PARALLEL MULTISTEP METHODS FOR LINEAR EVOLUTION PROBLEMS

LEHEL BANJAI AND DANIEL PETERSEIM*

Abstract. Time-stepping procedures for the solution of evolution equations can be performed on parallel architecture by parallelizing the space computation at each time step. This, however, requires heavy communication between processors and becomes inefficient when many time-steps are to be computed and many processors are available. In such cases parallelization in time is advantageous.

In this paper we present a method for parallelization in time of linear multistep discretizations of linear evolution problems; we consider a model parabolic and a model hyperbolic problem, and their, respectively, $A(\theta)$-stable and A-stable linear multistep discretizations. The method consists of a discrete decoupling procedure, whereby $N+1$ decoupled Helmholtz problems with complex frequencies are obtained; N being the number of time steps computed in parallel. The usefulness of the method rests on our ability to solve these Helmholtz problems efficiently. We discuss the theory and give numerical examples for multigrid preconditioned iterative solvers of relevant complex frequency Helmholtz problems. The parallel approach can easily be combined with a time-stepping procedure, thereby obtaining a block time-stepping method where each block of steps is computed in parallel. In this way we are able to optimize the algorithm with respect to the number of processors available, the difficulty of solving the Helmholtz problems, and the possibility of both time and space adaptivity. Extensions to other linear evolution problems and to Runge-Kutta time discretization are briefly mentioned.

keywords: wave equation, heat equation, Helmholtz equation, triangular Toeplitz systems, shifted Laplacian preconditioner, multigrid.

1. Introduction

We describe a parallel numerical method for the solution of linear evolution problems. The two model problems we discuss are the heat and the wave equation. After a time discretization by a linear multistep method, a semi-discrete, lower triangular Toeplitz system of equations is obtained. We will show that up to a controllable error this system is equivalent to $N+1$ decoupled Helmholtz problems with complex frequencies; N is here the number of time steps computed in parallel. The complex wavenumbers depend on the linear multistep method, the underlying evolution problem, and the number of time steps computed in parallel. We discuss $A(\theta)$-stable linear multistep methods for the parabolic problem and A-stable and explicit linear multistep methods for the hyperbolic problem. An important property of our approach is that it can be combined with a time-stepping procedure: after a certain number of time steps computed in parallel, the computation can be restarted in a sequential way. Thereby we are able to use adaptivity in time and space. In this paper we will use the Galerkin finite element method (FEM)

* The second author is supported by the DFG Research Center MATHEON Berlin.

to discretize the problem in space, though other discretization methods could also be used.

Let us stress that we do not require the computation of inverse Laplace transforms. This allows uniform discussion of both parabolic and hyperbolic problems and makes our approach different from the methods developed for parabolic problems in [15, 22]. In order to obtain a decoupled system of Helmholtz problems we use purely algebraic transformations including the discrete Fourier transform; the resulting algorithm is also known as Bini's algorithm for approximate inversion of special Toeplitz matrices [3]. The original algorithm is restricted to accuracy $\sqrt{\text{eps}}$ where eps is the machine precision; $\text{eps} \sim 10^{-16}$ for double precision computations. We describe a correction term that can efficiently be computed and that reduces the maximum accuracy to $\text{eps}^{2/3}$.

The effectiveness of the above described method rests on our ability to efficiently solve the linear systems arising from the FEM discretization of the Helmholtz problems. We will describe a multigrid preconditioned GMRES iteration for the solution of these systems. We will argue both theoretically and experimentally the following points.

- $A(\theta)$-stable time discretization of linear parabolic problems: the speed of convergence of the preconditioned GMRES is independent of either the time or space discretization parameters or of the number of time steps computed in parallel.
- A-stable time discretization of linear hyperbolic problems: if the computational (time) interval $[0, T]$ is fixed and the time step Δt decreases, then for A-stable, pth order BDF methods the number of iterations increases as $\Delta t^{-p/(p+1)}$.
- A-stable time discretization of linear hyperbolic problems: if the time step Δt is fixed and the number of time-steps to be computed in parallel is increased, then the iteration count is bounded for BDF methods and increases linearly for the Trapezoidal rule.

Hence, in some situations, the iteration count is negatively affected by the number of time steps that are computed in parallel. Since we can control the number of time steps computed in parallel by combining the parallel computation with time stepping, this problem can be avoided. In the numerical examples we will show that we can nevertheless efficiently compute hundreds of time steps in parallel for high order $A(\theta)$-stable discretization of the heat equation and A-stable discretization of the wave equation.

Though in this paper we restrict the discussion to the heat equation and the wave equation and to linear multistep discretizations, the methods described are extendible to other linear evolution problems and other time discretizations based on equal time steps. In particular, an important generalization which we only briefly describe, is to Runge-Kutta time discretization.

The investigation of the efficient solution of Helmholtz problems with complex frequencies is of interest beyond our decoupling procedure. Such problems need to be solved also in methods resting on computation of inverse Laplace transforms; see [15] and [22].

2. Triangular Toeplitz systems

In this section we will describe how to approximate a lower triangular matrix by a matrix that can be inverted in $\mathcal{O}(N \log N)$ time. This method is often attributed to Bini [3] and has a number of times been re-interpreted (and re-invented), see [2, 14]. In

fact Bini's method is also closely related to the earlier work by Schönhage [21] on fast methods for operations on polynomials. Here we give a short proof of the main result.

