$\hat{j}_k^{n+1} = (1-\lambda v_k)\hat{j}_k^* + \lambda \cos(\xi h) v_k \hat{j}_k^* - \mathrm{i}\lambda \sin(\xi h) v_k \hat{r}_k^*.$$

Eliminating r^* and j^* yields

$$\begin{aligned} &\hat{r}_k^{n+1} + c_{1,\epsilon}\hat{r}_k^n + c_{2,\epsilon}\hat{j}_k^n + \gamma c_{1,\epsilon} \sum_{k'=1}^{N} w_{k'} \hat{r}_{k'}^n = 0, \\ &\hat{j}_k^{n+1} + d_{1,\epsilon}\hat{j}_k^n + d_{2,\epsilon}\hat{r}_k^n + \gamma d_{2,\epsilon} \sum_{k'=1}^{N} w_{k'} \hat{r}_{k'}^n = 0, \end{aligned} \tag{3.3}$$

where $n = 0, 1, \ldots, N_t - 1$ and

$$c_{1,\epsilon} = -\frac{1}{1+\gamma}\big[(1-\lambda v_k) + \lambda v_k \cos(\xi h)\big] - \frac{1}{(1+\gamma)^2}\frac{1-\epsilon^2}{\epsilon^2}\big(\mathrm{i}\lambda v_k \sin(\xi h)\big)^2,$$
$$c_{2,\epsilon} = \frac{1}{1+\gamma}\mathrm{i}\lambda v_k \sin(\xi h),$$
$$d_{1,\epsilon} = -\frac{1}{1+\gamma}\big[(1-\lambda v_k) + \lambda v_k \cos(\xi h)\big]$$
$$d_{2,\epsilon} = \frac{1}{(1+\gamma)^2}\frac{1-\epsilon^2}{\epsilon^2}\big[(1-\lambda v_k) + \lambda v_k \cos(\xi h)\big]\mathrm{i}\lambda v_k \sin(\xi h) + \frac{1}{1+\gamma}\mathrm{i}\lambda v_k \sin(\xi h).$$

For the new variables, the linear system (3.3) should be changed to

$$\tau\tilde{r}_k^{n+1} + c_{1,\epsilon}\tau\tilde{r}_k^n + c_{2,\epsilon}\tilde{j}_k^n + \gamma c_{1,\epsilon}\tau\sum_{k'=1}^{N} w_{k'}\tilde{r}_{k'}^n = 0,$$
$$\tilde{j}_k^{n+1} + d_{1,\epsilon}\tilde{j}_k^n + d_{2,\epsilon}\tau\tilde{r}_k^n + \gamma d_{2,\epsilon}\tau\sum_{k'=1}^{N} w_{k'}\tilde{r}_{k'}^n = 0.$$

Let $\tilde{\boldsymbol{r}}_k = [\tilde{r}_k^1, \cdots, \tilde{r}_k^{N_t}]^T$, $\tilde{\boldsymbol{j}}_k = [\tilde{j}_k^1, \cdots, \tilde{j}_k^{N_t}]^T$, and

$$P = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}_{N_t \times N_t}.$$

Then Eq. (3.3) can be written as

$$\tau(I + c_{1,\epsilon}P)\tilde{\boldsymbol{r}}_k + c_{2,\epsilon}P\tilde{\boldsymbol{j}}_k + \tau\gamma c_{1,\epsilon}P(w_1\tilde{\boldsymbol{r}}_1 + \cdots + w_N\tilde{\boldsymbol{r}}_N) = \tilde{\boldsymbol{f}}_k,$$
$$(I + d_{1,\epsilon}P)\tilde{\boldsymbol{j}}_k + \tau d_{2,\epsilon}P\tilde{\boldsymbol{r}}_k + \tau\gamma d_{2,\epsilon}P(w_1\tilde{\boldsymbol{r}}_1 + \cdots + w_N\tilde{\boldsymbol{r}}_N) = \tilde{\boldsymbol{g}}_k,$$

where $k = 1, 2, \ldots, N$, and the right-hand vectors are

$$\tilde{\boldsymbol{f}}_k = -\left[\tau c_{1,\epsilon}\tilde{r}_k^0 + c_{2,\epsilon}\tilde{j}_k^0 + \tau\gamma c_{1,\epsilon}\left(w_1\tilde{r}_1^0 + \cdots + w_N\tilde{r}_N^0\right), 0, \cdots, 0\right]^T,$$
$$\tilde{\boldsymbol{g}}_k = -\left[d_{1,\epsilon}\tilde{j}_k^0 + \tau d_{2,\epsilon}\tilde{r}_k^0 + \tau\gamma d_{2,\epsilon}\left(w_1\tilde{r}_1^0 + \cdots + w_N\tilde{r}_N^0\right), 0, \cdots, 0\right]^T.$$

Let $\tilde{\boldsymbol{R}} = [\tilde{\boldsymbol{r}}_1; \tilde{\boldsymbol{r}}_2; \cdots; \tilde{\boldsymbol{r}}_N]$, $\tilde{\boldsymbol{J}} = [\tilde{\boldsymbol{j}}_1; \tilde{\boldsymbol{j}}_2; \cdots; \tilde{\boldsymbol{j}}_N]$ and $\tilde{\boldsymbol{S}} = [\tilde{\boldsymbol{R}}; \tilde{\boldsymbol{J}}]$, one obtains the linear system

$$\tilde{\boldsymbol{L}}_\epsilon\tilde{\boldsymbol{S}} = \tilde{\boldsymbol{F}},$$

where $\tilde{\boldsymbol{F}} = [\tilde{\boldsymbol{f}}_1/\tau; \cdots; \tilde{\boldsymbol{f}}_N/\tau; \tilde{\boldsymbol{g}}_1; \cdots; \tilde{\boldsymbol{g}}_N]$

$$\tilde{\boldsymbol{L}}_\epsilon = \begin{bmatrix} I_N \otimes (I + c_{1,\epsilon}P) + \gamma c_{1,\epsilon}W \otimes P & \tau^{-1}I_N \otimes (c_{2,\epsilon}P) \\ \tau(I_N \otimes (d_{2,\epsilon}P) + \gamma d_{2,\epsilon}W \otimes P) & I_N \otimes (I + d_{1,\epsilon}P) \end{bmatrix}.$$

