East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.2022-304.070323

Vol. **xx**, No. x, pp. 1-32
xxx 2023

A Diagonalization-Based Parallel-in-Time Algorithm for Crank-Nicolson's Discretization of the Viscoelastic Equation

Fu Li and Yingxiang Xu*

School of Mathematics and Statistics, Northeast Normal University, Changchun 130024, P.R. China.

Received 2 November 2022; Accepted (in revised version) 7 March 2023.

Abstract. In this paper, we extend a diagonalization-based parallel-in-time (PinT) algorithm to the viscoelastic equation. The central difference method is used for spatial discretization, while for temporal discretization, we use the Crank-Nicolson scheme. Then an all-at-once system collecting all the solutions at each time level is formed and solved using a fixed point iteration preconditioned by an α-circulant matrix in parallel. Via a rigorous analysis, we find that the spectral radius of the iteration matrix is uniformly bounded by $\alpha/(1-\alpha)$, independent of the model parameters (the damping coefficient ε and the wave velocity $\sqrt{\gamma}$) and the discretization parameters (the time step τ and the spatial mesh size h). Unlike the classical wave equation with Dirichlet boundary condition where the upper bound $\alpha/(1-\alpha)$ is very sharp, we find that the occurrence of the damping term $-\varepsilon\Delta y_t$, as well as the large final time T, leads to even faster convergence of the algorithm, especially when α is not very small. We illustrate our theoretical findings with several numerical examples.

AMS subject classifications: 65F08, 65F10, 65M22, 65Y05

Key words: Parallel-in-time (PinT) algorithm, Crank-Nicolson method, diagonalization, viscoelastic equation, convergence analysis.

1. Introduction

In this paper, we consider the following viscoelastic equation:

$$\begin{aligned} &y_{tt}-\varepsilon\Delta y_t-\gamma\Delta y=f, && (\mathbf{x},t)\in\Omega\times J,\\ &y(\mathbf{x},t)=\phi(\mathbf{x},t), && (\mathbf{x},t)\in\partial\Omega\times J,\\ &y(\mathbf{x},0)=\psi_1(\mathbf{x}), && \mathbf{x}\in\Omega,\\ &y_t(\mathbf{x},0)=\psi_2(\mathbf{x}), && \mathbf{x}\in\Omega, \end{aligned} \tag{1.1}$$

*Corresponding author. *Email addresses:* lif097@nenu.edu.cn (F. Li), yxxu@nenu.edu.cn (Y. Xu)

where $\Omega \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ is an open domain, $J = (0, T]$ is the time interval, $\varepsilon \geq 0$ is the damping coefficient and $\sqrt{\gamma}$ is the wave velocity with $\gamma > 0$. The source function f, the boundary function $\phi(\mathbf{x}, t)$ and the initial functions $\psi_1(\mathbf{x})$ and $\psi_2(\mathbf{x})$ are all given. Without loss of generality, we assume that a homogeneous boundary condition is applied, i.e. $\phi(\mathbf{x}, t) = 0$, to simplify the theoretical analysis.

Compare to the classical wave equation, the viscoelastic equation uses a damping term $-\varepsilon \Delta y_t$ to provide a more accurate model in many applications, for example, the propagation of vibration waves through viscoelastic media [26]. We refer the reader to [50] for more applications in science and engineering. Recently, Gander *et al.* [14] numerically validated that the viscoelastic damping $-\varepsilon \Delta y_t$ works much better than the first order damping term $-y_t$ (which results in a telegrapher's equation) for modeling the vibration of an elastic string.

The well-posedness of the viscoelastic equation had been addressed a long time ago [2, 39, 42]. However, like most PDEs in applications, it is very hard to find the analytical solutions due to the complicated given data, or, the complex defining domain. As a result, it is very essential to study the numerical methods for solving the viscoelastic equation. In fact, all prevalent techniques for spatial discretization can be applied to the viscoelastic equation. See, for example, [9, 45] for finite element methods, [3, 17, 25, 38] for mixed finite element methods, [20, 54] for finite difference methods, [22, 23] for generalized finite difference methods (also known as finite volume element methods), [43] for discontinuous Galerkin methods and [49] for weak Galerkin finite element methods. Recently, the meshless methods also attract much attention in solving the viscoelastic equation (1.1), see for instance [35, 37].

However, most numerical methods for solving Eq. (1.1) are based on time stepping. For example, the reader can refer to [30, 53] for a Crank-Nicolson scheme, where an extrapolation approach and a proper orthogonal decomposition technique are used to reduce the degree of freedom. To speedup the computation, one can apply parallel computing technique at each time level due to the fact that the time discretization leads to a steady partial differential equation. Amongst the various parallel computing methods, the domain decomposition method attracts the most attention. The domain decomposition method originated from Schwarz's seminal work [41] in 1870, and was developed in 1990 by Lions [28] as a parallel solver, which finally leads to the optimized Schwarz methods, cf. [10] and references cited therein. Other efficient domain decomposition methods include the restricted additive Schwarz (RAS) method [47], the finite element tearing and interconnecting (FETI) method [8], the balancing domain decomposition by constraints (BDDC) [6], etc. We refer the reader to the monographs [7, 47] for many other variants of the aforementioned methods in detail. An alternative to the domain decomposition method would be the multigrid method, cf. for example [48]. Obviously, these methods can be straightforwardly applied to the viscoelastic equation (1.1) after time discretization.

In order to use the computing resources more efficiently, one can apply the parallel-in-time (PinT) computation to further accelerate the solving of the viscoelastic equation (1.1). In fact, Adey and Brebbia [1] proposed long ago to solve the viscoelastic equation in parallel in the Laplace transformed plane by the finite element method, and then a least

square collocation method is applied to construct the whole solution in the time domain. However, except for the above-mentioned work, we do not find any published articles concerning the parallel computing of viscoelastic equations. While for other PDEs arising in science and engineering, the parallel algorithm has been extensively studied. In fact, the PinT approach can be traced back at least to 1964 [34], and we refer the reader to [11] for a historical review. Among the numerous PinT algorithms, the parareal method proposed by Lions *et al.* [27] attracts the most attention. The parareal method is a prediction-correction method with prediction generated by a coarse grid propagator and correction via a fine grid propagator. It can be interpreted as a two-level multigrid method [15] and leads to a good convergence for parabolic problems. However, the parareal method does not converge well for hyperbolic problems, though many efforts had been laid on improving the algorithm [33,40]. Concerning the hyperbolic problems, the PinT algorithms based on diagonalization perform quite well. This method was proposed by Maday and Rønquist in [31] as a direct PinT solver. The main idea is to collect the unknowns at all time levels as an all-at-once system and to diagonalize the time stepping matrix while keeping the spatial discretization matrices unchanged, which allows solving all time steps at once in parallel. To ensure that the time discretization matrix is diagonalizable, Gander *et al.* proposed in [12] using variable non-uniform time steps. However, when the number of time points is large, the algorithm is difficult to balance the discretization error and the rounding error. Very recently, by borrowing the idea of preconditioning Toeplitz matrices with a circulant one from Strang [44] and Olkin [36], Liu and Wu [29] proposed an iterative PinT algorithm based on diagonalization for solving the wave equation, where the all-at-once system of block Toeplitz structure is preconditioned with a block α-circulant matrix [4]. The analysis for an implicit leap-frog time stepping and a central difference scheme for spatial discretization shows that the non-unit eigenvalues of the preconditioned matrix $\mathscr{P}_\alpha^{-1}\mathscr{K}$ all belong to the annulus

$$\mathbb{A}_\alpha := \left\{ z \in \mathbb{C} : \frac{\alpha}{1+\alpha} \le |z-1| \le \frac{\alpha}{1-\alpha} \right\}.$$

In addition, this result also holds when the Numerov scheme is applied for the time discretization, together with the compact finite difference in space, see [46]. Particularly, Wu *et al*. [52] proved under a more general framework that this property holds if the time stepping scheme is stable under several mild assumptions, which, however, do not meet by our algorithm. For a case where the non-unit eigenvalues of the preconditioned matrix $\mathscr{P}_\alpha^{-1}\mathscr{K}$ are not bounded from below, we refer the reader to [51] for the two-stage singly diagonally implicit Runge-Kutta (SDIRK) method. For many other variants of diagonalization-based PinT algorithms, we refer the reader to [13].

In this paper, we would like to solve the viscoelastic equation (1.1) using the aforementioned diagonalization-based PinT algorithm. We use the Crank-Nicolson method for temporal discretization since it is unconditionally stable, and choose the central difference scheme as spatial discretization due to its second-order accuracy. Through rewriting the total unknowns at all time levels as an all-at-once system $\mathscr{K} y_h = b_h$, we solve it with a block α-circulant preconditioner $\mathscr{P}_\alpha$ that could be inversed by a diagonalization technique in parallel. The spectral analysis shows that, because of the occurrence of the damping term in

the viscoelastic equation, the spectral distribution of the preconditioned matrix $\mathscr{P}_\alpha^{-1}\mathscr{K}$ is somehow different from those for wave equations. More precisely, like for the wave equation, the non-zero eigenvalues of the preconditioned iteration matrix $\mathscr{M} = \mathscr{I} - \mathscr{P}_\alpha^{-1}\mathscr{K}$ are still bounded from above by a uniform bound of $\alpha/(1-\alpha)$, independent of the damping coefficient ε. However, unlike the wave equation, they are not bounded from below. Furthermore, the upper bound $\alpha/(1-\alpha)$ is not sharp unless $\varepsilon = 0$, that is to say, compared to the wave equation, the damping term leads to even faster convergence of the preconditioned iteration (2.4).

The rest of this paper is organized as follows. In Section 2, we briefly introduce the discretization scheme for the viscoelastic equation (1.1) and the corresponding all-at-once system, as well as the diagonalization-based PinT algorithm for solving it. In Section 3, we analyze in detail the spectral distribution of the preconditioned matrix $\mathscr{P}_\alpha^{-1}\mathscr{K}$ and derive a uniform upper bound. Especially, we lay our efforts on explaining how the damping term affects the convergence behavior of the iterative method (2.4) in Section 4. We then in Section 5 conduct several numerical examples to illustrate our theoretical findings and finally draw conclusions in Section 6.

2. Diagonalization-Based PinT Algorithm

2.1. Discretization using Crank-Nicolson in time and central difference in space

For given integers N_{x_i}, $i = 1, \ldots, d$ and N_t, we denote by $h_i = 1/(N_{x_i}+1)$ and $\tau = T/N_t$ the spatial and temporal mesh sizes, respectively. We partition the time interval $[0, T]$ uniformly by the time points $\{t_n = n\tau\}_{n=0}^{N_t}$. Let Δ_h be the discrete Laplacian obtained by using central difference, which gives a second-order approximation to the Laplacian Δ with boundary conditions described in (1.1). More exactly, $\Delta_h = (1/h_1^2)L$ for the 1D case, $\Delta_h = (1/h_1^2)L \otimes I_{x_2} + I_{x_1} \otimes (1/h_2^2)L$ for the 2D case, and $\Delta_h = (1/h_1^2)L \otimes I_{x_2} \otimes I_{x_3} + I_{x_1} \otimes (1/h_2^2)L \otimes I_{x_3} + I_{x_1} \otimes I_{x_2} \otimes (1/h_3^2)L$ for the 3D case, with $(1/h_i^2)L$ being the 1D discrete Laplacian. Then, discretizing (1.1) temporally using Crank-Nicolson and spatially using central difference, we arrive at the fully discrete scheme

$$\begin{aligned} &\frac{Y_{n+1} - 2Y_n + Y_{n-1}}{\tau^2} - \varepsilon\Delta_h \frac{Y_{n+1} - Y_{n-1}}{2\tau} - \gamma\Delta_h \frac{Y_{n+1} + 2Y_n + Y_{n-1}}{4} \\ &= \frac{F_{n+1} + 2F_n + F_{n-1}}{4}, \end{aligned} \tag{2.1}$$

where $n = 1, 2, \ldots, N_t - 1$, and Y_n is a lexicographic ordered vector collecting the approximate solutions of $y(\cdot, t_n)$ over all the space grids. F_n, Ψ_0 and Ψ_1 are similarly defined for f, ψ_0 and ψ_1, respectively. The initial value Y_0 is directly given by the third equality in (1.1). However, Y_1 has to be determined by an approximation considering the discretization accuracy. For example, the Taylor expansion

$$y(\cdot, \tau) = y(\cdot, 0) + \tau y_t(\cdot, 0) + \frac{\tau^2}{2} y_{tt}(\cdot, 0) + \mathscr{O}(\tau^3)$$

suggests the following second-order approximation:

$$Y_1 = \Psi_0 + \tau\Psi_1 + \frac{\tau^2}{2}(F_0 + \varepsilon\Delta_h\Psi_1 + \gamma\Delta_h\Psi_0),$$

since

$$y_{tt}(\cdot,0) = f(\cdot,0) + \varepsilon\Delta_h y_t(\cdot,0) + \gamma\Delta_h y(\cdot,0)$$

according to (1.1).

We comment here that our analysis also works for many other discretization techniques. For example, applying the implicit leap-frog method investigated in [29] leads to

$$\frac{Y_{n+1}-2Y_n+Y_{n-1}}{\tau^2} - \varepsilon\Delta_h\frac{Y_{n+1}-Y_{n-1}}{2\tau} - \gamma\Delta_h\frac{Y_{n+1}+Y_{n-1}}{2} = F_n. \tag{2.2}$$

Obviously, the scheme (2.2) approximates the term $-\gamma\Delta y$ differently, which leads to a different coefficient matrix in the all-at-once system (2.3), and still could be analyzed by our methods. In addition, the different definition of the right-hand side of (2.2) does not affect the analysis.