To shorten the presentation we will use the following notation: Given a vector $\mathbf{c} = (c_0, c_1, \ldots, c_N)^T$, $T_U(\mathbf{c})$ is the upper triangular Toeplitz matrix with $\mathbf{c}$ as the first row, similarly $T_L(\mathbf{c})$ is the lower triangular Toeplitz matrix with $\mathbf{c}$ as the first column, and $C(\mathbf{c})$ is the circulant matrix with $\mathbf{c}$ as the first column, i.e.

$$T_U(\mathbf{c}) = \begin{pmatrix} c_0 & c_1 & \cdots & c_{N-1} & c_N \\ 0 & c_0 & \cdots & c_{N-2} & c_{N-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & c_0 & c_1 \\ 0 & 0 & \cdots & 0 & c_0 \end{pmatrix}, \; T_L(\mathbf{c}) = T_U(\mathbf{c})^T, \text{ and } C(\mathbf{c}) = \begin{pmatrix} c_0 & c_N & \cdots & c_2 & c_1 \\ c_1 & c_0 & \cdots & c_3 & c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{N-1} & c_{N-2} & \cdots & c_0 & c_N \\ c_N & c_{N-1} & \cdots & c_1 & c_0 \end{pmatrix}.$$

We refer to [13] for a detailed introduction to Toeplitz and circulant matrices. We will also make use of a permutation-like operator $P \in \mathbb{R}^{(N+1)\times(N+1)}$ defined by

$$P\mathbf{c} = (0, c_N, c_{N-1}, \ldots, c_1)^T, \qquad \text{for all } \mathbf{c} = (c_0, c_1, \ldots, c_N)^T. \tag{1}$$

Lemma 2.1. *Let $\mathbf{c} \in \mathbb{C}^{N+1}$ and let $\Lambda = \operatorname{diag}(1, \lambda, \lambda^2, \ldots, \lambda^N)$, where $0 < \lambda < 1$. Then*

$$T_L(\mathbf{c}) = \Lambda^{-1} C(\mathbf{c}_\lambda)\Lambda - \lambda^{N+1} T_U(P\mathbf{c}), \quad \textit{where } \mathbf{c}_\lambda = \Lambda\mathbf{c}.$$

Proof. Multiplication of a vector $\mathbf{x}$ by $T_L(\mathbf{c})$ can be written as

$$(T_L(\mathbf{c})\mathbf{x})_n = \sum_{j=0}^{n} c_{n-j} x_j = \lambda^{-n} \sum_{j=0}^{n} \lambda^{n-j} c_{n-j} \lambda^j x_j, \quad n = 0, 1, \ldots, N,$$

which shows that

$$T_L(\mathbf{c}) = \Lambda^{-1} T_L(\mathbf{c}_\lambda)\Lambda = \Lambda^{-1} C(\mathbf{c}_\lambda)\Lambda - \Lambda^{-1} T_U(P\mathbf{c}_\lambda)\Lambda.$$

It is now easily checked that $\Lambda^{-1} T_U(P\mathbf{c}_\lambda)\Lambda = \lambda^{N+1} T_U(P\mathbf{c})$. □

The circulant matrix $C(\mathbf{c}_\lambda)$ is diagonalized by the discrete Fourier change of basis [13], therefore the above result allows us to invert an approximation to the lower triangular Toeplitz matrix by using the FFT and inversion of a diagonal matrix. In later sections the entries of the diagonal matrix will be operators, or matrices, hence there we will greatly benefit from the trivial parallelization of the approximate inversion procedure.

Remark 2.2. *Considering $\Lambda^{-1} C(\mathbf{c}_\lambda)\Lambda$ as an approximation of the lower triangular Toeplitz matrix $T_L(\mathbf{c})$, the error matrix $\Lambda^{-1} T_U(P\mathbf{c}_\lambda)\Lambda = \lambda^{N+1} T_U(P\mathbf{c})$ has entries of order $\mathcal{O}(\lambda^{N+1})$, nevertheless in finite precision arithmetic arbitrary accuracy cannot be obtained as the matrix Λ is highly ill-conditioned for small λ. If* eps *is the machine precision then multiplication by Λ^{-1} increases the rounding errors to λ^{-N} eps, hence in order to make the combined error $\lambda^{-N}\,\text{eps} + \lambda^{N+1}$ as small as possible the optimal choice of the parameter is $\lambda = \text{eps}^{\frac{1}{2N+1}} \approx \sqrt{\text{eps}}^{\frac{1}{N}}$ giving accuracy $\text{eps}^{\frac{N+1}{2N+1}} \approx \sqrt{\text{eps}}$. In double precision arithmetic* $\text{eps} \approx 10^{-16}$*, however, to allow for further accumulation of rounding errors we have found it advisable to choose $\lambda = 10^{\frac{-6}{N}}$; the choice is motivated by experimental evidence and is used throughout in the numerical experiments of Section 6.1.*

2.1. **Coefficients given by a Taylor expansion.** The coefficients c_j in Lemma 2.1 can be thought of as the leading coefficients of a Taylor expansion

$$c(\zeta) = \sum_{j=0}^{\infty} c_j \zeta^j.$$

In this case we will still use the notation $\mathbf{c} = (c_0, c_1, \dots, c_{N+1})^T$, i.e., $\mathbf{c}$ will represent the vector of leading $N+1$ coefficients of the generating function c.