2. In the following, we utilize the perturbation technique to analyze the condition number of the coefficient matrix. We first briefly explain the idea of the perturbation technique: Let $\tilde{\boldsymbol{L}}_\epsilon$ be the coefficient matrix and $\tilde{\boldsymbol{L}}_\epsilon = \tilde{\boldsymbol{L}}_0 + \boldsymbol{E}$, where $\tilde{\boldsymbol{L}}_0$ is the coefficient matrix with $\epsilon = 0$. By the Weyl's inequality [27],

$$\sigma_{\max}(\tilde{\boldsymbol{L}}_\epsilon) \le \sigma_{\max}(\tilde{\boldsymbol{L}}_0) + \|\boldsymbol{E}\|, \quad \sigma_{\min}(\tilde{\boldsymbol{L}}_\epsilon) \ge \sigma_{\min}(\tilde{\boldsymbol{L}}_0) - \|\boldsymbol{E}\|.$$

Thus, it suffices to determine the condition number of $\tilde{\boldsymbol{L}}_0$ and the upper bound of $\|\boldsymbol{E}\|$.

Let $\epsilon = 0$. One has

$$\frac{1}{1+\gamma_0} = 0, \quad \frac{\gamma_0}{1+\gamma_0} = 1,$$

where γ_0 corresponds to $\epsilon = 0$. A simple calculation shows that

$$\begin{aligned}
&c_{1,0} = c_{2,0} = d_{1,0} = d_{2,0} = 0,\\
&\gamma_0 c_{1,0} = -\big[(1-\lambda v_k) + \lambda v_k \cos(\xi h)\big] - \frac{1}{\tau}\big(\mathrm{i}\lambda v_k \sin(\xi h)\big)^2,\\
&\gamma_0 d_{2,0} = \frac{1}{\tau}\big[(1-\lambda v_k) + \lambda v_k \cos(\xi h)\big]\mathrm{i}\lambda v_k \sin(\xi h) + \mathrm{i}\lambda v_k \sin(\xi h),
\end{aligned}$$

and hence

$$\tilde{\boldsymbol{L}}_0 = \begin{bmatrix} I_N \otimes I_{N_t} & O \\ O & I_N \otimes I_{N_t} \end{bmatrix} + \begin{bmatrix} \gamma_0 c_{1,0} W \otimes P & O \\ \tau\gamma_0 d_{2,0} W \otimes P & O \end{bmatrix}.$$

For $\gamma_0 c_{1,0}$, one easily gets

$$\begin{aligned}
|\gamma_0 c_{1,0}| &= \left| -\big[(1-\lambda v_k) + \lambda v_k \cos(\xi h)\big] - \frac{1}{\tau}\big(\mathrm{i}\lambda v_k \sin(\xi h)\big)^2 \right| \\
&= \left| 1 - \lambda v_k - \frac{\tau}{h^2}\big(v_k \sin(\xi h)\big)^2 + \lambda v_k \cos(\xi h) \right| =: |a+b|,
\end{aligned}$$

where

$$a = 1 - \lambda v_k - \frac{\tau}{h^2}\big(v_k \sin(\xi h)\big)^2, \quad b = \lambda v_k \cos(\xi h).$$

Noting that

$$a = 1 - \lambda v_k - \frac{\tau}{h^2}\big(v_k \sin(\xi h)\big)^2 \geq 1 - \lambda - \frac{\tau}{h^2},$$

one has $a \geq 0$ when

$$\lambda + \frac{\tau}{h^2} \leq 1 \quad \text{or} \quad \frac{\tau}{h^2} \leq \frac{1}{1+h}. \tag{3.4}$$

Then,

$$|\gamma_0 c_{1,0}| = |a+b| \leq |a| + |b| = 1 - \lambda v_k\big(1 - |\cos(\xi h)|\big) - \frac{\tau}{h^2}\big(v_k \sin(\xi h)\big)^2 =: b',$$

where the right-hand side satisfies $b' \geq a \geq 0$ under the condition of (3.4), which also implies $b' \leq 1$. Let $c = \lambda v_k(1-\cos(\xi h))$. One has

$$\begin{aligned}
|\tau\gamma_0 d_{2,0}| &= |(1+\tau) - \lambda v_k(1-\cos(\xi h))| \cdot |\lambda v_k \sin(\xi h)| \\
&\leq |(1+\tau) - c| \leq \max\{|1+\tau-c_{\min}|, |1+\tau-c_{\max}|\} \\
&= \max\{|1+\tau+0|, |1+\tau-2|\} = 1+\tau.
\end{aligned}$$

3. We first consider the maximum singular value. Since $w_1 + \cdots + w_N = 1$, one can check that $WW^T = \|\boldsymbol{w}\|^2 \cdot \mathbf{1}_N$, where $\|\boldsymbol{w}\|^2 = w_1^2 + \cdots + w_N^2$, and $\mathbf{1}_N$ is the N-th order matrix with all entries being 1. Then,

$$\|WW^T\| \leq N\|\boldsymbol{w}\|^2 \leq N(w_1 + \cdots + w_N)^2 = N,$$

which gives $\|W\| \leq N^{1/2}$ and

$$\sigma_{\max}(\tilde{\boldsymbol{L}}_0) = \|\tilde{\boldsymbol{L}}_0\| \leq 1 + \max\{\gamma_0 c_{1,0}, \tau\gamma_0 d_{2,0}\}\|W\| \cdot \|P\| \lesssim N^{1/2}.$$

4. For the minimum singular value, since $\sigma_{\min}(\tilde{\boldsymbol{L}}_0) = 1/\|\tilde{\boldsymbol{L}}_0^{-1}\|$, we only need to provide a upper bound for $\|\tilde{\boldsymbol{L}}_0^{-1}\|$. By definition,

$$\big\|\tilde{\boldsymbol{L}}_0^{-1}\big\| = \max_{\|\boldsymbol{b}\|\le 1} \big\|\tilde{\boldsymbol{L}}_0^{-1}\boldsymbol{b}\big\|, \quad \boldsymbol{b} = [\boldsymbol{f};\boldsymbol{g}],$$

where $\tilde{\boldsymbol{L}}_0^{-1}\boldsymbol{b}$ clearly corresponds to the following linear system:

$$\tilde{r}_k^{n+1} + \gamma_0 c_{1,0} \sum_{k'=1}^{N} w_{k'} \tilde{r}_{k'}^n = \boldsymbol{f}_k^n,$$
$$\tilde{j}_k^{n+1} + \tau\gamma_0 c_{2,0} \sum_{k'=1}^{N} w_{k'} \tilde{r}_{k'}^n = \boldsymbol{g}_k^n,$$

which can be written in matrix form as

$$\tilde{\boldsymbol{r}}^{n+1} = A\tilde{\boldsymbol{r}}^n + \boldsymbol{f}^n, \quad A = -\gamma_0 c_{1,0} W,$$
$$\tilde{\boldsymbol{j}}^{n+1} = B\tilde{\boldsymbol{r}}^n + \boldsymbol{g}^n, \quad B = -\tau\gamma_0 c_{2,0} W.$$