2.2. An all-at-once system

Let $I_t \in \mathbb{R}^{N_t\times N_t}$ and I_x ($I_x \in \mathbb{R}^{N_x\times N_x}$ in 1D, $I_x \in \mathbb{R}^{N_{x_1}N_{x_2}\times N_{x_1}N_{x_2}}$ in 2D, and $I_x \in \mathbb{R}^{N_{x_1}N_{x_2}N_{x_3}\times N_{x_1}N_{x_2}N_{x_3}}$ in 3D) be the identity matrices. Denote $D = (\varepsilon\tau/2)\Delta_h, E = (\gamma\tau^2/4)\Delta_h$. With these notations, we rewrite scheme (2.1) as the following all-at-once system:

$$\mathscr{K}y_h := \frac{1}{\tau^2}(B_1\otimes I_x - B_2\otimes D - B_3\otimes E)y_h = b_h, \tag{2.3}$$

where

$$B_1 = \begin{bmatrix} 1 & & & & \\ -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 & & & & \\ 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \end{bmatrix},$$

$$B_3 = \begin{bmatrix} 1 & & & & \\ 2 & 1 & & & \\ 1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 2 & 1 \end{bmatrix}, \quad y_h = \begin{bmatrix} Y_2 \\ Y_3 \\ \vdots \\ Y_{N_t-1} \\ Y_{N_t} \end{bmatrix}, \quad b_h = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{N_t-1} \end{bmatrix},$$

and

$$\begin{aligned} b_1 &= \frac{1}{4}(F_0 + 2F_1 + F_2) + \frac{2}{\tau^2}(I_x + E)Y_1 - \frac{1}{\tau^2}(I_x + D - E)Y_0, \\ b_2 &= \frac{1}{4}(F_1 + 2F_2 + F_3) - \frac{1}{\tau^2}(I_x + D - E)Y_1, \\ b_i &= \frac{1}{4}(F_{i-1} + 2F_i + F_{i+1}), \quad i = 3,4,\ldots,N_t-1. \end{aligned}$$

Unlike to the all-at-once system for wave equations [29,46], we get an extra tensor term in the coefficient matrix $\mathscr{K}$ due to the occurrence of the damping term $-\varepsilon\Delta y_t$, which brings difficulties to the convergence analysis in Section 3.

2.3. Fixed point iteration preconditioned by a block α-circulant matrix

The system (2.3) is of large scale and it is better to solve it with an iterative method. In a generic form, the fixed point iteration for solving (2.3) can be written as

$$y_h^{k+1} = y_h^k + \mathscr{P}_\alpha^{-1} r^k \tag{2.4}$$

with the residual $r^k = b_h - \mathscr{K} y_h^k$ and an initial guess y_h^0, where $\mathscr{P}_\alpha$ is a preconditioner, which should be determined such that the spectral radius of the iteration matrix $\mathscr{M} = \mathscr{I} - \mathscr{P}_\alpha^{-1}\mathscr{K}$ is as small as possible. In this paper, based on the idea of [29,46], we choose the preconditioner $\mathscr{P}_\alpha$ as the following block α-circulant matrix:

$$\mathscr{P}_\alpha = \frac{1}{\tau^2}\Big(C_1^{(\alpha)} \otimes I_x - C_2^{(\alpha)} \otimes D - C_3^{(\alpha)} \otimes E\Big),$$

where

$$C_1^{(\alpha)} = \begin{bmatrix} 1 & & & \alpha & -2\alpha \\ -2 & 1 & & & \alpha \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{bmatrix}, \quad C_2^{(\alpha)} = \begin{bmatrix} 1 & & & -\alpha & 0 \\ 0 & 1 & & & -\alpha \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \end{bmatrix},$$

$$C_3^{(\alpha)} = \begin{bmatrix} 1 & & & \alpha & 2\alpha \\ 2 & 1 & & & \alpha \\ 1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 2 & 1 \end{bmatrix}$$

are Strang-type circulant matrices obtained by modifying the Toeplitz matrices B_1, B_2 and B_3, and $\alpha \in (0, 1/2)$ is a free parameter. This idea of preconditioning a Toeplitz matrix with a circulant matrix can be traced back to [36,44] and was applied to the all-at-once system generated by the discretization of evolutionary PDE in [18,32].

Now we address how to solve (2.4) in a PinT pattern. Actually, the proposed preconditioner $\mathscr{P}_\alpha$ can be inverted — i.e. $\mathscr{P}_\alpha^{-1} r$ can be efficiently computed, in a PinT pattern due to the fact that C_1^α, C_2^α and C_3^α are simultaneously diagonalizable. Consider discrete Fourier matrix

$$\mathbb{F} = \frac{1}{\sqrt{N_t - 1}}\Big[\omega^{(l_1-1)(l_2-1)}\Big]_{l_1,l_2=1}^{N_t-1}, \quad \omega = e^{\frac{2\pi i}{N_t-1}}, \quad i = \sqrt{-1}$$

and the diagonal matrix

$$\Gamma_\alpha = \mathrm{diag}\Big(1, \alpha^{\frac{1}{N_t-1}}, \ldots, \alpha^{\frac{N_t-2}{N_t-1}}\Big).$$

Then the α-circulant matrices $C^{\alpha}_{1,2,3}$ can be simultaneously diagonalized as

$$C^{(\alpha)}_j = VD_jV^{-1}, \quad j = 1,2,3,$$

where $V = \Gamma_\alpha^{-1}\mathbb{F}^*$ and $D_{1,2,3} = \mathrm{diag}(\sqrt{N_t-1}\mathbb{F}\Gamma_\alpha C^{(\alpha)}_{1,2,3}(:,1))$ with $C^{(\alpha)}_{1,2,3}(:,1)$ being the first column of $C^{(\alpha)}_{1,2,3}$. Obviously, it holds that

$$\mathscr{P}_\alpha = \frac{1}{\tau^2}(V\otimes I_x)(D_1\otimes I_x - D_2\otimes D - D_3\otimes E)(V^{-1}\otimes I_x). \tag{2.5}$$

Thus, for any input vector r, the inversion $z := \mathscr{P}_\alpha^{-1}r$ can be implemented via the following steps:

$$\begin{aligned}
&\text{Step (a).} \quad S_1 = (V^{-1}\otimes I_x)r.\\
&\text{Step (b).} \quad S_{2,n} = \tau^2(\lambda_{1,n}I_x - \lambda_{2,n}D - \lambda_{3,n}E)^{-1}S_{1,n},\ n = 1,2,\ldots,N_t-1.\\
&\text{Step (c).} \quad z = (V\otimes I_x)S_2,
\end{aligned} \tag{2.6}$$

where $\lambda_{j,n}$ is the diagonal entries of D_j with the following explicit expressions:

$$\begin{aligned}
\lambda_{1,n} &= 1 - 2\alpha^{\frac{1}{N_t-1}}e^{\frac{2(n-1)\pi i}{N_t-1}} + \alpha^{\frac{2}{N_t-1}}e^{\frac{4(n-1)\pi i}{N_t-1}},\\
\lambda_{2,n} &= 1 - \alpha^{\frac{2}{N_t-1}}e^{\frac{4(n-1)\pi i}{N_t-1}},\\
\lambda_{3,n} &= 1 + 2\alpha^{\frac{1}{N_t-1}}e^{\frac{2(n-1)\pi i}{N_t-1}} + \alpha^{\frac{2}{N_t-1}}e^{\frac{4(n-1)\pi i}{N_t-1}},
\end{aligned}$$

and $S_{j,n} = S_j((n-1)N_x : nN_x)$ denotes the n-th block of S_j for $j = 1,2$.

Note that steps (a) and (c) can be computed efficiently via FFT. We then consider step (b). Direct calculations show that $\lambda_{1,n} \neq 0$ for $\alpha \in (0,1/2)$. As a result, one has to solve the complex-shifted Laplacian systems

$$\left(-\Delta_h + \frac{4\lambda_{1,n}}{2\varepsilon\lambda_{2,n}\tau + \gamma\lambda_{3,n}\tau^2}I_x\right)S_{2,n} = \frac{4\tau^2}{2\varepsilon\lambda_{2,n}\tau + \gamma\lambda_{3,n}\tau^2}S_{1,n}, \quad n = 1,2,\ldots,N_t-1.$$

Actually, efficient numerical methods for such Helmholtz-like complex systems have been extensively studied, see for example [5].

3. Spectral Analysis of $\mathscr{P}_\alpha^{-1}\mathscr{K}$

To describe the convergence of the scheme (2.4), it is sufficient to analyze the spectral distribution of the preconditioned matrix $\mathscr{P}_\alpha^{-1}\mathscr{K}$. We consider only the one-dimensional case, where $D = (\varepsilon\tau/2h^2)L$ and $E = (\gamma\tau^2/4h^2)L$. Two- and three-dimensional situations can be considered analogously. Denote

$$Q_1 = I_x - D - E, \quad Q_2 = I_x + E, \quad Q_3 = I_x + D - E, \quad M_i = \mathbf{e}_i\otimes I_x,$$

where $\mathbf{e}_i$ is the i-th column of the identity matrix $I_t \in \mathbb{R}^{(N_t-1)\times(N_t-1)}$. According to (2.3) and (2.5), $\mathscr{K}$ and $\mathscr{P}_\alpha$ have the following representations in terms of Q_1, Q_2 and Q_3:

$$\mathscr{K} = \frac{1}{\tau^2}\begin{bmatrix} Q_1 & & & & \\ -2Q_2 & Q_1 & & & \\ Q_3 & -2Q_2 & Q_1 & & \\ & \ddots & \ddots & \ddots & \\ & & Q_3 & -2Q_2 & Q_1 \end{bmatrix},$$

$$\mathscr{P}_\alpha = \frac{1}{\tau^2}\begin{bmatrix} Q_1 & & & \alpha Q_3 & -2\alpha Q_2 \\ -2Q_2 & Q_1 & & 0 & \alpha Q_3 \\ Q_3 & -2Q_2 & Q_1 & & \\ & \ddots & \ddots & \ddots & \\ & & Q_3 & -2Q_2 & Q_1 \end{bmatrix}.$$

Obviously, $\mathscr{P}_\alpha$ is a low-rank perturbation of $\mathscr{K}$,

$$\begin{aligned} \alpha\mathscr{R} := \mathscr{P}_\alpha - \mathscr{K} &= \frac{\alpha}{\tau^2}\begin{bmatrix} 0 & \cdots & 0 & Q_3 & -2Q_2 \\ & 0 & \cdots & 0 & Q_3 \\ & & 0 & \cdots & 0 \\ & & & \ddots & \vdots \\ & & & & 0 \end{bmatrix} \\ &= \frac{1}{\tau^2}\begin{bmatrix} M_1 & M_2 \end{bmatrix}\begin{bmatrix} \alpha Q_3 & -2\alpha Q_2 \\ 0 & \alpha Q_3 \end{bmatrix}\begin{bmatrix} M_{N_t-2} & M_{N_t-1} \end{bmatrix}^T \\ &=: \frac{1}{\tau^2} UGV^T. \end{aligned}$$

Lemma 3.1. *The preconditioned matrix $\mathscr{P}_\alpha^{-1}\mathscr{K}$ can be written as $\mathscr{P}_\alpha^{-1}\mathscr{K} = \mathscr{I} - \mathscr{K}^{-1}UZ^{-1}V^T$ with $\mathscr{I} = I_t \otimes I_x$, $Z = \tau^2 G^{-1} + V^T\mathscr{K}^{-1}U$. Furthermore, $\mathscr{P}_\alpha^{-1}\mathscr{K}$ has $(N_t-3)N_x$ eigenvalues equal to 1 and the other $2N_x$ eigenvalues are the eigenvalues of the matrix $(I_{2x}+Y)^{-1}$, where $Y = (1/\tau^2)V^T\mathscr{K}^{-1}UG$.*

Proof. The first assertion is obtained by using the well-known Sherman-Morrison-Woodbury formula [19], viz.

$$\begin{aligned} \mathscr{P}_\alpha^{-1}\mathscr{K} &= (\mathscr{K} + \alpha\mathscr{R})^{-1}\mathscr{K} \\ &= \left(\mathscr{K} + \frac{1}{\tau^2}UGV^T\right)^{-1}\mathscr{K} \\ &= \mathscr{I} - \mathscr{K}^{-1}U(\tau^2 G^{-1} + V^T\mathscr{K}^{-1}U)^{-1}V^T \\ &= \mathscr{I} - \mathscr{K}^{-1}UZ^{-1}V^T. \end{aligned}$$

Noting the fact that $\mathscr{K}$ is a block Toeplitz matrix of triangular form, we know that $\mathscr{K}^{-1}$ has the same structure and can be written as

$$\mathscr{K}^{-1}=\begin{bmatrix} K_0^{-1} & & & & \\ K_1^{-1} & K_0^{-1} & & & \\ K_2^{-1} & K_1^{-1} & K_0^{-1} & & \\ \vdots & \ddots & \ddots & \ddots & \\ K_{N_t-2}^{-1} & \cdots & K_2^{-1} & K_1^{-1} & K_0^{-1} \end{bmatrix}.$$

Using this notation, a careful calculation gives that $\mathscr{P}_\alpha^{-1}\mathscr{K}$ has $(N_t-3)N_x$ eigenvalues equal to 1 and the other $2N_x$ eigenvalues are determined by its last block $I_{2x}-V^T\mathscr{K}^{-1}UZ^{-1}$. Further calculation shows

$$\begin{aligned} I_{2x}-V^T\mathscr{K}^{-1}UZ^{-1} &= \tau^2G^{-1}Z^{-1} \\ &= \tau^2G^{-1}\left(G^{-1}+V^T\mathscr{K}^{-1}U\right)^{-1} \\ &= \left((G^{-1}+V^T\mathscr{K}^{-1}U)(\tau^2G^{-1})^{-1}\right)^{-1} \\ &= \left(I_{2x}+\frac{1}{\tau^2}V^T\mathscr{K}^{-1}UG\right)^{-1}. \end{aligned}$$

This ends the proof. □

Lemma 3.1 shows that we need to analyze the eigenvalues of Y. To this end, we reformulate the matrix $\mathscr{K}$ as

$$\mathscr{K}=\frac{1}{\tau^2}(B_1\otimes I_x-B_2\otimes D-B_3\otimes E)=\frac{1}{\tau^2}(I_t\otimes Q_1)\mathscr{J}, \tag{3.1}$$

where

$$\mathscr{J}=\begin{bmatrix} I_x & & & & \\ -2Q_1^{-1}Q_2 & I_x & & & \\ Q_1^{-1}Q_3 & -2Q_1^{-1}Q_2 & I_x & & \\ & \ddots & \ddots & \ddots & \\ & & Q_1^{-1}Q_3 & -2Q_1^{-1}Q_2 & I_x \end{bmatrix}.$$