In order to easily invert the circulant matrix $C(\mathbf{c}_\lambda)$, we can write it as

$$C(\mathbf{c}_\lambda) = \mathcal{F}_{N+1}^{-1} \operatorname{diag}(\mathcal{F}_{N+1}\mathbf{c}_\lambda)\mathcal{F}_{N+1},$$

where the matrix $\mathcal{F}_{N+1}$ represents the discrete Fourier transform:

$$(\mathcal{F}_{N+1}\boldsymbol{x})_n = \sum_{j=0}^{N} x_j \zeta_{N+1}^{jn}, \qquad \zeta_{N+1} = e^{-2\pi \mathrm{i}/(N+1)}, \quad n = 0, 1, \dots, N.$$

If $c(\zeta)$ is known explicitly, it may be advantageous to approximate $C(\boldsymbol{c}_\lambda)$ as follows

$$C(\boldsymbol{c}_\lambda) \approx \mathcal{F}_{N+1}^{-1} \operatorname{diag}\left(c(\lambda), c(\lambda\zeta_{N+1}^1), \dots, c(\lambda\zeta_{N+1}^N)\right) \mathcal{F}_{N+1}. \tag{2}$$

If $c(\zeta)$ is a polynomial of degree k, then for $N \geq k$, $c(\lambda\zeta_{N+1}^n) = (\mathcal{F}_{N+1}\boldsymbol{c}_\lambda)_n$ and no error is committed in (2), otherwise

$$c(\lambda\zeta_{N+1}^n) - (\mathcal{F}_{N+1}\boldsymbol{c}_\lambda)_n = \sum_{j=N+1}^{\infty} \lambda^j c_j \zeta_{N+1}^{nj},$$

which finally gives an $\mathcal{O}(\lambda^{N+1})$ error in the approximation (2). The approximation proposed in Lemma 2.1 together with (2) is equivalent to approximating the c_j by the trapezoidal quadrature of the Cauchy integral representation; see [2, 21].

Let us make one final remark here. If

$$d(\zeta) = 1/c(\zeta) = \sum_{j=0}^{\infty} d_j \zeta^j,$$

then $T_L(\boldsymbol{c})^{-1} = T_L(\boldsymbol{d})$. If $c_0 \neq 0$ and $|c_j| \leq p^j$, for some constant p, such an expansion $d(\zeta)$ can always be constructed, e.g. by using $1/(1-w) = \sum_{l=0}^{\infty} w^l$ with $w = -\sum_{j=1}^{\infty}(c_j/c_0)\zeta^j$ and ζ small enough so that $|w| < 1$. Approximating $T_L(\boldsymbol{c})^{-1}$ by $\Lambda^{-1}C(\boldsymbol{c}_\lambda)^{-1}\Lambda$ is then related to the trapezoidal quadrature of the Cauchy integral representation of the d_j.

3. Linear multistep discretization of evolution problems

We consider a time dependent problem of the form

$$\partial_t^l u + \mathcal{L}u = f, \qquad t \in [0, T], \tag{3}$$

where ∂_t^l is the lth partial derivative w.r.t. time, $\mathcal{L}$ is a time independent linear operator, and u, f are functions of time with $u(t), f(t) \in \mathrm{X}$ for some Banach space X and $t \in [0, T]$.

Method	α	β	k	type	order
Implicit Euler (BDF1)	$1-\zeta$	1	1	implicit	1
BDF2	$\frac{3}{2}-2\zeta+\frac{1}{2}\zeta^2$	1	2	implicit	2
Trapezoidal Rule (TR)	$1-\zeta$	$\frac{1}{2}+\frac{1}{2}\zeta$	1	implicit	2
Explicit Euler (EE)	$1-\zeta$	ζ	1	explicit	1
Leapfrog (LF)	$1-\zeta^2$	2ζ	2	explicit	2
BDF3	$\frac{11}{6}-3\zeta+\frac{3}{2}\zeta^2-\frac{1}{3}\zeta^3$	1	3	implicit	3

TABLE 1. Generating polynomials of some popular linear k-step methods.

We discretize (3) in time as follows. Let $\Delta t > 0$, $t_j = j\Delta t$, $j = 0, 1, \ldots, N$, and let a linear k-step method be given by the coefficients of its generating polynomials

$$\alpha(\zeta) = \sum_{j=0}^{k} \alpha_j \zeta^j, \qquad \beta(\zeta) = \sum_{j=0}^{k} \beta_j \zeta^j,$$

we will also make use of the quotient of the generating polynomials

$$\delta(\zeta) := \frac{\alpha(\zeta)}{\beta(\zeta)} = \sum_{j=0}^{\infty} \delta_j \zeta^j.$$

We use the convention that $\alpha_j = \beta_j = 0$ whenever $j > k$. Some well-known examples of such methods are given in Table 1. In this work we find it more convenient to use the ordering of the multistep coefficients which is reversed compared to most of the literature. At this stage, the only condition we make on the linear multistep method is that $\delta(\zeta)$ is analytic and never zero inside the annulus $0 < |\zeta| < 1$. Later some further conditions will be imposed that ensure A or $A(\theta)$-stability.