Assume the maximum value is attained at $\boldsymbol{b}$. Since $|\gamma_0 c_{1,0}| \le 1$, $\|W\| \le N^{1/2}$ and $W^2 = W$,

$$\|\tilde{\boldsymbol{r}}^n\| \le \|A^n\|\|\boldsymbol{f}^0\| + \|A^{n-1}\|\|\boldsymbol{f}^1\| + \cdots + \|\boldsymbol{f}^{n-1}\| \le N^{1/2}\big(\|\boldsymbol{f}^0\| + 1\big).$$

Similarly, we obtain from $\tau\gamma_0 c_{2,0} \le 1 + \tau$ that

$$\|\tilde{\boldsymbol{j}}^n\| \le (1+\tau)^n N^{1/2}\big(\|\boldsymbol{g}^0\| + 1\big) \lesssim N^{1/2}\big(\|\boldsymbol{g}^0\| + 1\big).$$

Thus,

$$\big\|\tilde{\boldsymbol{L}}_0^{-1}\big\| = \big\|\tilde{\boldsymbol{L}}_0^{-1}\boldsymbol{b}\big\| \lesssim N^{1/2} N_t\big(\tilde{\boldsymbol{r}}^0 + 1\big),$$

which shows $\sigma_{\min}(\tilde{\boldsymbol{L}}_0) \gtrsim 1/(N^{1/2}N_t)$.

5. Now we calculate the L^2 norm $\|\tilde{\boldsymbol{L}}_\epsilon - \tilde{\boldsymbol{L}}_0\|_2$ for the perturbation term, where

$$\tilde{\boldsymbol{L}}_\epsilon - \tilde{\boldsymbol{L}}_0 = \begin{bmatrix} I_N \otimes c_{1,\epsilon} P + (\gamma c_{1,\epsilon} - \gamma_0 c_{1,0}) W \otimes P & \tau^{-1} I_N \otimes (c_{2,\epsilon} P) \\ \tau(I_N \otimes d_{2,\epsilon} P + (\gamma d_{2,\epsilon} - \gamma_0 d_{2,0}) W \otimes P) & I_N \otimes d_{1,\epsilon} P \end{bmatrix}.$$

A straightforward calculation gives

$$|c_{1,\epsilon}| \lesssim \frac{\epsilon^2}{\tau+\epsilon^2} + \frac{\epsilon^2(1-\epsilon^2)}{(\epsilon^2+\tau)^2},$$
$$|\gamma c_{1,\epsilon} - \gamma_0 c_{1,0}| \lesssim \frac{\epsilon^2}{\tau+\epsilon^2} + \frac{\epsilon^2(\epsilon^2+2\tau+\tau^2)}{\tau(\epsilon^2+\tau)^2},$$
$$|\tau^{-1} c_{2,\epsilon}| \lesssim \frac{\epsilon^2}{\tau(\epsilon^2+\tau)},$$

$$|\tau d_{2,\epsilon}| \lesssim \frac{\tau(1-\epsilon^2)\epsilon^2}{(\epsilon^2+\tau)^2} + \frac{\tau\epsilon^2}{\epsilon^2+\tau},$$
$$|\tau(\gamma d_{2,\epsilon} - \gamma_0 d_{2,0})| \lesssim \frac{\epsilon^2(\epsilon^2+2\tau+\tau^2)}{(\epsilon^2+\tau)^2} + \frac{\tau\epsilon^2}{\epsilon^2+\tau},$$
$$|d_{1,\epsilon}| \lesssim \frac{\epsilon^2}{\tau+\epsilon^2}.$$

Since $\|W\| \le N^{1/2}$ and $\|P\| \le 1$, there holds

$$\|I_N \otimes c_{1,\epsilon} P\| \le \|I_N\|\|c_{1,\epsilon} P\| \le |c_{1,\epsilon}|,$$
$$\|(\gamma c_{1,\epsilon} - \gamma_0 c_{1,0}) W \otimes P\| \le (\gamma c_{1,\epsilon} - \gamma_0 c_{1,0})\|W\|\|P\| \le N^{1/2}|\gamma c_{1,\epsilon} - \gamma_0 c_{1,0}|,$$
$$\|\tau^{-1}(I_N \otimes c_{2,\epsilon} P)\| \le \tau^{-1} c_{2,\epsilon}\|I_N\|\|P\| \le |\tau^{-1} c_{2,\epsilon}|,$$
$$\|\tau I_N \otimes d_{2,\epsilon} P\| \le \tau d_{2,\epsilon}\|I_N\|\|P\| \le |\tau d_{2,\epsilon}|,$$
$$\|\tau(\gamma d_{2,\epsilon} - \gamma_0 d_{2,0}) W \otimes P)\| \le \tau(\gamma d_{2,\epsilon} - \gamma_0 d_{2,0})\|W\|\|P\| \le N^{1/2}|\tau(\gamma d_{2,\epsilon} - \gamma_0 d_{2,0})|,$$
$$\|I_N \otimes d_{1,\epsilon} P\| \le d_{1,\epsilon}\|I_N\|\|P\| \le |d_{1,\epsilon}|,$$

and

$$\|\tilde{\boldsymbol{L}}_{11,\epsilon} - \tilde{\boldsymbol{L}}_{11,0}\| \le |c_{1,\epsilon}| + N^{1/2}|\gamma c_{1,\epsilon} - \gamma_0 c_{1,0}|,$$
$$\|\tilde{\boldsymbol{L}}_{12,\epsilon} - \tilde{\boldsymbol{L}}_{12,0}\| \le |\tau^{-1} c_{2,\epsilon}|,$$
$$\|\tilde{\boldsymbol{L}}_{21,\epsilon} - \tilde{\boldsymbol{L}}_{21,0}\| \le |\tau d_{2,\epsilon}| + N^{1/2}|\tau(\gamma d_{2,\epsilon} - \gamma_0 d_{2,0})|,$$
$$\|\tilde{\boldsymbol{L}}_{22,\epsilon} - \tilde{\boldsymbol{L}}_{22,0}\| \le |d_{1,\epsilon}|.$$