The block triangular Toeplitz matrix $\mathscr{J}$ is invertible and its inverse has the form

$$\mathscr{J}^{-1}=\begin{bmatrix} J_0^{-1} & & & \\ J_1^{-1} & J_0^{-1} & & \\ \vdots & \ddots & \ddots & \\ J_{N_t-2}^{-1} & \cdots & J_1^{-1} & J_0^{-1} \end{bmatrix},$$

where J_n^{-1} represents the n-th diagonal block of $\mathscr{J}^{-1}$. Besides, it is easily checked that $\{J_n^{-1}\}_{n=1}^{N_t-2}$ satisfies the following recurrence relation:

$$J_{n+1}^{-1}=2Q_1^{-1}Q_2J_n^{-1}-Q_1^{-1}Q_3J_{n-1}^{-1},\quad n=1,2,\ldots,N_t-2 \tag{3.2}$$

with $J_0^{-1}=I_x$ and $J_1^{-1}=2Q_1^{-1}Q_2$. Solving (3.2), we get the solution

$$J_n^{-1}=\Theta_1r_1^n+\Theta_2r_2^n,\quad n=0,1,2,\ldots,N_t-1,$$

where

$$r_1 = Q_1^{-1}Q_2 - \left((Q_1^{-1}Q_2)^2 - Q_1^{-1}Q_3\right)^{\frac{1}{2}},$$
$$r_2 = Q_1^{-1}Q_2 + \left((Q_1^{-1}Q_2)^2 - Q_1^{-1}Q_3\right)^{\frac{1}{2}},$$

and

$$\Theta_1 = \frac{1}{2}\left(I_x - Q_1^{-1}Q_2\left((Q_1^{-1}Q_2)^2 - Q_1^{-1}Q_3\right)^{-\frac{1}{2}}\right),$$
$$\Theta_2 = \frac{1}{2}\left(I_x + Q_1^{-1}Q_2\left((Q_1^{-1}Q_2)^2 - Q_1^{-1}Q_3\right)^{-\frac{1}{2}}\right).$$

Using (3.1) we get $\mathscr{K}^{-1} = \tau^2\mathscr{J}^{-1}(I_t \otimes Q_1^{-1})$, which implies that the n-th block diagonal of $\mathscr{K}^{-1}$ satisfies $K_n^{-1} = \tau^2 J_n^{-1}Q_1^{-1}$. This equality implies

$$\begin{aligned} Y &= \frac{1}{\tau^2}V^T\mathscr{K}^{-1}UG \\ &= \frac{1}{\tau^2}\begin{bmatrix} K_{N_t-3}^{-1} & K_{N_t-4}^{-1} \\ K_{N_t-2}^{-1} & K_{N_t-3}^{-1} \end{bmatrix}\begin{bmatrix} \alpha Q_3 & -2\alpha Q_2 \\ 0 & \alpha Q_3 \end{bmatrix} \\ &= \alpha\begin{bmatrix} J_{N_t-3}^{-1}Q_1^{-1}Q_3 & -J_{N_t-2}^{-1} \\ J_{N_t-2}^{-1}Q_1^{-1}Q_3 & -J_{N_t-1}^{-1} \end{bmatrix} \\ &=: \alpha X. \end{aligned}$$

Now it is sufficient to analyze the spectral distribution of the matrix X. We comment here that, if there is no damping, say $\varepsilon = 0$, then $Q_1 = Q_3$. Hence, the expression of X is quite simple and the eigenvalues of X can be easily determined — cf. [46]. In particular, we use a diagonalization technique. From the definitions of Q_1, Q_2 and Q_3, one finds that there exists an orthogonal matrix R diagonalizing $Q_i, i = 1,2,3$, simultaneously

$$Q_1 = RG_1R^{-1}, \quad Q_2 = RG_2R^{-1}, \quad Q_3 = RG_3R^{-1},$$

where $G_i, i = 1,2,3$, are diagonal matrices. Hence,

$$Q_1^{-1}Q_2 = RG_1^{-1}G_2R^{-1} =: R\Lambda_1R^{-1},$$
$$Q_1^{-1}Q_3 = RG_1^{-1}G_3R^{-1} =: R\Lambda_2R^{-1}.$$

Using R, we diagonalize the matrices J_n^{-1} for $n = N_t-1, N_t-2, N_t-3$, thus obtaining

$$\begin{aligned} \Sigma_n := R^{-1}J_n^{-1}R = &\frac{1}{2}\left(I_x - \Lambda_1\left(\Lambda_1^2 - \Lambda_2\right)^{-\frac{1}{2}}\right)\left(\Lambda_1 - \left(\Lambda_1^2 - \Lambda_2\right)^{\frac{1}{2}}\right)^n \\ &+ \frac{1}{2}\left(I_x + \Lambda_1\left(\Lambda_1^2 - \Lambda_2\right)^{-\frac{1}{2}}\right)\left(\Lambda_1 + \left(\Lambda_1^2 - \Lambda_2\right)^{\frac{1}{2}}\right)^n. \end{aligned} \tag{3.3}$$

For simplicity, we denote by $\Lambda_1 = \mathrm{diag}(\zeta_1, \ldots, \zeta_{N_x}), \Lambda_2 = \mathrm{diag}(\rho_1, \ldots, \rho_{N_x})$. Then the eigenvalues of J_n^{-1} for $n = N_t-1, N_t-2, N_t-3$ can be determined from the Eq. (3.3) as

$$\mu_{n,j} = \frac{1}{2}\left(1 - \zeta_j\left(\zeta_j^2 - \rho_j\right)^{-\frac{1}{2}}\right)\left(\zeta_j - \left(\zeta_j^2 - \rho_j\right)^{\frac{1}{2}}\right)^n$$

$$+\frac{1}{2}\left(1+\zeta_j\left(\zeta_j^2-\rho_j\right)^{-\frac{1}{2}}\right)\left(\zeta_j+\left(\zeta_j^2-\rho_j\right)^{\frac{1}{2}}\right)^n, \quad j=1,2,\ldots,N_x. \tag{3.4}$$

Now, we are ready to calculate the eigenvalues of X. The following similarity transformation:

$$\begin{aligned}\begin{bmatrix} R^{-1} & 0 \\ 0 & R^{-1}\end{bmatrix} X \begin{bmatrix} R & 0 \\ 0 & R\end{bmatrix} &= \begin{bmatrix} R^{-1} & 0 \\ 0 & R^{-1}\end{bmatrix}\begin{bmatrix} J_{N_t-3}^{-1}Q_1^{-1}Q_3 & -J_{N_t-2}^{-1} \\ J_{N_t-2}^{-1}Q_1^{-1}Q_3 & -J_{N_t-1}^{-1}\end{bmatrix}\begin{bmatrix} R & 0 \\ 0 & R\end{bmatrix} \\ &= \begin{bmatrix} \Sigma_{N_t-3}\Lambda_2 & -\Sigma_{N_t-2} \\ \Sigma_{N_t-2}\Lambda_2 & -\Sigma_{N_t-1}\end{bmatrix}\end{aligned} \tag{3.5}$$

shows that we only have to calculate the eigenvalues of the right-hand side of (3.5). Noting that each block entry is a diagonal matrix, one can then find a simple permutation matrix P that transforms this matrix into a block diagonal one

$$P^{-1}\begin{bmatrix} \Sigma_{N_t-3}\Lambda_2 & -\Sigma_{N_t-2} \\ \Sigma_{N_t-2}\Lambda_2 & -\Sigma_{N_t-1}\end{bmatrix} P = \mathrm{blkdiag}(\Phi_1,\Phi_2,\ldots,\Phi_{N_x}),$$

where

$$\Phi_j = \begin{bmatrix} \rho_j\mu_{N_t-3,j} & -\mu_{N_t-2,j} \\ \rho_j\mu_{N_t-2,j} & -\mu_{N_t-1,j}\end{bmatrix}, \quad j=1,2,\ldots,N_x$$

are 2×2 matrices. Obviously, the two eigenvalues of Φ_j are

$$\nu_j^{\pm} = \frac{1}{2}\left(\rho_j\mu_{N_t-3,j}-\mu_{N_t-1,j} \pm \sqrt{(\rho_j\mu_{N_t-3,j}+\mu_{N_t-1,j})^2-4\rho_j\mu_{N_t-2,j}^2}\right). \tag{3.6}$$

Along with Lemma 3.1, this leads to the following result.

Theorem 3.1. *The spectrum of $\mathscr{P}_\alpha^{-1}\mathscr{K}$ consists of $(N_t-3)N_x$ eigenvalues 1 and $2N_x$ eigenvalues located in the set*

$$\hat{\sigma}\left(\mathscr{P}_\alpha^{-1}\mathscr{K}\right) := \left\{\frac{1}{1+\alpha\nu_j^{\pm}},\ j=1,2,\ldots,N_x\right\},$$

i.e.

$$\sigma\left(\mathscr{P}_\alpha^{-1}\mathscr{K}\right) = \underbrace{\{1,1,\ldots,1\}}_{(N_t-3)N_x} \cup \hat{\sigma}\left(\mathscr{P}_\alpha^{-1}\mathscr{K}\right).$$

In order to better understand the spectrum of $\mathscr{P}^{-1}\mathscr{K}$, we to further analyze Eq. (3.6). From the recursion formula (3.2) one finds $\mu_{n,j} = 2\zeta_j\mu_{n-1,j} - \rho_j\mu_{n-2,j}$, which simpli-

fies (3.6) to

$$\begin{aligned}
\nu_j^{\pm} &= \frac{1}{2}\Big(\rho_j\mu_{N_t-3,j} - \mu_{N_t-1,j} \pm \sqrt{(\rho_j\mu_{N_t-3,j} + \mu_{N_t-1,j})^2 - 4\rho_j\mu_{N_t-2,j}^2}\Big) \\
&= \frac{1}{2}\Big(\rho_j\mu_{N_t-3,j} - (2\zeta_j\mu_{N_t-2,j} - \rho_j\mu_{N_t-3,j}) \\
&\qquad \pm \sqrt{(\rho_j\mu_{N_t-3,j} + (2\zeta_j\mu_{N_t-2,j} - \rho_j\mu_{N_t-3,j}))^2 - 4\rho_j\mu_{N_t-2,j}^2}\Big) \\
&= (\rho_j\mu_{N_t-3,j} - \zeta_j\mu_{N_t-2,j}) \pm \frac{1}{2}\sqrt{4(\zeta_j\mu_{N_t-2,j})^2 - 4\rho_j\mu_{N_t-2,j}^2} \\
&= (\rho_j\mu_{N_t-3,j} - \zeta_j\mu_{N_t-2,j}) \pm \mu_{N_t-2,j}\sqrt{\zeta_j^2 - \rho_j}. \qquad (3.7)
\end{aligned}$$

It follows from (3.7) and (3.4) that in order to better understand the spectral distribution of $\mathscr{P}_\alpha^{-1}\mathscr{K}$, one has to study the values of $\zeta_j^2 - \rho_j$, i.e. the eigenvalues of $(Q_1^{-1}Q_2)^2 - Q_1^{-1}Q_3$.

Before going further, we make the following assumption.

Assumption 3.1. Any eigenvalue $\lambda_j(L)$ of L belongs to the interval $(-4, 0]$.

We comment here that this assumption is quite reasonable, see for example [46].

Assuming that Assumption 3.1 holds, we investigate the eigenvalues of $Q_1^{-1}Q_2$ and $Q_1^{-1}Q_3$.

Proposition 3.1. *For any given $\varepsilon, \gamma \geq 0, \tau, h > 0$, we have $\lambda_j(Q_1^{-1}Q_2) \in (-1, 1]$ and $\lambda_j(Q_1^{-1}Q_3) \in (-1, 1]$.*

Proof. Expressing Q_1, Q_2 and Q_3 as

$$\begin{aligned}
Q_1 &= I_x - D - E = I_x - \frac{2\varepsilon\tau + \gamma\tau^2}{4h^2}L, \\
Q_2 &= I_x + E = I_x + \frac{\gamma\tau^2}{4h^2}L, \\
Q_3 &= I_x + D - E = I_x + \frac{2\varepsilon\tau - \gamma\tau^2}{4h^2}L,
\end{aligned}$$

we find

$$\begin{aligned}
\zeta_j = \lambda_j\big(Q_1^{-1}Q_2\big) &= \lambda_j\left(\left(I_x - \frac{2\varepsilon\tau + \gamma\tau^2}{4h^2}L\right)^{-1}\left(I_x + \frac{\gamma\tau^2}{4h^2}L\right)\right) \\
&= \frac{1 + \gamma\tau^2/(4h^2)\lambda_j(L)}{1 - (2\varepsilon\tau + \gamma\tau^2)/(4h^2)\lambda_j(L)}, \qquad (3.8) \\
\rho_j = \lambda_j\big(Q_1^{-1}Q_3\big) &= \lambda_j\left(\left(I_x - \frac{2\varepsilon\tau + \gamma\tau^2}{4h^2}L\right)^{-1}\left(I_x + \frac{2\varepsilon\tau - \gamma\tau^2}{4h^2}L\right)\right) \\
&= \frac{1 + (2\varepsilon\tau - \gamma\tau^2)/(4h^2)\lambda_j(L)}{1 - (2\varepsilon\tau + \gamma\tau^2)/(4h^2)\lambda_j(L)}. \qquad (3.9)
\end{aligned}$$

Since $\lambda_j(L)$ is non-positive according to Assumption 3.1, one easily finds

$$\lambda_j\left(Q_1^{-1}Q_2\right) \le 1, \quad \lambda_j\left(Q_1^{-1}Q_3\right) \le 1.$$

We show the lower bounds of $\lambda_j(Q_1^{-1}Q_2)$ and $\lambda_j(Q_1^{-1}Q_3)$ by contradiction. If $\lambda_j(Q_1^{-1}Q_2) \le -1$, then

$$1 + \frac{(\varepsilon\tau + \gamma\tau^2)/(2h^2)\lambda_j(L)}{1-(2\varepsilon\tau + \gamma\tau^2)/(4h^2)\lambda_j(L)} \le -1,$$

i.e.

$$2 \le \frac{\varepsilon\tau}{2h^2}\lambda_j(L),$$

which contradicts the non-positivity of $\lambda_j(L)$. The estimate $\lambda_j(Q_1^{-1}Q_3) > -1$ is obtained analogously. □

Remark 3.1. In fact, the j-th eigenvalues of $Q_1^{-1}Q_2$ and $Q_1^{-1}Q_3$ satisfy the following relation:

$$\lambda_j\left(Q_1^{-1}Q_2\right) = \lambda_j\left(Q_1^{-1}Q_3\right) + \frac{(\gamma\tau^2 - \varepsilon\tau)/(2h^2)\lambda_j(L)}{1-(2\varepsilon\tau + \gamma\tau^2)/(4h^2)\lambda_j(L)}.$$

Checking the sign of $\gamma\tau^2 - \varepsilon\tau$, one easily obtains that if $\tau > \varepsilon/\gamma$, then $\lambda_j(Q_1^{-1}Q_3) > \lambda_j(Q_1^{-1}Q_2)$. The inequality still holds if we replace '>' by '<'. While the equality holds if $\tau = \varepsilon/\gamma$, and this is exactly the case dealt with by [46], though no damping ($\varepsilon = 0$) is considered in (1.1) there.