To discretize higher order derivatives, i.e., $l > 1$ above, we will need also higher powers of generating polynomials for which we will use the following notation:

$$(\alpha(\zeta))^l = \sum_{j=0}^{lk} \alpha_j^{(l)} \zeta^j, \qquad (\beta(\zeta))^l = \sum_{j=0}^{lk} \beta_j^{(l)} \zeta^j.$$

Discretizing (3) in time by the linear multistep discretization transforms (3) to the discrete convolutional system which at time $t_N = N\Delta t$ has the form

$$\frac{1}{(\Delta t)^l} \sum_{j=0}^{N} \alpha_{N-j}^{(l)} u_j + \mathcal{L} \sum_{j=0}^{N} \beta_{N-j}^{(l)} u_j = \sum_{j=0}^{N} \beta_{N-j}^{(l)} f_j + \widetilde{f}_N, \tag{4}$$

where the sub-indices of functions indicate its evaluation at the corresponding time step, i.e., for $u : \mathbb{R} \to X$, $u_j := u(j\Delta t)$, $j \in \mathbb{Z}$. The initial data $u_{-lk}, \ldots, u_{-1}$ and $f_{-lk}, \ldots, f_{-1}$ is assumed to be known and to have been moved to the right-hand side; see the definition of $\tilde{\mathbf{f}}$ in (5).

For the first order time derivative, i.e., $l = 1$, it is clear how the above discrete convolutional system arises after the linear multistep discretization. For higher order

derivatives, one can first construct a first order system equivalent to (3), e.g. for $l = 2$,

$$\partial_t U + \begin{pmatrix} 0 & -\mathcal{I} \\ \mathcal{L} & 0 \end{pmatrix} U = \begin{pmatrix} 0 \\ f \end{pmatrix}, \qquad \text{where } U = \begin{pmatrix} u \\ \partial_t u \end{pmatrix},\ \mathcal{I}u := u, \quad \forall u \in X,$$

discretize it by the linear multistep formula, and then recover the scalar semi-discrete system (4).

Writing (4) in matrix notation we obtain

$$T_L(\boldsymbol{\alpha}^l/\Delta t^l) \otimes \mathcal{I}\boldsymbol{u} + T_L(\boldsymbol{\beta}^l) \otimes \mathcal{L}\boldsymbol{u} = T_L(\boldsymbol{\beta}^l) \otimes \mathcal{I}\boldsymbol{f} + \widetilde{\boldsymbol{f}},$$

where as usual $\boldsymbol{u} = (u_0, u_1, \dots, u_N)^T$, $\boldsymbol{f} = (f_0, f_1, \dots, f_N)^T$, $\boldsymbol{\alpha}^l = (\alpha_0^{(l)}, \alpha_1^{(l)}, \dots, \alpha_N^{(l)})^T$, and $\boldsymbol{\beta}^l = (\beta_0^{(l)}, \beta_1^{(l)}, \dots, \beta_N^{(l)})^T$. Note that here we chose to use tensor notation because u_j can be scalars, but also operators, or, as in the next section, vectors.

The vector $\widetilde{\boldsymbol{f}}$ modifying the right-hand side, is given in matrix notation by

$$\widetilde{\boldsymbol{f}} := T_U(P\boldsymbol{\beta}^l) \otimes \mathcal{I}\boldsymbol{f}_{\text{in}} - T_U(P\boldsymbol{\alpha}^l/\Delta t^l) \otimes \mathcal{I}\boldsymbol{u}_{\text{in}} - T_U(P\boldsymbol{\beta}^l) \otimes \mathcal{L}\boldsymbol{u}_{\text{in}}, \tag{5}$$

where the initial data is given by $\boldsymbol{u}_{\text{in}} = (0, 0, \dots, u_{-lk}, \dots, u_{-1})^T$, similarly for $\boldsymbol{f}_{\text{in}}$. Note that only the first lk elements of the modifying vector $\widetilde{\boldsymbol{f}}$ are non-zero.

For a parameter $0 < \lambda < 1$ we can proceed as in the previous section by approximating the Toeplitz operators by circulant operators to write a perturbed system

$$C(\boldsymbol{\alpha}_\lambda^l/\Delta t^l) \otimes \mathcal{I}\boldsymbol{u}_\lambda + C(\boldsymbol{\beta}_\lambda^l) \otimes \mathcal{L}\boldsymbol{u}_\lambda = C(\boldsymbol{\beta}_\lambda^l) \otimes \mathcal{I}\boldsymbol{f}_\lambda + \widetilde{\boldsymbol{f}}_\lambda, \tag{6}$$

where we use the notation $\boldsymbol{v}_\lambda = \Lambda \otimes \mathcal{I}\boldsymbol{v}$ for $\boldsymbol{v} \in X^{N+1}$. Note that $\widetilde{\boldsymbol{u}} = \Lambda^{-1} \otimes \mathcal{I}\boldsymbol{u}_\lambda$ is then the approximation to $\boldsymbol{u}$. The error $\boldsymbol{e} = \boldsymbol{u} - \widetilde{\boldsymbol{u}}$ satisfies the equation

$$\begin{aligned} &T_L(\boldsymbol{\alpha}^l/\Delta t^l) \otimes \mathcal{I}\boldsymbol{e} + T_L(\boldsymbol{\beta}^l) \otimes \mathcal{L}\boldsymbol{e} = \\ &\qquad \lambda^{N+1} \left(-T_U(P\boldsymbol{\beta}^l) \otimes \mathcal{I}\boldsymbol{f} + T_U(P\boldsymbol{\alpha}^l/\Delta t^l) \otimes \mathcal{I}\widetilde{\boldsymbol{u}} + T_U(P\boldsymbol{\beta}^l) \otimes \mathcal{L}\widetilde{\boldsymbol{u}} \right). \end{aligned} \tag{7}$$

Comparing the right-hand side in the above equation with (5) we realize that the error is a solution of an evolution problem whose initial data are the last lk entries of $-\boldsymbol{f}$ and $-\widetilde{\boldsymbol{u}}$ scaled by λ^{N+1}.