This means $\|\tilde{\boldsymbol{L}}_\epsilon - \tilde{\boldsymbol{L}}_0\| \le \alpha(\epsilon)$, with

$$\alpha(\epsilon) = \frac{\epsilon^2}{\tau+\epsilon^2}\left(N^{1/2}\tau + N^{1/2} + \tau + \frac{1}{\tau}\right) + \frac{\epsilon^2(1-\epsilon^2)}{(\epsilon^2+\tau)^2}(1+\tau) + \frac{\epsilon^2(\epsilon^2+2\tau+\tau^2)}{\tau(\epsilon^2+\tau)^2} N^{1/2}\left(1+\frac{1}{\tau}\right).$$

6. Finally, we analyze the time complexity. The classical algorithm is to iteratively solve the following equations:

$$\boldsymbol{r}^{n+1} = \boldsymbol{B}_1 \boldsymbol{r}^n - \boldsymbol{A}_1 \boldsymbol{j}^n + \widetilde{\boldsymbol{f}}^{n+1},$$
$$\boldsymbol{j}^{n+1} = \boldsymbol{A}_2 \boldsymbol{j}^n - \boldsymbol{B}_2 \boldsymbol{r}^n + \widetilde{\boldsymbol{g}}^{n+1}.$$

The sparsity of $\boldsymbol{B}_i$ and $\boldsymbol{A}_i$ is $\mathcal{O}(N)$ and the matrix order is $\mathcal{O}(NN_x)$. Thus the time complexity of each iteration step is $\mathcal{O}(N^2 N_x)$, and the time complexity after N_t iterations is

$$C = \mathcal{O}\left(N^2 N_t N_x\right).$$

For the quantum treatment, by the estimates of singular values, the condition number $\kappa = \mathcal{O}(NN_t)$, and the sparsity $s = \mathcal{O}(N)$. Under the given conditions, the error bound $\delta = \mathcal{O}(h)$.

Plug these quantities in (3.2) to get

$$Q = \mathcal{O}\left(s\kappa \log\left(\frac{1}{\delta}\right)\right) = \mathcal{O}\left(N^2 N_t \log N_x\right).$$

This completes the proof. □

4. Time Complexity of the Explicit Scheme

4.1. Time complexity analysis

As a comparison, in this section we discuss the time complexity of the explicit scheme for both the classical and quantum treatments. We use the upwind finite difference to discretize (1.1). The upwind scheme is

$$\begin{aligned}&\frac{f_{k,m}^{n+1}-f_{k,m}^{n}}{\tau}+\frac{1}{\epsilon}v_k^{+}\frac{f_{k,m}^{n}-f_{k,m-1}^{n}}{h}+\frac{1}{\epsilon}v_k^{-}\frac{f_{k,m+1}^{n}-f_{k,m}^{n}}{h}\\&=\frac{1}{\epsilon^2}\left(\frac{1}{2}\sum_{k'=-N}^{N}w_{k'}f_{k',m}^{n}-f_{k,m}^{n}\right),\end{aligned}\tag{4.1}$$

or

$$f_{k,m}^{n+1}-\left[1-\frac{\lambda}{\epsilon}\left(v_k^{+}-v_k^{-}\right)-\frac{\tau}{\epsilon^2}\right]f_{k,m}^{n}-\frac{\lambda}{\epsilon}v_k^{+}f_{k,m-1}^{n}+\frac{\lambda}{\epsilon}v_k^{-}f_{k,m+1}^{n}-\frac{\tau}{2\epsilon^2}\sum_{k'}w_{k'}f_{k',m}^{n}=0,$$

where $\lambda=\tau/h$,

$$\alpha^{+}=\max\{\alpha,0\}=\frac{\alpha+|\alpha|}{2}\geq 0,\quad \alpha^{-}=\min\{\alpha,0\}=\frac{\alpha-|\alpha|}{2}\leq 0.$$

Let

$$\boldsymbol{f}_m=\left[f_{-N,m},f_{-N+1,m},f_{-N+2,m},\cdots,f_{N-2,m},f_{N-1,m},f_{N,m}\right]^T$$

and define

$$c_k=1-\frac{\lambda}{\epsilon}\left(v_k^{+}-v_k^{-}\right)-\frac{\tau}{\epsilon^2}.$$

Then (4.1) can be written as

$$\boldsymbol{f}_m^{n+1}-\boldsymbol{C}\boldsymbol{f}_m^{n}-\frac{\lambda}{\epsilon}\boldsymbol{V}^{+}\boldsymbol{f}_{m-1}^{n}+\frac{\lambda}{\epsilon}\boldsymbol{V}^{-}\boldsymbol{f}_{m+1}^{n}-\frac{\tau}{2\epsilon^2}\boldsymbol{W}\boldsymbol{f}_m^{n}=\boldsymbol{0},$$

where

$$\begin{aligned}\boldsymbol{C}&=\operatorname{diag}(c_{-N},\cdots,c_{-1},c_1,\cdots,c_N),\\\boldsymbol{V}^{\pm}&=\operatorname{diag}(v_{-N}^{\pm},\cdots,v_{-1}^{\pm},v_1^{\pm},\cdots,v_N^{\pm}),\end{aligned}$$

and

$$\boldsymbol{W}=\begin{bmatrix}w_{-N}&\cdots&w_{-1}&w_1&\cdots&w_N\\\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\w_{-N}&\cdots&w_{-1}&w_1&\cdots&w_N\end{bmatrix}_{2N\times 2N}.$$