Using (3.8) and (3.9) we find

$$\lambda_j\left(\left(Q_1^{-1}Q_2\right)^2 - Q_1^{-1}Q_3\right) = \frac{\lambda_j(L)(\gamma\tau^2/h^2 + (\varepsilon\tau/(2h^2))^2\lambda_j(L))}{(1-((2\varepsilon\tau + \gamma\tau^2)/(4h^2))\lambda_j(L))^2}.$$

For ease of discussion on the $\nu_j^{\pm}$ shown in (3.7), we consider the following three situations:

$$\text{Case I.} \quad \lambda_j(L) > -\frac{4h^2\gamma}{\varepsilon^2}.$$

$$\text{Case II.} \quad \lambda_j(L) = -\frac{4h^2\gamma}{\varepsilon^2}.$$

$$\text{Case III.} \quad \lambda_j(L) < -\frac{4h^2\gamma}{\varepsilon^2}.$$

The condition in Case I implies $\lambda_j((Q_1^{-1}Q_2)^2 - Q_1^{-1}Q_3) < 0$, i.e. $\zeta_j^2 - \rho_j < 0$. Let $a_j = \sqrt{\rho_j - \zeta_j^2}$. We have

$$\mu_{n,j} = \frac{1}{2}\left(1 + \frac{i\zeta_j}{a_j}\right)\left(\zeta_j - ia_j\right)^n + \frac{1}{2}\left(1 - \frac{i\zeta_j}{a_j}\right)\left(\zeta_j + ia_j\right)^n. \tag{3.10}$$

Hence, $\nu_j^{\pm}$ reads

$$\begin{aligned}\nu_j^{\pm} &= (\rho_j \mu_{N_t-3,j} - \zeta_j \mu_{N_t-2,j}) \pm \mu_{N_t-2,j}\sqrt{\zeta_j^2 - \rho_j} \\ &= (\rho_j \mu_{N_t-3,j} - \zeta_j \mu_{N_t-2,j}) \pm i a_j \mu_{N_t-2,j}. \end{aligned} \tag{3.11}$$

Using (3.10), one finds

$$\begin{aligned}\rho_j \mu_{N_t-3,j} &= \frac{1}{2}\rho_j\left(1+\frac{i\zeta_j}{a_j}\right)\left(\zeta_j - ia_j\right)^{N_t-3} + \frac{1}{2}\rho_j\left(1-\frac{i\zeta_j}{a_j}\right)\left(\zeta_j + ia_j\right)^{N_t-3}, \\ \zeta_j \mu_{N_t-2,j} &= \frac{1}{2}\zeta_j\left(1+\frac{i\zeta_j}{a_j}\right)\left(\zeta_j - ia_j\right)^{N_t-2} + \frac{1}{2}\zeta_j\left(1-\frac{i\zeta_j}{a_j}\right)\left(\zeta_j + ia_j\right)^{N_t-2} \\ &= \frac{1}{2}\zeta_j\left(1+\frac{i\zeta_j}{a_j}\right)\left(\zeta_j - ia_j\right)\left(\zeta_j - ia_j\right)^{N_t-3} \\ &\quad + \frac{1}{2}\zeta_j\left(1-\frac{i\zeta_j}{a_j}\right)\left(\zeta_j + ia_j\right)\left(\zeta_j + ia_j\right)^{N_t-3}, \\ ia_j \mu_{N_t-2,j} &= \frac{1}{2}ia_j\left(1+\frac{i\zeta_j}{a_j}\right)\left(\zeta_j - ia_j\right)^{N_t-2} + \frac{1}{2}ia_j\left(1-\frac{i\zeta_j}{a_j}\right)\left(\zeta_j + ia_j\right)^{N_t-2} \\ &= \frac{1}{2}ia_j\left(1+\frac{i\zeta_j}{a_j}\right)\left(\zeta_j - ia_j\right)\left(\zeta_j - ia_j\right)^{N_t-3} \\ &\quad + \frac{1}{2}ia_j\left(1-\frac{i\zeta_j}{a_j}\right)\left(\zeta_j + ia_j\right)\left(\zeta_j + ia_j\right)^{N_t-3} \\ &= \frac{(a_j+i\zeta_j)^2}{2}\left(\zeta_j - ia_j\right)^{N_t-3} - \frac{(a_j+i\zeta_j)^2}{2}\left(\zeta_j + ia_j\right)^{N_t-3}. \end{aligned}$$

Substituting these representations into (3.11) gives

$$\nu_j^{\pm} = \left(a_j \pm i\zeta_j\right)^2\left(\zeta_j \mp ia\right)^{N_t-3} = \left(\sqrt{\rho_j - \zeta_j^2} \pm i\zeta_j\right)^2\left(\zeta_j \mp i\sqrt{\rho_j - \zeta_j^2}\right)^{N_t-3}.$$

Thus, it holds

$$\left|\nu_j^{\pm}\right| = \left|\sqrt{\rho_j - \zeta_j^2} \pm i\zeta_j\right|^2 \left|\zeta_j \mp i\sqrt{\rho_j - \zeta_j^2}\right|^{N_t-3} = \left|\rho_j\right|^{N_t-1} \le 1. \tag{3.12}$$

Using (3.12) one finds for $\alpha \in (0, 1/2)$ that

$$\left|\frac{1}{1+\alpha\nu_j^{\pm}} - 1\right| = \frac{\alpha|\nu_j^{\pm}|}{|1+\alpha\nu_j^{\pm}|} \le \frac{\alpha}{1-\alpha}. \tag{3.13}$$

That is to say, in Case I those eigenvalues of $\mathscr{P}_\alpha^{-1}\mathscr{K}$ described by $1/(1+\alpha\nu_j^{\pm})$ (see Theorem 3.1) are contained in a disk centered at $(1,0)$ with radius $\alpha/(1-\alpha)$

$$\hat{\sigma}\left(\mathscr{P}_\alpha^{-1}\mathscr{K}\right) \subset \left\{z \in \mathbb{C} : |z-1| \le \frac{\alpha}{1-\alpha}\right\}, \tag{3.14}$$

where the equality holds if and only if $\rho_j = 1$ for some j, which is equivalent to $\max_j \lambda_j(L) = 0$, this happens, for example, when periodic or Neumann boundary condition is imposed. Note that for a well-posed problem one has in general $\lambda_j(L) \neq 0$.

Case II means $\rho_j - \zeta_j^2 = 0$, which reduces (3.4) to $\mu_{n,j} = (\zeta_j)^n$. Inserting it into (3.7), we arrive for Case II at

$$v_j^{\pm} = \rho_j \mu_{N_t-3,j} - \zeta_j \mu_{N_t-2,j} = \rho_j \zeta_j^{N_t-3} - \zeta_j \zeta_j^{N_t-2} = \zeta_j^{N_t-3}\left(\rho_j - \zeta_j^2\right) = 0.$$

We then get $\lambda_j(\mathscr{P}_\alpha^{-1}\mathscr{K}) = 1$. That is to say, (3.14) holds as well for Case II.

Case III is equivalent to $\zeta_j^2 - \rho_j > 0$. Denoting by $a_j = \sqrt{\zeta_j^2 - \rho_j}$, one finds

$$\mu_{n,j} = \frac{1}{2}\left(1 - \frac{\zeta_j}{a_j}\right)\left(\zeta_j - a_j\right)^n + \frac{1}{2}\left(1 + \frac{\zeta_j}{a_j}\right)\left(\zeta_j + a_j\right)^n. \tag{3.15}$$

And $v_j^{\pm}$ simplifies to

$$\begin{aligned} v_j^{\pm} &= (\rho_j \mu_{N_t-3,j} - \zeta_j \mu_{N_t-2,j}) \pm \mu_{N_t-2,j}\sqrt{\zeta_j^2 - \rho_j} \\ &= (\rho_j \mu_{N_t-3,j} - \zeta_j \mu_{N_t-2,j}) \pm a_j \mu_{N_t-2,j}. \end{aligned} \tag{3.16}$$

Using (3.15), we have

$$\begin{aligned} \rho_j \mu_{N_t-3,j} &= \frac{1}{2}\rho_j\left(1 - \frac{\zeta_j}{a_j}\right)\left(\zeta_j - a_j\right)^{N_t-3} + \frac{1}{2}\rho_j\left(1 + \frac{\zeta_j}{a_j}\right)\left(\zeta_j + a_j\right)^{N_t-3}, \\ \zeta_j \mu_{N_t-2,j} &= \frac{1}{2}\zeta_j\left[1 - \frac{\zeta_j}{a_j}\right]\left[\zeta_j - a_j\right]^{N_t-2} + \frac{1}{2}\zeta_j\left[1 + \frac{\zeta_j}{a_j}\right]\left[\zeta_j + a_j\right]^{N_t-2} \\ &= \frac{-2\zeta_j^3 + \zeta_j\rho_j + 2a_j\zeta_j^2}{2a_j}\left(\zeta_j - a_j\right)^{N_t-3} + \frac{2\zeta_j^3 - \zeta_j\rho_j + 2a_j\zeta_j^2}{2a_j}\left(\zeta_j + a_j\right)^{N_t-3}, \\ a_j \mu_{N_t-2,j} &= \frac{1}{2}a_j\left(1 - \frac{\zeta_j}{a_j}\right)\left(\zeta_j - a_j\right)^{N_t-2} + \frac{1}{2}a_j\left(1 + \frac{\zeta_j}{a_j}\right)\left(\zeta_j + a_j\right)^{N_t-2} \\ &= \frac{-2\zeta_j^2 + 2a_j\zeta_j + \rho_j}{2}\left(\zeta_j - a_j\right)^{N_t-3} + \frac{2\zeta_j^2 + 2a_j\zeta_j - \rho_j}{2}\left(\zeta_j + a_j\right)^{N_t-3}. \end{aligned}$$

Substituting these identities into (3.16), we get using (3.15) and the definition of a_j

$$\begin{aligned} v_j^{\pm} &= \left(\frac{1}{2}\rho_j\left(1 - \frac{\zeta_j}{a_j}\right) - \frac{-2\zeta_j^3 + \zeta_j\rho_j + 2a_j\zeta_j^2}{2a}\right)\left(\zeta_j - a_j\right)^{N_t-3} \\ &\quad + \left(\frac{1}{2}\rho_j\left(1 + \frac{\zeta_j}{a_j}\right) - \frac{2\zeta_j^3 - \zeta_j\rho_j + 2a_j\zeta_j^2}{2a_j}\right)\left(\zeta_j + a_j\right)^{N_t-3} \pm i a_j \mu_{N_t-2,j} \\ &= \frac{2\zeta_j^3 - 2a_j\zeta_j^2 - 2\zeta_j\rho_j + \rho_j a_j}{2a_j}\left(\zeta_j - a_j\right)^{N_t-3} \end{aligned}$$

$$+\frac{-2\zeta_j^3-2a_j\zeta_j^2+2\zeta_j\rho_j+\rho_j a_j}{2a_j}\left(\zeta_j+a_j\right)^{N_t-3}\pm ia_j\mu_{N_t-2,j}$$
$$=-(\zeta_j\mp a_j)^{N_t-1}.$$

Obviously, to understand the range of $|\nu_j^{\pm}|$, we need to estimate the magnitude of $|\zeta_j\mp\sqrt{\zeta_j^2-\rho_j}|$. From the expressions of ζ_j and ρ_j, we get

$$1+\zeta_j=\frac{2-(\varepsilon\tau/(2h^2))\lambda_j(L)}{1-((2\varepsilon\tau+\gamma\tau^2)/(4h^2))\lambda_j(L)}>\frac{-2-(\varepsilon\tau/(2h^2))\lambda_j(L)}{1-((2\varepsilon\tau+\gamma\tau^2)/(4h^2))\lambda_j(L)}=-(\zeta_j+\rho_j),$$

which is equivalent to $1+2\zeta_j+\zeta_j^2>\zeta_j^2-\rho_j$, i.e. $\zeta_j-\sqrt{\zeta_j^2-\rho_j}>-1$. In addition, by the definition of ζ_j and the assumption of Case III, we have $\sqrt{\zeta_j^2-\rho_j}>0\geq\zeta_j-1$, i.e. $\zeta_j-\sqrt{\zeta_j^2-\rho_j}<1$. We therefore have $|\zeta_j-\sqrt{\zeta_j^2-\rho_j}|<1$, i.e. $|\nu_j^+|<1$. On the other hand, from the expressions of ζ_j and ρ_j, we get

$$1-\zeta_j=\frac{-((\varepsilon\tau+\gamma\tau^2)/(2h^2))\lambda_j(L)}{1-((2\varepsilon\tau+\gamma\tau^2)/(4h^2))\lambda_j(L)}>\frac{((\varepsilon\tau-\gamma\tau^2)/(2h^2))\lambda_j(L)}{1-((2\varepsilon\tau+\gamma\tau^2)/(4h^2))\lambda_j(L)}=\zeta_j-\rho_j.$$

Hence, we obtain $\sqrt{\zeta_j^2-\rho_j}<1-\zeta_j$, i.e. $1>\zeta_j+\sqrt{\zeta_j^2-\rho_j}$. Similarly, in this case we have $\sqrt{\zeta_j^2-\rho_j}>0>-1-\zeta_j$. Hence, $|\zeta_j+\sqrt{\zeta_j^2-\rho_j}|<1$ holds — i.e. $|\nu_j^-|<1$. Using the result $|\nu_j^{\pm}|<1$, we find that (3.13) holds.