Remark 3.1. *In (7) we can again substitute the circulant matrix approximation of the lower Toeplitz matrices to compute a correction. In this way we obtain an error of order $\mathcal{O}(\lambda^{2N+2})$. The correction can be repeated until satisfactory accuracy is obtained thereby avoiding the accuracy restriction of $\sqrt{\text{eps}}$. For example choosing $\lambda^{N+1} = \text{eps}^{1/3}$ after one correction gives an accuracy of $\text{eps}^{2/3}$; for two corrections the optimal choice would be $\lambda^{N+1} = \text{eps}^{1/4}$. The correction strategy is well known as iterative refinement.*

Final manipulation will be to perform a discrete Fourier transform to obtain a decoupled system of equations

$$\frac{1}{(\Delta t)^l}\Sigma_1 \otimes \mathcal{I}\widehat{\boldsymbol{u}} + \Sigma_2 \otimes \mathcal{L}\widehat{\boldsymbol{u}} = \Sigma_2 \otimes \mathcal{I}\widehat{\boldsymbol{f}} + \widehat{\widetilde{\boldsymbol{f}}}, \tag{8}$$

where $\widehat{\boldsymbol{v}} = \mathcal{F}_{N+1} \otimes \mathcal{I}\boldsymbol{v}_\lambda$. Further, Σ_1 and Σ_2 are diagonal matrices with $(\Sigma_1)_{mm} = (\mathcal{F}_{N+1}\boldsymbol{\alpha}_\lambda^l)_m$ and $(\Sigma_2)_{mm} = (\mathcal{F}_{N+1}\boldsymbol{\beta}_\lambda^l)_m$, $m = 0, 1, \dots, N$. Note that if $N \geq lk$ then

$$(\Sigma_1)_{mm} = \alpha^l(\lambda\zeta_{N+1}^m) \text{ and } (\Sigma_2)_{mm} = \beta^l(\lambda\zeta_{N+1}^m), \quad m = 0, 1, \dots, N, \tag{9}$$

otherwise the above equalities are true up to an error of $\mathcal{O}(\lambda^{N+1})$; see Section 2.1.

Multiplying (8) by $\Sigma_2^{-1}\otimes\mathcal{I}$ we see that we indeed obtain $N+1$ decoupled Helmholtz-like problems of the type

$$\left(\frac{s_m}{\Delta t}\right)^l \widehat{u}_m + \mathcal{L}\widehat{u}_m = \widehat{f}_m, \qquad s_m = \delta(\lambda\zeta_{N+1}^m) = \frac{\alpha(\lambda\zeta_{N+1}^m)}{\beta(\lambda\zeta_{N+1}^m)}, \tag{10}$$

where, to simplify the notation, $\widehat{f}_m = (\widehat{\boldsymbol{f}})_m + \left(\beta^l(\lambda\zeta_{N+1}^m)\right)^{-1}(\widehat{\widehat{\boldsymbol{f}}})_m$.

4. Semi-discretized evolution equation: Method of lines

Let Ω be a bounded domain in $\mathbb{R}^d$, $d=2,3$, with Lipschitz boundary Γ. We consider two model problems:

Heat equation: Find $u(\cdot,t)\in \mathrm{H}^1(\Omega)$ such that

$$(\mathcal{P}) \qquad \begin{aligned} \partial_t u(x,t) - \nabla.\nabla u(x,t) &= f(x,t), && (x,t)\in\Omega\times(0,T),\\ u(\cdot,t) &= u_0(x), && x\in\Omega\\ u|_\Gamma &= g, && (x,t)\in\Gamma\times[0,T], \end{aligned}$$

where $g(\cdot,t)\in \mathrm{H}^{1/2}(\Gamma)$ denotes the given boundary data, $u_0\in \mathrm{H}^1(\Omega)$ the initial condition, and $f(\cdot,t)\in \mathrm{H}^{-1}(\Omega)$ the given forcing term.

Wave equation: Find $u(\cdot,t)\in \mathrm{H}^1(\Omega)$ such that

$$(\mathcal{H}) \qquad \begin{aligned} \partial_t^2 u(x,t) - \nabla.\nabla u(x,t) &= f(x,t), && (x,t)\in\Omega\times(0,T),\\ u(\cdot,t)=u_0(x),\ \partial_t u(\cdot,t) &= v_0(x), && x\in\Omega\\ u|_\Gamma &= g, && (x,t)\in\Gamma\times[0,T], \end{aligned}$$

where $g(\cdot,t)\in H^{1/2}(\Gamma)$ is the given boundary data, $u_0, v_0\in H^1(\Omega)$ the initial conditions, and $f(\cdot,t)\in H^{-1}(\Omega)$ the given forcing term.