Let $\boldsymbol{f}^n = [\boldsymbol{f}_1^n; \boldsymbol{f}_2^n; \cdots; \boldsymbol{f}_{N_x}^n]$. One has

$$\boldsymbol{f}^{n+1} - \boldsymbol{B}\boldsymbol{f}^n = \boldsymbol{b}^n,$$

where

$$\boldsymbol{b}^n = \frac{\lambda}{\epsilon}\left[\boldsymbol{V}^+ \boldsymbol{f}_0^n; \boldsymbol{0}; \cdots; \boldsymbol{0}; -\boldsymbol{V}^- \boldsymbol{f}_{N_x+1}^n\right]$$

and

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{C} + \frac{\tau}{2\epsilon^2}\boldsymbol{W} & -\frac{\lambda}{\epsilon}\boldsymbol{V}^- & & & \\ \frac{\lambda}{\epsilon}\boldsymbol{V}^+ & \boldsymbol{C} + \frac{\tau}{2\epsilon^2}\boldsymbol{W} & -\frac{\lambda}{\epsilon}\boldsymbol{V}^- & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{\lambda}{\epsilon}\boldsymbol{V}^+ & \boldsymbol{C} + \frac{\tau}{2\epsilon^2}\boldsymbol{W} & -\frac{\lambda}{\epsilon}\boldsymbol{V}^- \\ & & & \frac{\lambda}{\epsilon}\boldsymbol{V}^+ & \boldsymbol{C} + \frac{\tau}{2\epsilon^2}\boldsymbol{W} \end{bmatrix}.$$

Let $\boldsymbol{U} = [\boldsymbol{f}^1; \boldsymbol{f}^2; \cdots; \boldsymbol{f}^{N_t}]$. The linear system for the quantum difference approach can be written as

$$\boldsymbol{L}\boldsymbol{U} = \boldsymbol{F}, \tag{4.2}$$

where

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{I} & & & \\ -\boldsymbol{B} & \boldsymbol{I} & & \\ & \ddots & \ddots & \\ & & -\boldsymbol{B} & \boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix} \boldsymbol{b}^0 + \boldsymbol{B}\boldsymbol{f}^0 \\ \boldsymbol{b}^1 \\ \vdots \\ \boldsymbol{b}^{N_t-1}, \end{bmatrix}.$$

Theorem 4.1. *Let δ be the error bound and let $h = \mathcal{O}(\epsilon\delta)$ be the spatial step. Suppose the temporal step satisfies $\tau \le h\epsilon^2/(\epsilon + h)$. Then one has*

(1) The singular values of the coefficient matrix in (4.2) *satisfies*

$$\sigma_{\min} \gtrsim \frac{1}{N_t}, \quad \sigma_{\max} \lesssim N^{1/2}.$$

(2) The condition number $\kappa = \mathcal{O}(N^{1/2}N_t)$ and the sparsity $s = \mathcal{O}(N)$.

(3) The time complexities of the classical treatment and the quantum treatment for solving (4.2) *are*

$$C = \mathcal{O}\left(N^2 N_t N_x\right) = \mathcal{O}\left(N^2 \epsilon^{-3} \delta^{-2}\right),$$
$$Q = \mathcal{O}\left(N^{3/2} N_t \log N_x\right) = \mathcal{O}\left(N^{3/2} \epsilon^{-2} \delta^{-1} \log\left((\epsilon\delta)^{-1}\right)\right).$$

Proof. 1) The truncation error is $\mathcal{O}(\tau + h/\epsilon + ((2N)!)^{-1}/\epsilon^2)$. Let the error bound be δ. Then we can choose $h = \mathcal{O}(\epsilon\delta)$.

DOI https://doi.org/10.4208/eajam.2022-315.200423 | Generated on 2024-10-16 08:29:06

2) We first consider the minimum singular value. By definition, one has $\sigma_{\min}(\boldsymbol{L}) = 1/\|\boldsymbol{L}^{-1}\|$. It suffices to give an upper bound of $\|\boldsymbol{L}^{-1}\|$. By a direct calculation, one obtains

$$\boldsymbol{L}^{-1} = \begin{bmatrix} \boldsymbol{I} & & & & \\ \boldsymbol{B} & \boldsymbol{I} & & & \\ \boldsymbol{B}^2 & \ddots & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & \\ \boldsymbol{B}^{N_t-1} & \cdots & \boldsymbol{B}^2 & \boldsymbol{B} & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & & & \\ & \boldsymbol{I} & & \\ & & \ddots & \\ & & & \boldsymbol{I} \end{bmatrix} + \begin{bmatrix} \boldsymbol{O} & & & \\ \boldsymbol{B} & \boldsymbol{O} & & \\ & \ddots & \ddots & \\ & & \ddots & \ddots \\ & & & \boldsymbol{B} \quad \boldsymbol{O} \end{bmatrix} + \cdots,$$

hence,

$$\|\boldsymbol{L}^{-1}\| \le \|\boldsymbol{I}\| + \|\boldsymbol{B}\| + \cdots + \|\boldsymbol{B}^{N_t-1}\|.$$

Now we estimate $\|\boldsymbol{B}^n\|$. The matrix $\boldsymbol{B}$ can be split as $\boldsymbol{B} = \boldsymbol{B}_1 + \alpha \boldsymbol{B}_2$, where $\alpha = \tau/\epsilon^2$,

$$\boldsymbol{B}_1 = \begin{bmatrix} \boldsymbol{C} & -\frac{\lambda}{\epsilon}\boldsymbol{V}^- & & & \\ \frac{\lambda}{\epsilon}\boldsymbol{V}^+ & \boldsymbol{C} & -\frac{\lambda}{\epsilon}\boldsymbol{V}^- & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{\lambda}{\epsilon}\boldsymbol{V}^+ & \boldsymbol{C} & -\frac{\lambda}{\epsilon}\boldsymbol{V}^- \\ & & & \frac{\lambda}{\epsilon}\boldsymbol{V}^+ & \boldsymbol{C} \end{bmatrix}, \quad \boldsymbol{B}_2 = \frac{1}{2}\begin{bmatrix} \boldsymbol{W} & & & & \\ & \boldsymbol{W} & & & \\ & & \ddots & & \\ & & & \boldsymbol{W} & \\ & & & & \boldsymbol{W} \end{bmatrix}.$$

When $\tau \le h\epsilon^2/(\epsilon + h)$ or $\lambda \le \epsilon^2/(\epsilon + h)$, one easily finds that

$$c_k = 1 - \frac{\lambda}{\epsilon}\left(v_k^+ - v_k^-\right) - \frac{\tau}{\epsilon^2} \ge 0.$$