Integrating considerations in Cases I-III, we have the following theorem.

Theorem 3.2. *The spectral radius of the preconditioned iteration matrix $\mathscr{M}=\mathscr{I}-\mathscr{P}_\alpha^{-1}\mathscr{K}$ is bounded from above by $\alpha/(1-\alpha)$ for $0<\alpha<1$, i.e.*

$$\sigma(\mathscr{M})\leq\frac{\alpha}{1-\alpha},$$

where the equality holds if and only if $\max_j\lambda_j(L)=0$.

Remark 3.2. We comment here that the lower bound $\alpha/(1+\alpha)$ of the non-zero eigenvalues of $\mathscr{M}$ for the classical wave equation does not hold for the viscoelastic equation (1.1) because of the occurrence of the damping term $-\varepsilon\Delta y_t$. How this term affects the convergence behavior of the fixed point iteration (2.4) will be detailedly addressed in the next section.

Remark 3.3. In both two- and three-dimensional situations, Cases I-III lead to the same results as illustrated in Theorem 3.2, including the spectral distribution of $\mathscr{P}_\alpha^{-1}\mathscr{K}$. Hence, one would expect the convergence behavior of the preconditioned fixed point iteration (2.4) similar to the 1D case.

4. Further Discussion

4.1. Influence of parameter α

Theorem 3.2 provides a uniform upper bound for the spectral radius of the iteration matrix $\mathscr{M}$, which does not depend on the mesh sizes τ and h, the damping coefficient ε and the wave velocity parameter γ and depends only on α

$$\rho(\mathscr{M}) = \max_{\lambda\in\sigma(\mathscr{P}_\alpha^{-1}\mathscr{K})} |\lambda - 1| \le \frac{\alpha}{1-\alpha}, \tag{4.1}$$

which is smaller than one for any $\alpha \in (0, 1/2)$. Hence, one can determine how many iterations would be required at most by the scheme (2.4) for reaching a prescribed tolerance. Actually, let $r^k = \mathscr{P}_\alpha^{-1}(b_h - \mathscr{K} y_h^k)$ denote the preconditioned residual vector at the k-th iteration. Then, we have

$$\|r^{k+1}\| = \|\mathscr{M} r^k\| = \|\mathscr{M}^k r^0\| \le \|\mathscr{M}\|^k \|r^0\|. \tag{4.2}$$

Using (4.1) and (4.2), one finds that the relative residual $\|r^k\|/\|r^0\| < tol$ requires for $\alpha \in (0, 1/2)$

$$\left(\frac{\alpha}{1-\alpha}\right)^k \le tol,$$

which gives

$$k = k(tol;\alpha) := \left\lceil \frac{\ln(tol)}{\ln\alpha - \ln(1-\alpha)} \right\rceil. \tag{4.3}$$

Here $\lceil s \rceil$ represents the smallest integer greater than or equal to s. Indeed, the fixed point iteration (2.4) intends to reach the tolerance tol with even fewer iterations, because of the occurrence of the damping term, and/or the large final time T. This will be detailed in the next subsection.

In addition, we find from (4.1) that the smaller the parameter α is, the faster the fixed point iteration (2.4) converges. However, choosing an optimal α in practice is not as easy as stated above. In fact, for a scheme mathematically equivalent to (2.4), the parameter α should not be arbitrarily small, and has to be chosen by considering the roundoff error, see Section 5.

4.2. Effect of the damping term

The damping term makes the viscoelastic equation advantageous in many applications over the classical wave equation. However, with the uniform upper bound $\alpha/(1-\alpha)$ one cannot show how the damping term affects the convergence of the fixed point iteration (2.4), which will be investigated in this subsection. To this end, we still consider the three cases mentioned in the previous section.

In Case I, we have

$$\rho(\mathscr{M}) = \max_j \left| \frac{1}{1+\alpha\nu_j^{\pm}} - 1 \right| \le \max_j \frac{\alpha|\nu_j^{\pm}|}{1-\alpha},$$

which is $\alpha/(1-\alpha)$ if and only if $|\nu_j^{\pm}| = 1$ for some j, that is to say, $\lambda_j(L) = 0$, and this happens only for the Neumann or the periodic boundary conditions. Otherwise, we have $\lambda_j(L) < 0$ and the function ρ_j defined in (3.9) is a decreasing function of ε with

$$\lim_{\varepsilon\to 0}\rho_j = 1, \quad \lim_{\varepsilon\to+\infty}\rho_j = -1.$$

As a result, one finds that $|\nu_j^{\pm}| = |\rho_j|^{N_t-1}$, when regarding as function of ε, decreases first and then increases to 1.

In Case II, the eigenvalues of $\mathscr{M}$ are all exactly zeros. In Case III, we have that if $\lambda_j(L) < 0$, then

$$\begin{aligned}\left|\nu_j^{\pm}\right| &= \left|\zeta_j \mp \sqrt{\zeta_j^2-\rho_j}\right|^{N_t-1} \\ &= \left|\frac{1+(\gamma\tau^2/(4h^2))\lambda_j(L) \mp \sqrt{(\gamma\tau^2/h^2)\lambda_j(L)+(\varepsilon\tau/(2h^2))^2\lambda_j^2(L)}}{1-((2\varepsilon\tau+\gamma\tau^2)/(4h^2))\lambda_j(L)}\right|^{N_t-1}.\end{aligned}$$

Similar to Case I, one can find that $|\nu_j^{\pm}|$ decreases first from 1 and then increases to 1 in ε.

Combining Cases I-III, we find when $\lambda_j(L) < 0$ for all j that $|\nu_j^{\pm}|$ decreases from 1 first in ε and then increases to 1. As a consequence, also noting that $\rho(\mathscr{M}) \le (\alpha/(1-\alpha))|\nu_j|$, we conclude that the occurrence of the damping term will lead to faster convergence of the fixed point iteration (2.4). Especially, there will be some values of ε such that where the fixed point iteration (2.4) converges the fastest. However, for the Neumann or the periodic boundary conditions, the performance of the fixed point iteration (2.4) would not be affected by the damping coefficient ε.

In Fig. 1, we show the spectral radius $\rho(\mathscr{M})$ as function of ε, compared with the upper bound $\alpha/(1-\alpha)$, where, without loss of generality, we set $\gamma = 1$. Note that a Dirichlet boundary condition is applied, and thus $\lambda_j(L) < 0$ for all j. We find that around $\varepsilon = 0.5$ the spectral radius $\rho(\mathscr{M})$ attains its minimum, which is about $1/7$ the upper bound $\alpha/(1-\alpha)$ and could lead to a significant acceleration when α is not very small. To see the dependence of the spectral distribution on the damping coefficient ε more clearly, we show in Fig. 2 the eigenvalues of the preconditioned matrix $\mathscr{P}_\alpha^{-1}\mathscr{K}$ for different values of ε. We find that for the relatively large ε the eigenvalues are real. With the decreasing of ε, the eigenvalues cluster more tightly and the complex eigenvalues are born. Finally, the eigenvalues scatter and recover the spectral distribution for the case of the classical wave equation gradually, where the non-unit eigenvalues have a lower bound $\alpha/(1+\alpha)$ [46]. However, the spectral radius $\rho(\mathscr{M})$ is exactly $\alpha/(1-\alpha)$ when $\lambda_j(L) = 0$ for some j. This phenomenon is well illustrated by Fig. 3.

4.3. Speedup analysis

Concerning the speedup analysis for the three-step implementation (2.6), one can refer to [46], where the analysis does not depend on the discrete matrix problem at each time

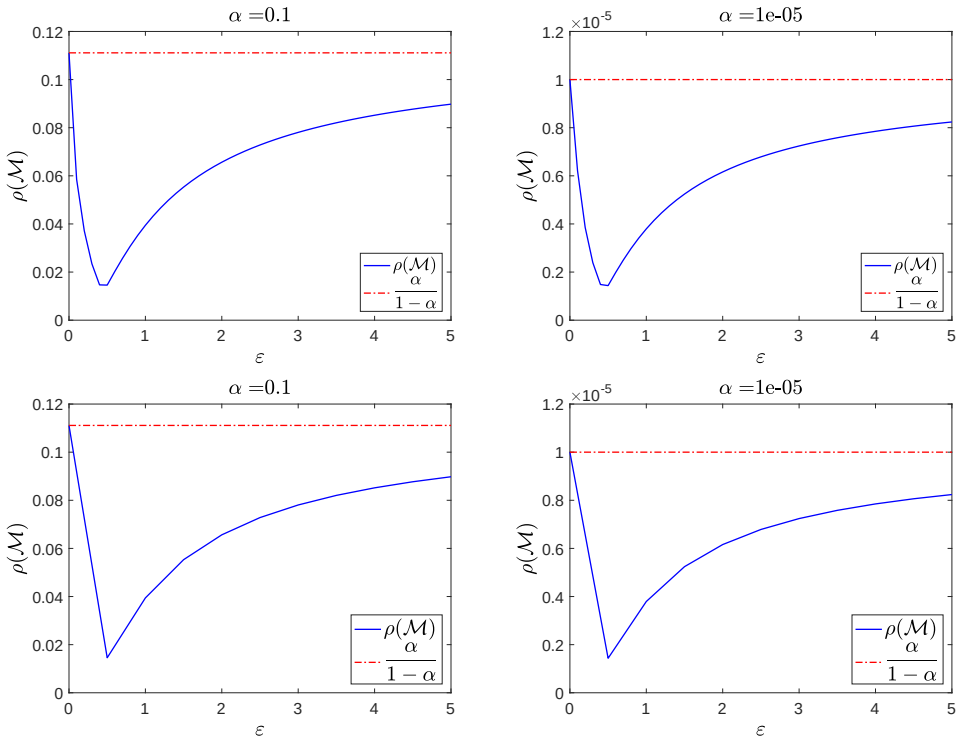

Figure 1: Spectral radius $\rho(\mathscr{M})$ as function of ε. Left: $\alpha = 0.1$. Right: $\alpha = 1e{-}5$. The Dirichlet boundary condition is applied. First row: 1D-case, $N_t = 32$, $N_x = 31$. Second row: 2D-case, $N_t = 32, N_x = N_y = 15$.

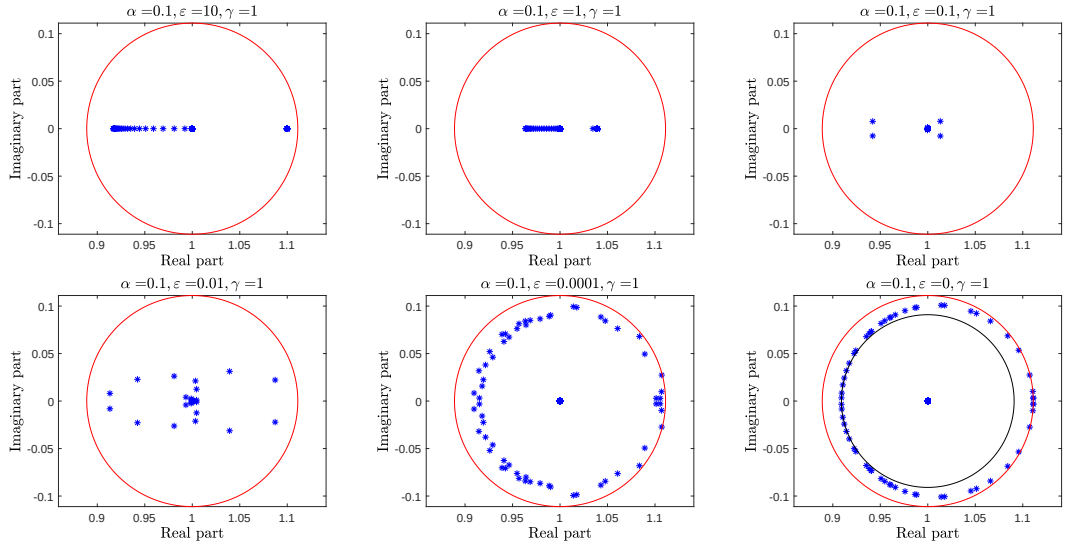

Figure 2: Example 5.1. The eigenvalues – blue stars – of $\mathscr{P}_\alpha^{-1}\mathscr{K}$ (Dirichlet boundary condition, $\alpha = 0.1$, $\gamma = 1$, $N_t = 32$, $N_x = 31$) are compared with the upper bound $\alpha/(1-\alpha)$ (red circle). In the last plot, lower bound $\alpha/(1+\alpha)$ for non-unit eigenvalues for the classical wave equation is shown in black.