We do not assume any smoothness conditions in time or compatibility condition of the initial and boundary data because, in this paper, we are only interested in errors committed by our method of solving the discrete convolutional system of equations by a decoupling procedure. Of course, if we wanted to give the convergence estimates for the full method we would need such assumptions.

Remark 4.1. *At this stage we could proceed as in the previous section and discretize in time and introduce the $\mathcal{O}(\lambda^{N+1})$ perturbation in order to arrive at a decoupled system. Thereby we would obtain $N+1$ decoupled Helmholtz problems:*

$$\begin{aligned} \mathcal{L}\hat{u}_n + \omega_n\hat{u}_n &= \hat{f}_n,\\ \hat{u}_n|_\Gamma &= \hat{g}_n, \end{aligned} \qquad n=0,1,\dots,N. \tag{11}$$

These Helmholtz problems are uniquely solvable if the linear multistep method is $A(\theta)$-stable for $(\mathcal{P})$ and for $(\mathcal{H})$ if it is A-stable; for more details and for explicit methods see the next section.

An approximate solution of (11) can then be computed by a number of standard numerical methods, e.g., finite differences, spectral methods, etc. In this paper we choose to solve the problems using the finite element method (FEM). In order to more easily fit the framework of the previous sections we introduce the space discretization by the FEM first and then apply the time-discretization to the resulting ODE.

4.1. **Finite Element discretization in space.** We next explain the discretization of domain operators by the finite element method. We will denote by $\mathrm{S} \subset \mathrm{H}^1(\Omega)$ a finite element space, by $\mathcal{Y} := \gamma_0 \mathrm{S}$ its restriction to the boundary, and by $\mathrm{X} := \mathrm{S} \cap \mathrm{H}^1_0(\Omega)$ the subspace of finite element functions with zero boundary trace.

Let the continuous sesquilinear forms $a, b, c : \mathrm{H}^1(\Omega) \times \mathrm{H}^1(\Omega) \to \mathbb{C}$ be defined by

$$a(u,v) := \int_\Omega \nabla u \cdot \overline{\nabla v}, \quad b(u,v) := \int_\Omega u\overline{v}, \quad c(u,v) := \int_\Gamma u\overline{v}. \tag{12}$$

We will also make use of the operators corresponding to the first two sesquilinear forms:

$$\mathcal{A}u = a(u,\cdot), \qquad \mathcal{B}u = b(u,\cdot). \tag{13}$$

Note that we can think of these operators as acting on the discrete spaces as well, where we then have $\mathcal{A} : \mathrm{X} \to \mathrm{X}'$ and $\mathcal{B} : \mathrm{X} \to \mathrm{X}$ or $\mathcal{B} : \mathrm{X}' \to \mathrm{X}'$.

4.2. **The fully discrete equations.** Let us assume that $g_j^* \in \mathrm{S}$ are given such that

$$c(g_j^*, v) = c(g_j, v), \qquad \text{for all } v \in \mathrm{S},\ j = 0, 1, \dots, N, \tag{14}$$

where $g_j(\cdot) := g(\cdot, t_j)$, $j = 0, 1, \dots, N$. Then the fully discrete system has the following form: Find $\boldsymbol{u}^* = (u_0^*, u_1^*, \dots, u_N^*)^T$, $u_j^* \in \mathrm{X}$, $j = 0, 1, \dots, N$, such that

$$\begin{aligned} &T_L(\boldsymbol{\alpha}^l/\Delta t^l) \otimes \mathcal{B}\boldsymbol{u}^* + T_L(\boldsymbol{\beta}^l) \otimes \mathcal{A}\boldsymbol{u}^* = \\ &\qquad T_L(\boldsymbol{\beta}^l) \otimes \mathcal{B}\boldsymbol{f} - T_L(\boldsymbol{\alpha}^l/\Delta t^l) \otimes \mathcal{B}\boldsymbol{g}^* - T_L(\boldsymbol{\beta}^l) \otimes \mathcal{A}\boldsymbol{g}^* + \widetilde{\boldsymbol{f}}, \end{aligned} \tag{15}$$

where $\boldsymbol{g}^* = (g_0^*, g_1^*, \dots, g_N^*)^T$. The approximate solution of the original problem, $(\mathcal{P})$ or $(\mathcal{H})$, is given by $\boldsymbol{u} = \boldsymbol{u}^* + \boldsymbol{g}^*$.

We next define the perturbed system with Toeplitz matrices replaced by circulant matrices: Find $\boldsymbol{u}_\lambda^* = (u_{\lambda,0}^*, u_{\lambda,1}^*, \dots, u_{\lambda,N}^*)^T$, $u_{\lambda,j}^* \in \mathrm{X}$, $j = 0, 1, \dots, N$, such that

$$\begin{aligned} &C(\boldsymbol{\alpha}_\lambda^l/\Delta t^l) \otimes \mathcal{B}\boldsymbol{u}_\lambda^* + C(\boldsymbol{\beta}_\lambda^l) \otimes \mathcal{A}\boldsymbol{u}_\lambda^* = \\ &\qquad C(\boldsymbol{\beta}_\lambda^l) \otimes \mathcal{B}\boldsymbol{f}_\lambda - C(\boldsymbol{\alpha}_\lambda^l/\Delta t^l) \otimes \mathcal{B}\boldsymbol{g}_\lambda^* - C(\boldsymbol{\beta}_\lambda^l) \otimes \mathcal{A}\boldsymbol{g}_\lambda^* + \widetilde{\boldsymbol{f}_\lambda}. \end{aligned} \tag{16}$$

As in the previous sections, in the discrete Fourier space, the above system decouples into $N+1$ independent linear problems; these will be investigated in the next subsection.