Since $v_k^+ \ge 0$ and $v_k^- \le 0$, using the Gershgorin-type lemma for singular values, one gets

$$\|\boldsymbol{B}_1\| \le \left(1 - \frac{\lambda}{\epsilon}\left(v_k^+ - v_k^-\right) - \frac{\tau}{\epsilon^2}\right) + \frac{\lambda}{\epsilon}v_k^+ - \frac{\lambda}{\epsilon}v_k^- = 1 - \frac{\tau}{\epsilon^2},$$

and $\|\boldsymbol{B}_1\| + \alpha \le 1$. Let $\mathbb{1} \in \mathbb{R}^{2N}$ be a column vector with all entries being 1. Then $\boldsymbol{W} = \mathbb{1}\boldsymbol{w}^T$, where $\boldsymbol{w} = [w_{-N}, \cdots, w_N]^T$. It is known that $\boldsymbol{W}$ is diagonalizable since $\boldsymbol{W}$ is a rank-one matrix. Let $\Lambda = X\boldsymbol{W}X^{-1}$ be the diagonal matrix. One can verify that X can be written as $X = YD_{\boldsymbol{w}}^{1/2}$, where $D_{\boldsymbol{w}} = \mathrm{diag}(\boldsymbol{w})$ and Y is an orthogonal matrix. In fact, noting that

$$D_{\boldsymbol{w}}^{1/2}\boldsymbol{W}D_{\boldsymbol{w}}^{-1/2} = D_{\boldsymbol{w}}^{1/2}\mathbb{1}\boldsymbol{w}^T D_{\boldsymbol{w}}^{-1/2} = D_{\boldsymbol{w}}^{1/2}\mathbb{1}\mathbb{1}^T D_{\boldsymbol{w}}^{1/2}$$

is a diagonal matrix and similar to $\boldsymbol{W}$, we can select an orthogonal matrix Y such that

$$D_{\boldsymbol{w}}^{1/2}\boldsymbol{W}D_{\boldsymbol{w}}^{-1/2} = Y^{-1}\Lambda Y, \quad Y = Y^T,$$

as required.

Let $D_X = \mathrm{diag}(X, \cdots, X)$ and $D_\Lambda = \mathrm{diag}(\Lambda, \cdots, \Lambda)$ and define $\tilde{\boldsymbol{B}}_1 = D_X\boldsymbol{B}_1D_X^{-1}$. Then, $D_X\boldsymbol{B}_2D_X^{-1} = D_\Lambda/2$ and

$$\begin{aligned} \|\boldsymbol{B}^n\| &= \left\|\left(D_X^{-1}\tilde{\boldsymbol{B}}_1D_X + \frac{\alpha}{2}D_X^{-1}D_\Lambda D_X\right)^n\right\| = \left\|D_X^{-1}\left(\tilde{\boldsymbol{B}}_1 + \frac{\alpha}{2}D_\Lambda\right)^n D_X\right\| \\ &\le \left\|D_X^{-1}\right\| \|D_X\| \left\|\left(\tilde{\boldsymbol{B}}_1 + \frac{\alpha}{2}D_\Lambda\right)^n\right\| \le \kappa(D_X)\left(\|\tilde{\boldsymbol{B}}_1\| + \frac{\alpha}{2}\|D_\Lambda\|\right)^n \\ &= \kappa(X)\left(\|\tilde{\boldsymbol{B}}_1\| + \frac{\alpha}{2}\|\Lambda\|\right)^n, \end{aligned}$$

where $\kappa(X)$ is the condition number of X. A direct calculation gives

$$\begin{aligned}
\tilde{\boldsymbol{B}}_1 &= D_X \boldsymbol{B}_1 D_X^{-1} \\
&= \begin{bmatrix} YD_{\boldsymbol{w}}^{1/2} & & & \\ & YD_{\boldsymbol{w}}^{1/2} & & \\ & & \ddots & \\ & & & YD_{\boldsymbol{w}}^{1/2} \end{bmatrix} \begin{bmatrix} C & -\frac{\lambda}{\epsilon}V^- & & \\ \frac{\lambda}{\epsilon}V^+ & \ddots & \ddots & \\ & \ddots & \ddots & -\frac{\lambda}{\epsilon}V^- \\ & & \frac{\lambda}{\epsilon}V^+ & C \end{bmatrix} \\
&\quad \times \begin{bmatrix} D_{\boldsymbol{w}}^{-1/2}Y^{-1} & & & \\ & D_{\boldsymbol{w}}^{-1/2}Y^{-1} & & \\ & & \ddots & \\ & & & D_{\boldsymbol{w}}^{-1/2}Y^{-1} \end{bmatrix} \\
&= \begin{bmatrix} Y & & & \\ & Y & & \\ & & \ddots & \\ & & & Y \end{bmatrix} \begin{bmatrix} C & -\frac{\lambda}{\epsilon}V^- & & \\ \frac{\lambda}{\epsilon}V^+ & \ddots & \ddots & \\ & \ddots & \ddots & -\frac{\lambda}{\epsilon}V^- \\ & & \frac{\lambda}{\epsilon}V^+ & C \end{bmatrix} \begin{bmatrix} Y^{-1} & & & \\ & Y^{-1} & & \\ & & \ddots & \\ & & & Y^{-1} \end{bmatrix} \\
&=: D_Y \boldsymbol{B}_1 D_Y^{-1},
\end{aligned}$$

where we have used the fact that

$$D_{\boldsymbol{w}}^{1/2} A D_{\boldsymbol{w}}^{-1/2} = A \quad \text{for} \quad A = C, V^{\pm}$$

since they are diagonal matrices. It is obvious that D_Y is an orthogonal matrix, which implies $\|\tilde{\boldsymbol{B}}_1\| = \|\boldsymbol{B}_1\|$. Therefore,

$$\begin{aligned}
\|\boldsymbol{B}^n\| &\le \kappa(X)\left(\|\boldsymbol{B}_1\| + \frac{\alpha}{2}\|\Lambda\|\right)^n \le \kappa(X)(\|\boldsymbol{B}_1\| + \alpha)^n \le \kappa(X) \\
&= \left\|YD_{\boldsymbol{w}}^{1/2}\right\| \left\|D_{\boldsymbol{w}}^{-1/2}Y^{-1}\right\| \le \sqrt{\frac{\max(w_k)}{\min(w_k)}} =: r_w.
\end{aligned}$$

This shows

$$\|\boldsymbol{L}^{-1}\| \lesssim r_w N_t \quad \text{or} \quad \sigma_{\min}(\boldsymbol{L}) \gtrsim \frac{1}{(r_w N_t)}.$$

3) For the maximum singular value, one obtains from the Gershgorin-type lemma that

$$\|\boldsymbol{L}\| \le 1 + \|\boldsymbol{B}\| \le 1 + \|\boldsymbol{B}_1\| + \alpha\|\boldsymbol{B}_2\| \le 2 + \frac{1}{2}\|\boldsymbol{W}\| \lesssim N^{1/2}.$$

4) It is obvious that the sparsity $s = \mathcal{O}(N)$. From the above estimates, we know that the condition number $\kappa = \mathcal{O}(N^{1/2}N_t)$.