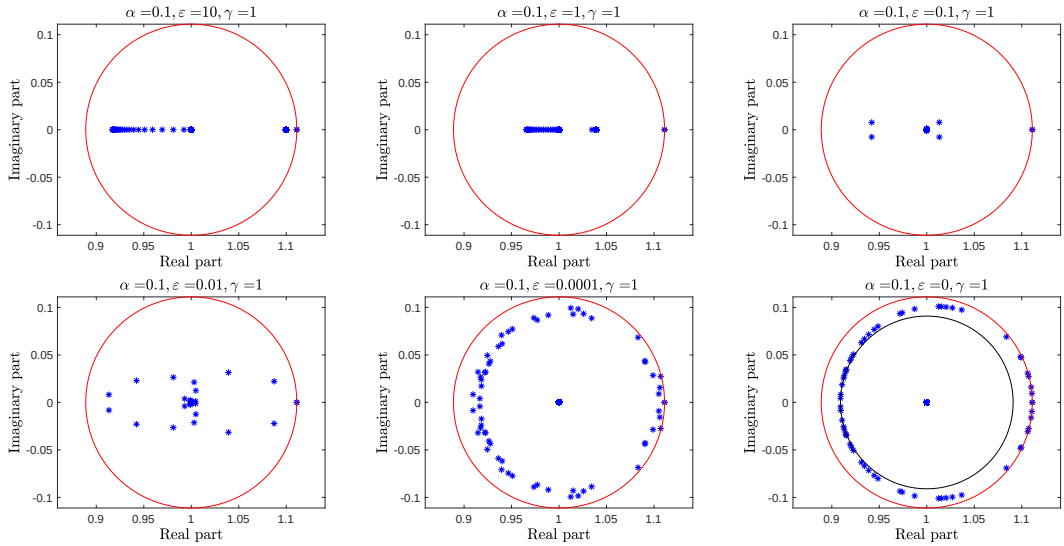

Figure 3: Example 5.3. The eigenvalues – blue stars – of $\mathscr{P}_\alpha^{-1}\mathscr{K}$ (Neumann boundary condition, $\alpha = 0.1$, $\gamma = 1$, $N_t = 32$, $N_x = 31$) are compared with the upper bound $\alpha/(1-\alpha)$ (red circle). In the last plot, lower bound $\alpha/(1+\alpha)$ for non-unit eigenvalues for classical wave equation is shown in black.

level and thus is still applicable in our algorithm. In fact, consider the following definition of speedup:

$$\text{speedup} = \frac{\mathbf{T}_{\text{serial}}}{\mathbf{T}_{\text{parallel}}},$$

where $\mathbf{T}_{\text{serial}}$ stands for the total cost for calculating the numerical solution in a time stepping fashion and $\mathbf{T}_{\text{parallel}}$ denotes the cost of obtaining the desired approximation using our algorithm. Clearly, we have $\mathbf{T}_{\text{serial}} = (N_t - 1)M$, where M is the total real floating point operations (flops) for solving at each time level. Suppose that $N_t - 1$ processors are available. According to [46], the cost $\mathbf{T}_{\text{diag}}$ for implementing the three-step algorithm (2.6) is

$$\begin{aligned}\mathbf{T}_{\text{diag}} = {} & 8m + 17m\log_2(N_t-1) + 3m\left(\lfloor \log_2(N_t-1)\rfloor + \log_2(N_t-1)\bmod 2\right)\\ & + 2\mathbf{T}_{\text{comm}} + 4M,\end{aligned}$$

where $\mathbf{T}_{\text{comm}}$ is the communication cost in the FFT algorithm and $m = N_x^d$ for the D-dimensional problem. Armed with these notations, one finds that the speedup for (2.6) satisfies

$$\text{speedup} = \frac{\mathbf{T}_{\text{Serial}}}{\mathbf{T}_{\text{parallel}}} = \frac{\mathbf{T}_{\text{Serial}}}{k\mathbf{T}_{\text{diag}}} = \frac{(N_t-1)M}{k\mathbf{T}_{\text{diag}}} \;\to\; \frac{N_t-1}{4k} \quad \text{as} \quad M \to \infty,$$

where k is the number of iterations required by our algorithm to reach a given precision. From the above asymptotic result, one finds that, for a fixed time step, a longer time interval $(0, T)$ would lead to a larger speedup. We comment here that the speedup is not related to the damping coefficient ε.

5. Numerical Results

In this section, we would like to provide several numerical examples to demonstrate our theoretical results. All simulations are implemented using MATLAB_R2016b, with a random initial guess and stop where the relative residual $\|r^k\|/\|r^0\|$ is less than a given tolerance tol. Without loss of generality, we choose $\gamma = 1$. We remark here that the Crank-Nicolson method is unconditionally stable, then there is no requirement on the mesh to be used for convergence.

We would like to illustrate the following theoretical results:

- The discretization is second-order accurate.
- The effect of the algorithm parameter α on the convergence behavior.
- The influence of the damping coefficient ε on the convergence behavior.
- The influence of the time interval $(0, T)$ on the convergence behavior.

5.1. 1D examples

In this subsection, we test our algorithm on several 1D examples with different properties. Noting that the solution to the viscoelastic equation decays exponentially in general, see [21, 42], we thus consider first the following 1D example modified from [24].

Example 5.1. Choose in the viscoelastic equation (1.1) $\Omega = (0,1)$, the initial conditions $y(x,0) = \sin(2\pi x)$, $y_t(x,0) = -\sin(2\pi x)$ and the source function

$$f(x,t) = (1 - 4\varepsilon\pi^2 + 4\gamma\pi^2)\sin(2\pi x)e^{-t}.$$

The exact solution reads

$$y(x,t) = \sin(2\pi x)e^{-t}.$$

A suitable driving force would lead the viscoelastic equation to a driven oscillations where the magnitude is not exponentially decay anymore. To test the performance of our algorithm on such model, we consider the following example.

Example 5.2. Let $\Omega = (0,1)$, $y(x,0) = \sin(\pi x)$, $y_t(x,0) = 0$ and the source function

$$f = (-1 + \gamma\pi^2)\sin(\pi x)\cos(t) - \varepsilon\pi^2\sin(\pi x)\sin(t).$$

The exact solution is

$$y(x,t) = \sin(\pi x)\cos(t).$$

To see the influence of the non-zero eigenvalue of L on our algorithm, we use the following example with Neumann boundary condition.

Example 5.3. Take $\Omega = (0,1)$, the initial conditions $y(x,0) = \cos(\pi x), y_t(x,0) = -\cos(\pi x)$, the boundary condition $\partial y/\partial \mathbf{n} = 0$ and the source function

$$f(x,t) = (1-\varepsilon\pi^2+\gamma\pi^2)\cos(\pi x)e^{-t}.$$

The exact solution now reads

$$y(x,t) = \cos(\pi x)e^{-t}.$$

Unless otherwise specified, we consider situation $T = 1$. We list the number of iterations required by our algorithm converging to a tolerance $tol = 10^{-10}$, as well as the corresponding errors and the numerical estimates of convergence order, in Table 1, with $\varepsilon = 0.5$ for $\alpha = 0.1$ and 0.01, respectively. We find that the second-order accuracy can be obtained in both time and space simultaneously. Here the order of accuracy is estimated by calculating the logarithmic ratio of the approximation errors between two successive refined meshes: $\log_2(e(h,\tau)/e(2h,2\tau))$.

We now test the effect of the algorithm parameter α on the convergence behavior. We solve the three examples mentioned above with damping coefficient $\varepsilon = 0.5$ by the fixed point iteration (2.4) and list the number of iterations required by converging to a tolerance $tol = 10^{-10}$ in Table 2 for different α. As a comparison, we also list the number of iterations predicted using (4.3). We find that the convergence is mesh independent, and depends only on the algorithm parameter α. In addition, the formula (4.3) predicts the number of iterations for Example 5.3 very well, while for Examples 5.1 and 5.2 it overestimates the number of iterations for relative large α. This observation conforms well our analysis: the upper bound $\alpha/(1-\alpha)$ is sharp for problems using the Neumann boundary conditions and the damping term $-\varepsilon\Delta y_t$ will lead to a smaller spectral radius of the iteration matrix $\mathscr{M}$ when Dirichlet boundary conditions are applied. To further study the effects

Table 1: Fixed point iteration (2.4). Examples 5.1, 5.2 and 5.3 with $\varepsilon = 0.5, tol = 10^{-10}$ for $\alpha = 0.1$ and $\alpha = 0.01$, respectively.

			$\alpha = 0.1$			$\alpha = 0.01$		
N_x	N_t	Ex	Error	Order	k	Error	Order	k
15	16	Ex 5.1	3.3974e-03	-	6	3.3974e-03	-	4
		Ex 5.2	2.0250e-03	-	6	2.0250e-03	-	4
		Ex 5.3	9.2623e-04	-	11	9.2623e-04	-	5
31	32	Ex 5.1	8.2769e-04	2.0372	6	8.2769e-04	2.0372	4
		Ex 5.2	5.0312e-04	2.0090	6	5.0312e-04	2.0090	4
		Ex 5.3	2.2851e-04	2.0191	11	2.2851e-04	2.0191	5
63	64	Ex 5.1	2.0561e-04	2.0092	6	2.0561e-04	2.0092	4
		Ex 5.2	1.2567e-04	2.0013	6	1.2567e-04	2.0013	4
		Ex 5.3	5.6894e-05	2.0059	11	5.6894e-05	2.0059	5
127	128	Ex 5.1	5.1312e-05	2.0026	6	5.1312e-05	2.0026	4
		Ex 5.2	3.1409e-05	2.0004	6	3.1409e-05	2.0004	4
		Ex 5.3	1.4201e-05	2.0023	11	1.4201e-05	2.0023	5

Table 2: Number of iterations required by fixed point iteration (2.4) when solving Example 5.1 (not enclosed), Example 5.2 (in parentheses) and Example 5.3 (in brackets) with $\varepsilon = 0.5$ for different α, $tol = 10^{-10}$.

		α							
N_x	N_t	0.1	0.01	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
31	32	6(6)[11]	4(4)[5]	3(3)[4]	3(3)[3]	2(2)[2]	2(2)[2]	2(2)[2]	2(2)[2]
63	64	6(6)[11]	4(4)[5]	3(3)[4]	3(3)[3]	2(2)[2]	2(2)[2]	2(2)[2]	2(2)[2]
127	128	6(6)[11]	4(4)[5]	3(3)[4]	3(3)[3]	2(2)[2]	2(2)[2]	2(2)[2]	2(2)[2]
255	256	6(6)[11]	4(4)[5]	3(3)[4]	3(3)[3]	2(2)[2]	2(2)[2]	2(2)[2]	2(2)[2]
511	512	6(6)[11]	4(4)[5]	3(3)[4]	3(3)[3]	2(2)[2]	2(2)[2]	2(2)[2]	2(2)[2]
$k(tol;\alpha)$		11	6	4	3	2	2	2	2

of smaller algorithm parameter α, we plot the errors for solving Example 5.1 as function of the iteration numbers for different α in the left panel of Fig. 4. From this plot we find that the smaller α leads to faster convergence, and all errors finally stagnate at a level determined by the discretization error. Thus no compromise between the machine precision and the discretization error is required as stated by [16]. However, if consider the following more economic scheme:

$$y_h^{k+1} = \mathscr{P}_\alpha^{-1}\left(\alpha\mathscr{R}y_h^k + b_h\right), \tag{5.1}$$

we find from the right plot of Fig. 4 that the roundoff error will pollute the numerical solution when α is too small. Hence, in this case the optimal α should be determined by compromising the discretization error with the machine precision, see [16] for more details. Note that mathematically the two iteration schemes (2.4) and (5.1) are equivalent, thus the different numerical performance of these two schemes deserves further investigation. We also comment here that from our numerical results there is no need to choose α extraordinarily small in practice.

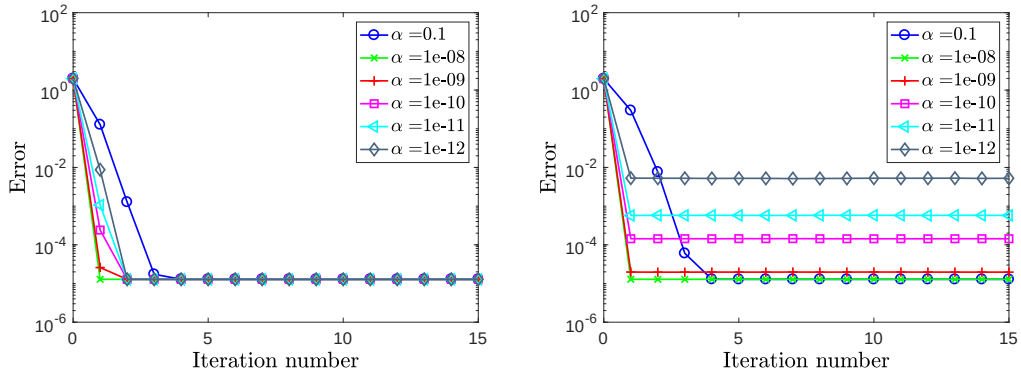

Figure 4: Example 5.1. Error reduction of fixed point iteration as function of the number of iterations for different α, $\varepsilon = 0.5$, $(N_x, N_t) = (255, 256)$. Left: Scheme (2.4). Right: Scheme (5.1).

To see the effect of the damping coefficient ε more clearly, we plot the number of iterations required by the scheme (2.4) converging to a tolerance $tol = 10^{-10}$ as function of ε in Fig. 5. Where we find that, when solving Examples 5.1 and 5.2, for $\alpha = 0.1$ the algorithm (2.4) converges much faster for $\varepsilon = 0.5$ than solving the classical wave equation ($\varepsilon = 0$). We observe as well that when ε is large, the required number of iterations is almost the same as for the classical wave equation. While this phenomenon does not happen for Example 5.3 as expected. In addition, similar observation happens as well for $\alpha = 0.01$, but the difference is not significant anymore, even for $\varepsilon = 0.5$. This phenomenon is also illustrated in Fig. 6 for Example 5.1 (similar plots can be obtained for Example 5.2) and Fig. 7 for Example 5.3 from the perspective of error reduction, where the iterative error finally stagnates at the level of the discretization error.

Now, we study how robust our algorithm is on the time interval $(0, T)$. To this end, we plot in Fig. 8 the number of iterations required by our algorithm converging to a tolerance $tol = 10^{-10}$ as function of T with parameters $\varepsilon = 0.5$, $h = \tau = 1/128$. We find that our algorithm is very robust in T for all examples. In particular, for Examples 5.1 and 5.2, when α is relatively large, the longer the time interval is, the faster the algorithm converges, this is because larger T (correspondingly larger N_t for fixed time step size) will lead to smaller spectral radius of the iteration matrix. However, for Example 5.3, the spectral radius is exactly $\alpha/(1-\alpha)$ due to the Neumann boundary condition, hence the number of iterations does not vary with the time interval $(0, T)$.

Next, we test how a non-smooth initial condition will affect the performance of our algorithm. To this end, we consider the following model problem [14], which gives a more realistic description of a plucked string.