4.3. **Analysis of the discrete Helmholtz problems.** The system (16) is equivalent to $N+1$ problems of the following type: Find $\hat{u}^* \in \mathrm{X}$ such that

$$\mathcal{A}\hat{u}^* + \omega\mathcal{B}\hat{u}^* = \mathcal{B}\hat{f} - \mathcal{A}\hat{g}^* - \omega\mathcal{B}\hat{g}^*. \tag{17}$$

The discussion in this section applies to all of the $N+1$ problems, therefore we omit the extra index.

Let $\{\mu_j\}$ be the eigenvalues of the discrete Laplacian:

$$\text{There exists } u_j \in \mathrm{X}\setminus\{0\},\ \text{s.t.,} \quad \mathcal{A}u_j + \mu_j\mathcal{B}u_j = 0. \tag{18}$$

If $\omega \notin \{\mu_j\}$ then the equation (17) has a unique solution. It is well known that $\mu_j < 0$ for all j and that $|\mu_j| \le Ch^{-2}$ where h is the diameter of the smallest element in the space discretization. The frequencies ω lie on the curve

$$\omega \in \mathfrak{C}_{\mathcal{P}} := \left\{ \frac{\delta(\lambda\zeta)}{\Delta t} \mid |\zeta| = 1 \right\}$$

for the parabolic model problem and are on

$$\omega \in \mathfrak{C}_{\mathcal{H}} := \left\{ \left(\frac{\delta(\lambda\zeta)}{\Delta t} \right)^2 \mid |\zeta| = 1 \right\}$$

for the hyperbolic problem. Therefore to determine whether the resulting Helmholtz problems are solvable it is important to check whether, and if so, where does the above curve intersect the negative real line.

In the case of $(\mathcal{P})$ and $A(\theta)$-stable linear multistep methods, $0 < \theta \leq \pi/2$, the wavenumbers are always well-separated from the negative real axis because

$$\mathfrak{C}_{\mathcal{P}} \subset \{z \mid z = r\exp(i\varphi), |\varphi| \leq \pi - \theta\}.$$

In the case of $(\mathcal{H})$ and explicit methods the CFL condition ensures that the curve $\mathfrak{C}_{\mathcal{H}}$ misses the eigenvalues of the discrete operator: For example for the leapfrog formula $\delta(\zeta) = \frac{1-\zeta^2}{2\zeta}$, the curve cuts the negative real line at $\frac{(1+\lambda^2)^2}{4\lambda^2\Delta t^2}$; the usual CFL condition $\Delta t = o(h)$ ensures solvability.

For A-stable linear multistep methods it holds that $\operatorname{Re}\delta(\zeta) > 0$ for $|\zeta| < 1$, therefore $\mathfrak{C}_{\mathcal{H}}$ misses the negative real line and (17) has a unique solution for all Δt. For example for the Backward Euler (BDF1), $\delta(\zeta) = 1 - \zeta$, backward difference formula of order 2 (BDF2), $\delta(\zeta) = \frac{3}{2} - 2\zeta + \frac{1}{2}\zeta^2$, and the second order trapezoidal rule, $\delta(\zeta) = 2(1-\zeta)/(1+\zeta)$, we have that

$$\operatorname{Re}\delta(\lambda\zeta) \geq (1-\lambda)/\Delta t > 0. \tag{19}$$

Lemma 4.2. *Let $\delta(\zeta)$ define a linear k-step method and let the number of time steps N be such that $N \geq lk$, where $l = 1$ for $(\mathcal{P})$ and $l = 2$ for $(\mathcal{H})$. If*

$$\mu_j \notin \mathfrak{C}_{\mathcal{P}}, \text{ for } (\mathcal{P}), \text{resp. } \mu_j \notin \mathfrak{C}_{\mathcal{H}}, \text{ for } (\mathcal{H}), \quad j = 0, 1, \ldots, \tag{20}$$

where μ_j are the eigenvalues of the discrete Laplacian as defined in (18), then the solution $\boldsymbol{u}_\lambda^*$ of (16) exists and is unique.

In particular the condition (20) is satisfied if the underlying linear multistep method is $A(\theta)$-stable, for $(\mathcal{P})$, resp. if it is A-stable for $(\mathcal{H})$.

Finally we can use the results from the first few sections to provide an expression for the error.