5) The analysis of the time complexity can be carried out as that of the diffusive relaxation scheme. □

4.2. Discussion on the lower bound

It is an interesting–albeit not trivial-project to derive a lower bound not just for the explicit scheme, but for all other schemes. The optimal quantum algorithm for the QLSP in [8] is constructed by using the discrete adiabatic theorem or the discrete-time adiabatic evolution, which is based on a sequence of T walk operators. Let $U_T(s)$ be the discrete adiabatic evolution operator defined in [8] and let U_T^A be the ideal adiabatic evolution operator. To determine the complexity of the algorithm in terms of κ, the authors in [8] show that

$$\left\|U_T - U_T^A\right\| \leq \frac{C\kappa}{T},$$

which means that to obtain the solution to fixed error one can use $T = \mathcal{O}(\kappa)$ steps. If one can prove that

$$\left\|U_T - U_T^A\right\| \geq \frac{C\kappa}{T}$$

for our specific problem, then the run time may be at least $\mathcal{O}(\kappa)$. Such a lower bound may be difficult to establish even for the upper bound as observed in [8].

We remark that it may be relatively easy to give a lower bound for the Fourier approach in [6]. The argument is described as follows. The Fourier approach is based on an approximation of A^{-1} as a linear combination of unitaries $\mathrm{e}^{-\mathrm{i}At_i}$ (LCU for short), where A is for the QLSP: $A|x\rangle = |b\rangle$. According to [6, Lemma 12],

$$g(A) = \frac{\mathrm{i}}{\sqrt{2\pi}} \int_0^{y_J} \mathrm{d}y \int_{-z_K}^{z_K} \mathrm{d}z \; z\mathrm{e}^{-z^2/2}\mathrm{e}^{-\mathrm{i}Ayz}$$

is β-close to A^{-1} for some $y_J = \Theta(\kappa\sqrt{\log(\kappa/\beta)})$ and some $z_K = \Theta(\sqrt{\log(\kappa/\beta)})$, where β is the error bound and $f = \Theta(g)$ means $f = \mathcal{O}(g)$ and $g = \mathcal{O}(f)$. The integral is then approximated by the following finite sum:

$$h(A) = \frac{\mathrm{i}}{\sqrt{2\pi}} \sum_{j=0}^{J} \Delta y \sum_{k=-K}^{K} \Delta z \; z_k \mathrm{e}^{-z_k^2/2}\mathrm{e}^{-\mathrm{i}Ay_j z_k},$$

which is a linear combination of the unitaries $\mathrm{e}^{-\mathrm{i}At_i}$ for $t_i \in \mathbb{R}$.

According to [6, Corollary 10], the algorithm for applying the LCU makes $\mathcal{O}(\alpha)$ uses of the oracles for the input models, where α is the L^1 norm of the coefficients of the linear combination. For the given linear combination, it is proved that

$$\alpha = \Theta(y_J) = \Theta\left(\kappa\sqrt{\log(\kappa/\beta)}\right),$$

which implies the lower bound

$$Q_{\text{lower}} \gtrsim \kappa,$$

if we assume that the time complexity of the LCU algorithm is optimal.

It remains to give a lower bound in the form of $\kappa(\boldsymbol{L}) \geq C/\epsilon^{\alpha}$ for some $\alpha > 0$. For the lower bound of $\sigma_{\max}(\boldsymbol{L})$, noting that $\boldsymbol{L}^T\boldsymbol{L} = \text{tril}(-\boldsymbol{B}, \boldsymbol{I} + \boldsymbol{B}^T\boldsymbol{B}, -\boldsymbol{B}^T)$, we obtain from the Rayleigh representation theorem of eigenvalues of symmetric matrices that

$$\lambda_{\max}(\boldsymbol{L}^T\boldsymbol{L}) \geq \lambda_{\max}(\boldsymbol{I} + \boldsymbol{B}^T\boldsymbol{B}) \geq \max_i(\boldsymbol{I} + \boldsymbol{B}^T\boldsymbol{B})_{ii} = \mathcal{O}(N)$$

or

$$\sigma_{\max}(\boldsymbol{L}) \geq \mathcal{O}(N^{1/2}).$$

Next we discuss the upper bound of $\sigma_{\min}(\boldsymbol{L})$. From the Rayleigh representation theorem, one obtains

$$\|\boldsymbol{L}^{-1}\|^2 \geq \lambda_{\max}\left(\boldsymbol{I} + \boldsymbol{B}^T\boldsymbol{B} + (\boldsymbol{B}^2)^T\boldsymbol{B}^2 + \cdots + (\boldsymbol{B}^{N_t-1})^T\boldsymbol{B}^{N_t-1}\right).$$

Let $\lambda_{\max}(\boldsymbol{B})$ be the largest eigenvalue of $\boldsymbol{B}$ with the corresponding eigenvector denoted by ξ. Then,

$$\|\boldsymbol{L}^{-1}\|^2 \geq \frac{\xi^T(\boldsymbol{I} + \boldsymbol{B}^T\boldsymbol{B} + (\boldsymbol{B}^2)^T\boldsymbol{B}^2 + \cdots + (\boldsymbol{B}^{N_t-1})^T\boldsymbol{B}^{N_t-1})\xi}{\xi^T\xi} = \sum_{k=0}^{N_t-1} \lambda_{\max}^{2k}(\boldsymbol{B}).$$

We want to show that $\lambda_{\max}(\boldsymbol{B}) \geq 1$, which implies

$$\sigma_{\min}(\boldsymbol{L}) \leq 1/N_t^{1/2},$$

and hence

$$\kappa(\boldsymbol{L}) \geq (NN_t)^{1/2} \geq \frac{N^{1/2}}{\epsilon},$$

where the CFL condition in Theorem 4.1 has been used. However, we do not find a rigorous proof of $\lambda_{\max}(\boldsymbol{B}) \geq 1$ although it was numerically verified. For example, when choosing the Gaussian points and weights as

$$\begin{aligned} v = [&-0.9602898565, -0.7966664774, -0.5255324099, -0.1834346425,\\ &0.1834346425, 0.5255324099, 0.7966664774, 0.9602898565],\\ w = [&0.1012285363, 0.2223810345, 0.3137066459, 0.3626837834,\\ &0.3626837834, 0.3137066459, 0.2223810345, 0.1012285363], \end{aligned}$$

and set

$$h = \delta\epsilon, \quad \tau = \frac{h\epsilon^2}{h + \epsilon},$$

we obtain $\lambda_{\max}(\boldsymbol{B}) = 1$ for $\delta = 0.1$ and $\epsilon = 0.01$ for instance.