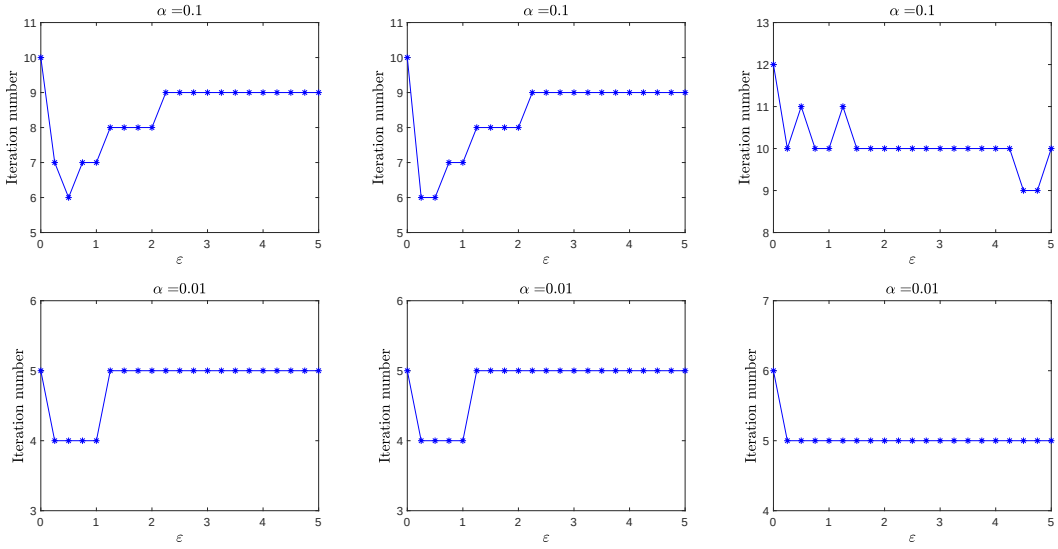

Figure 5: Number of iterations required by fixed point iteration (2.4), with $N_x = 31$, $N_t = 32$, converging to a tolerance $tol = 10^{-10}$ as function of ε. Left column: Example 5.1. Middle column: Example 5.2. Right column: Example 5.3. First row: $\alpha = 0.1$. Second row: $\alpha = 0.01$.

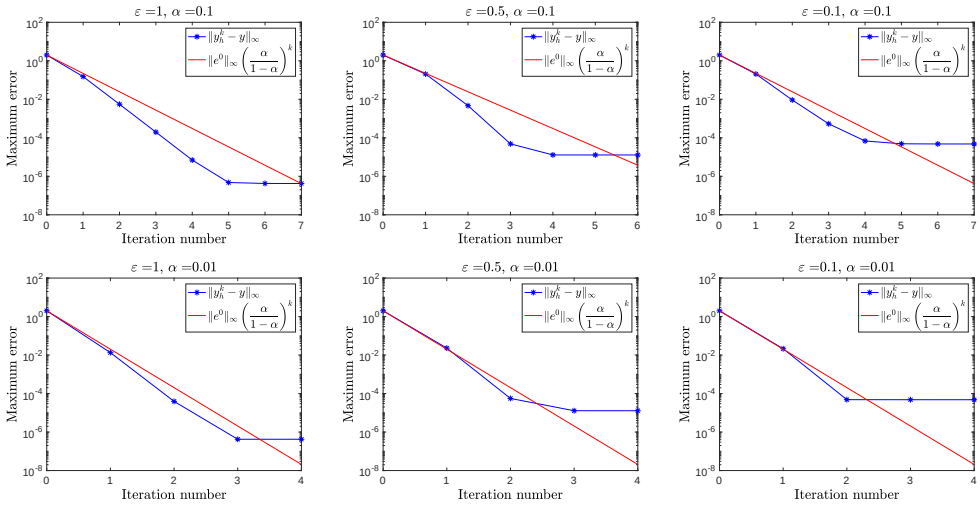

Figure 6: Example 5.1. Error reduction of fixed point iteration (2.4), $N_x = 255$, $N_t = 256$, compared with the prediction provided using the upper bound $\alpha/(1-\alpha)$.

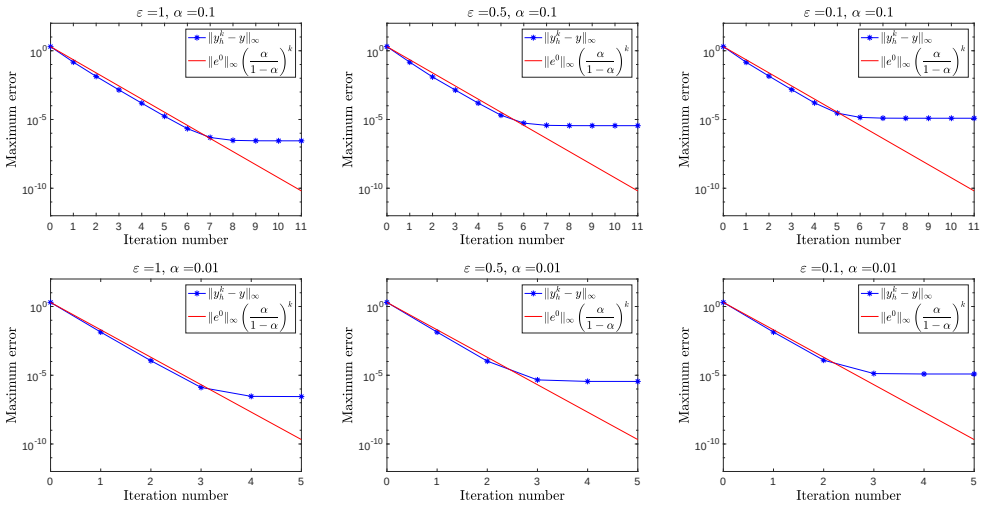

Figure 7: Example 5.3. Error reduction of fixed point iteration (2.4), $N_x = 255$, $N_t = 256$, compared with the prediction provided using the upper bound $\alpha/(1-\alpha)$.

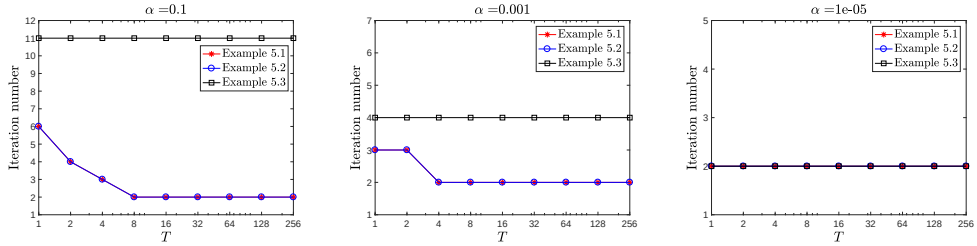

Figure 8: Number of iterations required by fixed point iteration (2.4) as function of T for different α, $\varepsilon = 0.5, tol = 10^{-10}, h = \tau = 1/128$.

Example 5.4. Let $\Omega=(0,1)$, $f=0$,

$$y(x,0)=\begin{cases}2x, & 0<x\le 1/2,\\ 2-2x, & 1/2<x<1,\end{cases}$$

and $y_t(x,0)=0$. The exact solution is

$$\begin{aligned}y(x,t)=&\sum_{1\le n<M}\sin(n\pi x)e^{-\alpha_n t}\big(a_n\cosh(\mu_n t)+b_n\sinh(\mu_n t)\big)\\&+\sum_{n>M}\sin(n\pi x)e^{-\alpha_n t}\big(a_n\cos(\mu_n t)+b_n\sin(\mu_n t)\big)\\&+\chi_{M=[M]}\sin(M\pi x)e^{-\alpha_M t}(a_M+b_M t),\end{aligned}$$

where

$$\alpha_n=\frac{\varepsilon n^2\pi^2}{2},\quad \mu_n=\frac{n\pi}{2}\sqrt{\varepsilon^2n^2\pi^2-4\gamma},$$
$$a_n=\frac{8}{n^2\pi^2}\sin\Big(\frac{n\pi}{2}\Big),\quad b_n=\frac{\alpha_n a_n}{\mu_n},\quad M=\frac{2\sqrt{\gamma}}{\varepsilon\pi}$$

and

$$\chi_{M=[M]}=\begin{cases}1, & \text{if}\quad M=[M],\\ 0, & \text{otherwise}\end{cases}$$

with $[M]$ being the integer part of M.

The solution to Example 5.4 has a low regularity due to the non-smooth initial condition. Hence, the central difference scheme cannot reach its second-order accuracy. Particularly, at the initial stage, it is very hard to damp the error. We then consider only the numerical results at late stage $T=1$. In the left panel of Fig. 9, we find that it is hard

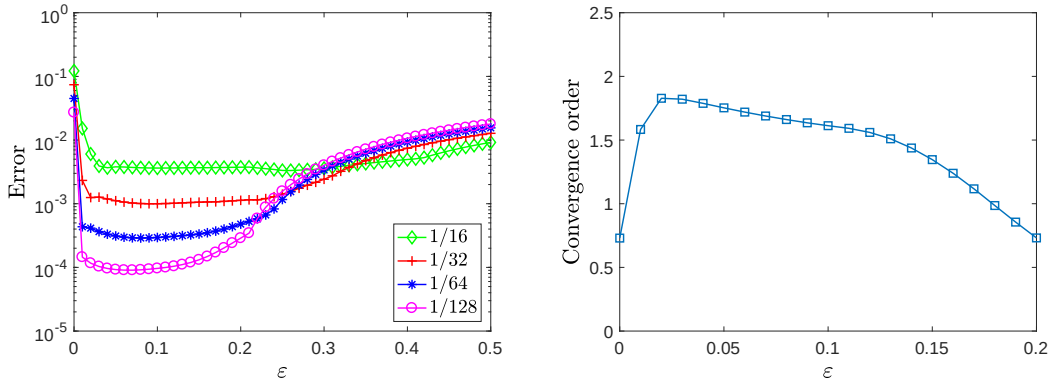

Figure 9: Example 5.4. Maximum error and convergence order at $T=1$ as function of ε with $\alpha=0.1$. Left: Maximum errors correspond to different mesh sizes. Right: Convergence order calculated from the numerical results with $\tau=1/64$ and $1/128$.

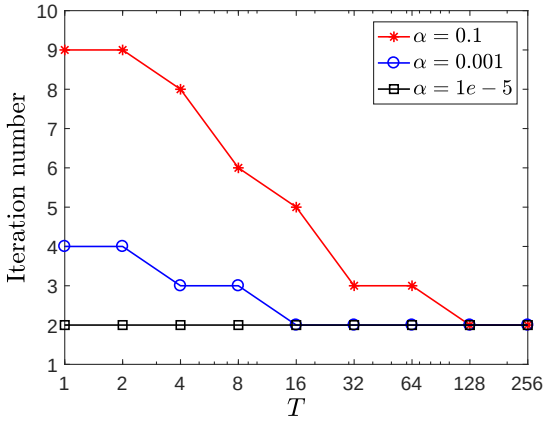

Figure 10: Example 5.4. Number of iterations required by fixed point iteration (2.4) converging to $tol = 10^{-10}$ as function of T, $\varepsilon = 0.05, h = \tau = 1/128$.

to reduce the error, especially for very small or very large damping coefficient ε. Interestingly, when ε is relatively small, say near $\varepsilon = 0.1$, the scheme performs very well. This observation is further convinced by the right plot of Fig. 9. We comment here that the above phenomena are essentially determined by the discretization method, not related to our iteration scheme (2.4). In fact, we observe exactly the same results when solving Example 5.4 using the time stepping scheme (2.1). We now test the performance of our algorithm. When solving Example 5.4 for $\varepsilon = 0.05$ and $T = 1$ with varying α, we obtain results very similar to those reported in Table 2, except where for each column the iteration numbers are replaced by $9, 5, 4, 3, 2, 2, 2, 2$, which show that our algorithm is still mesh independent and depends only on α. For varying T, we plot our results in Fig. 10, which shows a result similar to Example 5.1.

5.2. A 2D example

As a 2D example [24], we choose in the viscoelastic equation (1.1) $\Omega = (0,1)^2$, $T = 1$, the initial condition

$$y(x_1, x_2, 0) = \sin(2\pi x_1)\sin(2\pi x_2),$$
$$y_t(x_1, x_2, 0) = -\sin(2\pi x_1)\sin(2\pi x_2),$$

and the source function

$$f = (1 - 8\varepsilon\pi^2 + 8\gamma\pi^2)\sin(2\pi x_1)\sin(2\pi x_2)e^{-t}.$$

We find that the exact solution is

$$y(x_1, x_2, t) = \sin(2\pi x_1)\sin(2\pi x_2)e^{-t}.$$

Experiments similar to those for Example 5.1 are performed and we list the results in Tables 3 and 4, as well as Fig. 11. Results similar to the 1D case can be concluded, we hence omit the details.

Table 3: Fixed point iteration (2.4) for 2D viscoelastic equation, $\varepsilon = 0.5$, $tol = 10^{-10}$.

		$\alpha = 0.1$			$\alpha = 0.01$		
(N_x, N_y)	N_t	Error	Order	k	Error	Order	k
(15,15)	16	3.3847e-03	-	6	3.3847e-03	-	4
(31,31)	32	8.1233e-04	2.0589	6	8.1233e-04	2.0589	4
(63,63)	64	2.0098e-04	2.0150	6	2.0098e-04	2.0150	4
(127,127)	128	5.0113e-05	2.0038	6	5.0113e-05	2.0038	4

Table 4: Number of iterations required by fixed point iteration (2.4) for 2D viscoelastic equation (1.1), $\varepsilon = 0.5$, $tol = 10^{-10}$.

		α							
(N_x, N_y)	N_t	0.1	0.01	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
(15,15)	16	6	4	3	3	2	2	2	2
(31,31)	32	6	4	3	3	2	2	2	2
(63,63)	64	6	4	3	3	2	2	2	2
(127,127)	128	6	4	3	3	2	2	2	2
$k(tol;\alpha)$		11	6	4	3	2	2	2	2

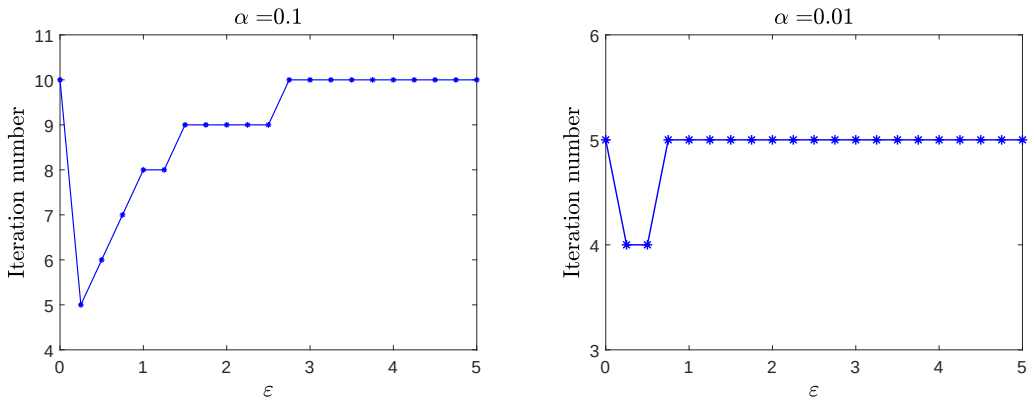

Figure 11: Number of iterations required by fixed point iteration (2.4) converging to a tolerance $tol = 10^{-10}$ as function of ε for 2D viscoelastic equation (1.1), $N_x = N_y = 31$, $N_t = 32$. Left: $\alpha = 0.1$. Right: $\alpha = 0.01$.

5.3. A 3D example

To test a 3D example, we choose $\Omega = (0,1)^3$, $T = 1$, the initial condition

$$y(x_1, x_2, x_3, 0) = \sin(2\pi x_1)\sin(2\pi x_2)\sin(2\pi x_3),$$
$$y_t(x_1, x_2, x_3, 0) = -\sin(2\pi x_1)\sin(2\pi x_2)\sin(2\pi x_3),$$

and the source function

$$f = (1 - 12\varepsilon\pi^2 + 12\gamma\pi^2)\sin(2\pi x_1)\sin(2\pi x_2)\sin(2\pi x_3)e^{-t}.$$

Table 5: Fixed point iteration (2.4) for 3D viscoelastic equation with $\varepsilon = 0.5$, $tol = 10^{-10}$.

		$\alpha = 0.1$			$\alpha = 0.01$		
(N_x, N_y, N_z)	N_t	Error	Order	k	Error	Order	k
(7,7,7)	8	2.0123e-02	-	6	2.0123e-02	-	4
(15,15,15)	16	3.4354e-03	2.5503	6	3.4354e-03	2.5503	4
(31,31,31)	32	8.1015e-04	2.0842	6	8.1015e-04	2.0842	4

Table 6: Number of iterations required by fixed point iteration (2.4) for 3D viscoelastic equation (1.1), $\varepsilon = 0.5$, $tol = 10^{-10}$.

		α							
(N_x, N_y, N_z)	N_t	0.1	0.01	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
(7,7,7)	8	6	4	3	3	2	2	2	2
(15,15,15)	16	6	4	3	3	2	2	2	2
(31,31,31)	32	6	4	3	3	2	2	2	2
$k(tol;\alpha)$		11	6	4	3	2	2	2	2

In this case the exact solution is

$$y(x_1, x_2, x_3, t) = \sin(2\pi x_1)\sin(2\pi x_2)\sin(2\pi x_3)e^{-t}.$$

The numerical results listed in Tables 5 and 6 imply again the observations in the 1D case.

6. Conclusion

In this paper, we extend a diagonalization-based PinT algorithm to solve the viscoelastic equation efficiently. Taking an algebraic system obtained by discretizing the viscoelastic equation via the Crank-Nicolson and the central difference schemes as an example, we propose to solve it using a fixed point iteration preconditioned by a block α-circulant matrix. A rigorous analysis shows that the spectral radius of the preconditioned iteration matrix is uniformly bounded from above by $\alpha/(1-\alpha)$, an upper bound independent of the model parameters and the discretization parameters, which means the smaller the algorithm parameter α is, the faster the algorithm converges. Unlike the classical wave equation with Dirichlet boundary conditions, the upper bound is not sharp anymore, hence the damping term, as well as the large time interval $(0, T)$, leads to even faster convergence. The proposed algorithm can be easily extended to the algebraic problems resulting from other discretization schemes.

Acknowledgments

The authors appreciate very much the anonymous referees for their valuable comments and suggestions, which greatly improved the quality of this paper.

This work was supported by NSFC-12071069, by the National Key R&D Program of China (Grant 2020YFA0714102), by the Science and Technology Development Planning of Jilin Province (Grant YDZJ202201ZYTS573), and by the Fundamental Research Funds for the Central Universities (Grant JGPY202101).

References

[1] R.A. Adey and C.A. Brebbia, *Efficient method for solution of viscoelastic problems*, J. Eng. Mech. Div. **99**(6), 1119–1127 (1973).
[2] G. Andrews, *On the existence of solutions to the equation* $u_{tt} = u_{xxt} + \sigma(u_x)_x$, J. Differential Equations, **35**(2), 200–231 (1980).
[3] E. Bécache, A. Ezziani and P. Joly, *A mixed finite element approach for viscoelastic wave propagation*, Comput. Geosci. **8**, 255–299 (2005).
[4] D. Bertaccini and M.K. Ng, *Block* $\{\omega\}$*-circulant preconditioners for the systems of differential equations*, Calcolo, **40**(2), 71–90 (2003).
[5] P.-H. Cocquet and M.J. Gander, *How large a shift is needed in the shifted Helmholtz preconditioner for its effective inversion by multigrid*? SIAM J. Sci. Comput. **39**(2), A438–A478 (2017).
[6] C.R. Dohrmann, *A preconditioner for substructuring based on constrained energy minimization*, SIAM J. Sci. Comput. **25**(1), 246–258 (2003).
[7] V. Dolean, P. Jolivet and F. Nataf, *An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation*, SIAM (2015).
[8] C. Farhat and F.-X. Roux, *A method of finite element tearing and interconnecting and its parallel solution algorithm*, Internat. J. Numer. Methods Engrg. **32**(6), 1205–1227 (1991).
[9] Y.P. Fiscales, *Finite element methods for the approximate solution of the linear viscoelastic wave equation*, Comput. Appl. Math. **4**(1), 35 (1985).
[10] M.J. Gander, *Optimized Schwarz methods*, SIAM J. Numer. Anal. **44**(2), 699–731 (2006).
[11] M.J. Gander, *50 years of time parallel time integration*, in: *Multiple Shooting and Time Domain Decomposition Methods*, Springer, 69–113 (2015).
[12] M.J. Gander, L. Halpern, J. Rannou and J. Ryan, *A direct time parallel solver by diagonalization for the wave equation*, SIAM J. Sci. Comput. **41**(1), A220–A245 (2019).
[13] M.J. Gander, J. Liu, S.-L. Wu, X. Yue and T. Zhou, *Paradiag: Parallel-in-time algorithms based on the diagonalization technique*, arXiv:2005.09158 (2021).
[14] M.J. Gander, L. Thibaut and P. Ausra, *Convergence of parareal for a vibrating string with viscoelastic damping*, in: *Domain Decomposition Methods in Science and Engineering XXVI*, Springer, 435–442 (2023).
[15] M.J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, SIAM J. Sci. Comput. **29**(2), 556–578 (2007).
[16] M.J. Gander and S.-L. Wu, *A diagonalization-based parareal algorithm for dissipative and wave propagation problems*, SIAM J. Numer. Anal. **58**(5), 2981–3009 (2020).
[17] L. Gao, D. Liang and B. Zhang, *Error estimates for mixed finite element approximations of the viscoelasticity wave equation*, Math. Methods Appl. Sci. **27**(17), 1997–2016 (2004).
[18] A. Goddard and A. Wathen, *A note on parallel preconditioning for all-at-once evolutionary PDEs*, Electron. Trans. Numer. Anal. **51**, 135–150 (2019).
[19] G.H. Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins University Press (2013).
[20] V.K. Kalyani, Pallavika and S.K. Chakraborty, *Finite-difference time-domain method for modelling of seismic wave propagation in viscoelastic media*, Appl. Math. Comput. **237**, 133–145 (2014).

[21] S. Kawashima and Y. Shibata, *Global existence and exponential stability of small solutions to nonlinear viscoelasticity*, Comm. Math. Phys. **148**, 189–208 (1992).
[22] H. Li, *Generalized difference methods for one-dimensional viscoelastic problems*, J. Korean Soc. Ind. Appl. Math. **9**(2), 55–64 (2005).
[23] H. Li, P. Sun, Y. Shang and Z. Luo, *A fully discrete finite volume element formulation and numerical simulations for viscoelastic equations*, Math. Numer. Sin. **34**(4), 413 (2012).
[24] H. Li, Z. Zhao and Z. Luo, *A space-time continuous finite element method for 2D viscoelastic wave equation*, Bound. Value Probl. **2016**, 1–17 (2016).
[25] X. Li, P. Sun, J. An and Z. Luo, *A new splitting positive definite mixed finite element method for viscoelastic equation*, Math. Numer. Sin. **35**(1), 49 (2013).
[26] Y. Lin, *A mixed type boundary problem describing the propagation of disturbances in viscous media I, weak solution for quasi-linear equations*, J. Math. Anal. Appl. **135**, 644–653 (1988).
[27] J.-L. Lions, Y. Maday and G. Turinici, *Résolution d'EDP par un schéma en temps «pararéel»*, Comptes Rendus de l'Académie des Sciences-Series I-Mathematics, **332**(7), 661–668 (2001).
[28] P.-L. Lions, *On the Schwarz alternating method. III: A variant for nonoverlapping subdomains*, in: *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia: SIAM **6**, pp. 202–223 (1990).
[29] J. Liu and S.-L. Wu, *A fast block α-circulant preconditoner for all-at-once systems from wave equations*, SIAM J. Matrix Anal. Appl. **41**(4), 1912–1943 (2020).
[30] Z. Luo and S. Jin, *A reduced-order extrapolated Crank-Nicolson collocation spectral method based on proper orthogonal decomposition for the two-dimensional viscoelastic wave equations*, Numer. Methods Partial Differential Equations **36**(1), 49–65 (2020).
[31] Y. Maday and E.M. Rønquist, *Parallelization in time through tensor-product space-time solvers*, C. R. Math. **346**(1-2), 113–118 (2008).
[32] E. McDonald, J. Pestana and A. Wathen, *Preconditioning and iterative solution of all-at-once systems for evolutionary partial differential equations*, SIAM J. Sci. Comput. **40**, A1012–A1033 (2018).
[33] H. Nguyen and R. Tsai, *A stable parareal-like method for the second order wave equation*, J. Comput. Phys. **405**, 109156 (2020).
[34] J. Nievergelt, *Parallel methods for integrating ordinary differential equations*, Communications of the ACM, **7**(12), 731–733 (1964).
[35] O. Nikan and Z. Avazzadeh, *Coupling of the Crank-Nicolson scheme and localized meshless technique for viscoelastic wave model in fluid flow*, J. Comput. Appl. Math. **398**, 113695 (2021).
[36] J.A. Olkin, *Linear and Nonlinear Deconvolution Problems (Optimization)*, PhD Thesis, Rice University (1986).
[37] Ö. Oruç, *Two meshless methods based on local radial basis function and barycentric rational interpolation for solving 2D viscoelastic wave equation*, Comput. Math. Appl. **79**(12), 3272–3288 (2020).
[38] A.K. Pani and J.Y. Yuan, *Mixed finite element method for a strongly damped wave equation*, Numer. Methods Partial Differential Equations **17**(2), 105–119 (2001).
[39] V. Pata and M. Squassina, *On the strongly damped wave equation*, Comm. Math. Phys. **253**(3), 511–533 (2005).
[40] D. Ruprecht, *Wave propagation characteristics of parareal*, Comput. Vis. Sci. **19**(1), 1–17 (2018).
[41] H.A. Schwarz, *Ueber einen Grenzübergang durch alternirendes Verfahren*, Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich **15**, 272–286 (1870).
[42] Y. Shibata, *On the rate of decay of solutions to linear viscoelastic equation*, Math. Methods Appl. Sci. **23**(3), 203–226 (2000).

[43] K. Shukla, J. Chan and V. Maarten, *A high order discontinuous Galerkin method for the symmetric form of the anisotropic viscoelastic wave equation*, Comput. Math. Appl. **99**, 113–132 (2021).
[44] G. Strang, *A proposal for Toeplitz matrix calculations*, Stud. Appl. Math. **74**(2), 171–176 (1986).
[45] P. Stucky and W. Lord, *Finite element modeling of ultrasonic waves in viscoelastic media*, in: *Review of Progress in Quantitative Nondestructive Evaluation*, Springer, 113–120 (1997).
[46] Y. Sun, S.-L. Wu and Y. Xu, *A parallel-in-time implementation of the Numerov method for wave equations*, J. Sci. Comput. **90**(1), 1–31 (2022).
[47] A. Toselli and O. Widlund, *Domain Decomposition Methods-Algorithms and Theory*, Vol. 34, Springer Science & Business Media (2004).
[48] U. Trottenberg, C.W. Oosterlee and A. Schuller, *Multigrid*, Elsevier (2000).
[49] X. Wang, F. Gao and Z. Sun, *Weak Galerkin finite element method for viscoelastic wave equations*, J. Comput. Appl. Math. **375**, 112816 (2020).
[50] Y. Wang, *Generalized viscoelastic wave equation*, Geophys. J. Int. **204**(2), 1216–1221 (2016).
[51] S.-L. Wu and T. Zhou, *Parallel implementation for the two-stage SDIRK methods via diagonalization*, J. Comput. Phys. **428**, 110076 (2021).
[52] S.-L. Wu, T. Zhou and Z. Zhou, *A uniform spectral analysis for a preconditioned all-at-once system from first-order and second-order evolutionary problems*, SIAM J. Matrix Anal. Appl. **43**(3), 1331–1353 (2022).
[53] H. Xia and Z. Luo, *A POD-based-optimized finite difference CN-extrapolated implicit scheme for the 2D viscoelastic wave equation*, Math. Methods Appl. Sci. **40**(18), 6880–6890 (2017).
[54] Y. Yuan, *Finite difference method and analysis for three-dimensional semiconductor device of heat conduction*, Sci. China, Ser. A. **39**(11),1140–1151 (1996).