5. Conclusions

We studied the time complexities of finite difference methods for solving the multiscale transport equation in the setting of quantum computing. Our results show that the quantum implementation of the classical Asymptotic-Preserving schemes, a popular multiscale framework for multiscale problems [13], is equally important in quantum computing since they allow the computational costs for quantum algorithms to be independent of the small physical scaling parameters. This study also suggests that one should take full advantage of state-of-the-art multiscale classical algorithms when designing quantum algorithms for multiscale PDEs.

Acknowledgments

S. Jin was partially supported by the NSFC (Grant No. 12031013), by the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDA25010404), by the Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX0102) and the Innovation Program of Shanghai Municipal Education Commission (Grant No. 2021-01-07-00-02-E00087). Y. Yu was partially supported by the China Postdoctoral Science Foundation (Grant No. 2022M712080).

References

[1] A. Alase, R.R. Nerem, M. Bagherimehrab, P. Høyer and B.C. Sanders, *Tight bound for estimating expectation values from a system of linear equations*, arXiv:2111.10485 (2021).
[2] D.W. Berry, *High-order quantum algorithm for solving linear differential equations*, J. Phys. A: Math. Theor. **47(10)**, 105301, 17 pp. (2014).
[3] D.W. Berry, A.M. Childs, A. Ostrander and G. Wang, *Quantum algorithm for linear differential equations with exponentially improved dependence on precision*, Comm. Math. Phys. **356(3)**, 1057–1081 (2017).
[4] K.W. Case and P.F. Zweifel, *Linear Transport Theory*, Addison-Wesley (1967).
[5] S. Chandrasekhar, *Radiative Transfer*, Dover (1960).
[6] A.M. Childs, R. Kothari and R.D. Somma, *Quantum algorithm for systems of linear equations with exponentially improved dependence on precision*, SIAM J. Comput. **46(6)**, 1920–1950 (2017).
[7] A.M. Childs, J.P. Liu and A. Ostrander, *High-precision quantum algorithms for partial differential equations*, Quantum **5**, 574 (2021).
[8] P.C.S. Costa, D. An, Y.A. Sanders, Y. Su, R. Babbush and D.W. Berry, *Optimal scaling quantum linear systems solver via discrete adiabatic theorem*, arXiv:2111.08152 (2021).
[9] I.Y. Dobin and E.A. Startsev, *On applications of quantum computing to plasma simulations*, Phys. Plasmas **28**, 092101 (2021).
[10] A. Gilyén, Y. Su, G. Low and N. Wiebe, *Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics*, in: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC, 193–204 (2019).
[11] N. Gleinig and T. Hoefler, *An efficient algorithm for sparse quantum state preparation*, in: *2021 58th ACM/IEEE Design Automation Conference (DAC)*, 433–438 (2021).

[12] A.W. Harrow, A. Hassidim and S. Lloyd, *Quantum algorithm for linear systems of equations*, Phys. Rev. Lett. **103(15)**, 150502, 4 pp. (2009).
[13] S. Jin, *Asymptotic-preserving schemes for multiscale physical problems*, Acta Numer. **31**, 415–489 (2022).
[14] S. Jin and N. Liu, *Quantum algorithms for computing observables of nonlinear partial differential equations*, arXiv:2202.07834 (2022).
[15] S. Jin, N. Liu and Y. Yu, *Time complexity analysis of quantum algorithms via linear representations for nonlinear ordinary and partial differential equations*, arXiv:2209.08478 (2022).
[16] S. Jin, N. Liu and Y. Yu, *Time complexity analysis of quantum difference methods for linear high dimensional and multiscale partial differential equations*, J. Comput. Phys. **471**, 111641 (2022).
[17] S. Jin, L. Pareschi and G. Toscani, *Diffusive relaxation schemes for multiscale discrete-velocity kinetic equations*, SIAM J. Numer. Anal. **35(6)**, 2405–2439 (1998).
[18] S. Jin, L. Pareschi and G. Toscani, *Uniformly accurate diffusive relaxation schemes for multiscale transport equations*, SIAM J. Numer. Anal. **38**, 913–936 (2000).
[19] I. Joseph, *Koopman–von Neumann approach to quantum simulation of nonlinear classical dynamics*, Phys. Rev. Research **2(4)**, 043102 (2020).
[20] L. Lin, *Lecture notes on quantum algorithms for scientific computation*, arXiv:2201.08309 (2022).
[21] N. Linden, A. Montanaro and C. Shao, *Quantum vs. classical algorithms for solving the heat equation*, arXiv:2004.06516 (2020).
[22] J. Liu, H.Ø. Kolden, H.K. Krovi, N.F. Loureiro, K. Trivisa and A.M. Childs, *Efficient quantum algorithm for dissipative nonlinear differential equations*, Proc. Natl. Acad. Sci. U. S. A. **118(35)** (2021).
[23] S. Lloyd, G.D. Palma, C. Gokler, B. Kiani, Z. Liu, M. Marvian, F. Tennie and T. Palmer, *Quantum algorithm for nonlinear differential equations*, arXiv:2011.06571 (2020).
[24] G.H. Low, *Hamiltonian simulation with nearly optimal dependence on spectral norm*, in: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC, 491–502 (2019).
[25] P.A. Markowich, C.A. Ringhofer and C. Schmeiser, *Semiconductor Equations*, Springer (1990).
[26] L. Ryzhik, G. Papanicolaou and J.B. Keller, *Transport equations for elastic and other waves in random media*, Wave Motion **24(4)**, 327–370 (1996).
[27] G.W. Stewart, *Perturbation theory for the singular value decomposition*, In: *SVD and Signal Processing, II. Algorithms, Analysis and Applications*, 99–109, Elsevier (1990).
[28] X. Zhang, T. Li and X. Yuan, *Quantum state preparation with optimal circuit depth: Implementations and applications*, arXiv:2201.11495 (2022).