Theorem 4.3. *Let $\boldsymbol{u}^*$ and $\boldsymbol{u}_\lambda^*$ be the solutions of (15) respectively (16) and let $\boldsymbol{u} = \boldsymbol{u}^* + \boldsymbol{g}^*$, $\boldsymbol{u}_\lambda = \boldsymbol{u}_\lambda^* + \boldsymbol{g}_\lambda^*$, and $\widetilde{\boldsymbol{u}} = \Lambda^{-1} \otimes \mathcal{I}\boldsymbol{u}_\lambda$. The error $\boldsymbol{e} = \boldsymbol{u} - \widetilde{\boldsymbol{u}}$ is then the solution of the discrete evolution problem*

$$T_L(\boldsymbol{\alpha}^l/\Delta t^l) \otimes \mathcal{B}\boldsymbol{e} + T_L(\boldsymbol{\beta}^l) \otimes \mathcal{A}\boldsymbol{e} = \lambda^{N+1} \left(T_U(P\boldsymbol{\alpha}^l/\Delta t^l) \otimes \mathcal{B}\widetilde{\boldsymbol{u}} + T_U(\boldsymbol{\beta}^l) \otimes \mathcal{A}\widetilde{\boldsymbol{u}} - T_U(\boldsymbol{\beta}^l) \otimes \mathcal{B}\boldsymbol{f} \right).$$

Proof. Using the results from the previous section, see (7), we immediately obtain that $\boldsymbol{e}^* = \boldsymbol{u}^* - \Lambda^{-1} \otimes \mathcal{I}\boldsymbol{u}_\lambda^*$ satisfies

$$T_L(\boldsymbol{\alpha}^l/\Delta t^l) \otimes \mathcal{B}\boldsymbol{e}^* + T_L(\boldsymbol{\beta}^l) \otimes \mathcal{A}\boldsymbol{e}^* = \lambda^{N+1} \left(T_U(P\boldsymbol{\alpha}^l/\Delta t^l) \otimes \mathcal{B}(\boldsymbol{g}^* + \widetilde{\boldsymbol{u}}^*) + T_U(P\boldsymbol{\beta}^l) \otimes \mathcal{A}(\boldsymbol{g}^* + \widetilde{\boldsymbol{u}}^*) - T_U(P\boldsymbol{\beta}^l) \otimes \mathcal{B}\boldsymbol{f} \right),$$

where $\widetilde{\boldsymbol{u}}^* = \Lambda^{-1} \otimes \mathcal{I}\boldsymbol{u}^*_\lambda$. The result follows from the identities $\boldsymbol{u} = \boldsymbol{u}^* + \boldsymbol{g}^*$ and $\widetilde{\boldsymbol{u}} = \widetilde{\boldsymbol{u}^*} + \boldsymbol{g}^*$. □

The importance of the above result is that it shows that the error is the solution of a discrete evolution problem scaled by λ^{N+1}, or equivalently, the solution of the problem with initial data of size $\mathcal{O}(\lambda^{N+1})$. This in turn allows us to easily compute a correction to the solution; see Remark 3.1. Note that, discretization errors, coming from, e.g., quadrature, are not aggravated by multiplication with the ill-conditioned matrix Λ^{-1}. However, errors in the solution of the Helmholtz problems can be.

5. Some extensions and generalizations

In this paper we consider the heat and wave equations and linear multistep discretization. Extensions to some other linear evolution problems are straightforward. For example to implement the dissipative wave equation $\partial_t^2 u + k\partial_t u - \Delta u = f$ the only change to the code would be to substitute the wave number $\delta(\zeta)/\Delta t$ in (10) by the dissipative equivalent $\sqrt{(\delta(\zeta)/\Delta t)^2 + k\delta(\zeta)/\Delta t}$.

Further, other time discretization methods based on equally spaced time steps produce lower triangular Toeplitz semi discrete-systems to which the decoupling procedure can also be applied. In particular Runge-Kutta methods are an important alternative to linear multistep methods, especially for hyperbolic problems, because A-stable Runge-Kutta methods of higher orders than 2 are readily available. In this case details of the implementation are somewhat more technical, for an m-stage Runge-Kutta method the wave numbers become $m\times m$-matrices, requiring the solution of m Helmholtz problems at the frequencies given by the eigenvalues of the $m \times m$-matrix. For a similar perspective on Runge-Kutta methods for the wave equation, but in the context of time domain boundary integral operators, see [1].

6. Algorithmic realization and numerical experiments

In this section we will discuss the practical realization of the proposed parallel time discretization approach and compare it to the classical time stepping procedures which correspond to forward elimination of the lower triangular Toeplitz system. The implementation of both methods is almost trivial. Nevertheless, Algorithm 1 gives some pseudo-code for a model situation in order to extract the differences. In the algorithm description we assume that elements of $\boldsymbol{u} \in X^{N+1}$ are stored as $M \times (N+1)$ matrices and we denote by $\mathbf{u}_j$ the jth column of $\mathbf{u}$ and by $(\mathbf{u}^T)_j$ the jth row of $\mathbf{u}$. To simplify the presentation we consider the fully discrete system (15) with homogeneous initial and boundary data written as a discrete convolutional system: Find $\boldsymbol{u} \in \mathrm{X}^{N+1}$, such that

$$\frac{1}{\Delta t^l}\sum_{j=0}^{n} \alpha^{(l)}_{n-j}\mathcal{B}u_j + \sum_{j=0}^{n}\beta^{(l)}_{n-j}\mathcal{A}u_j = \sum_{j=0}^{n}\beta^{(l)}_{n-j}\mathcal{B}f_j, \quad n = 0, 1, \dots, N.$$

In what follows we will always use M to denote the dimension of the space X.

Assuming the linear systems can be solved in nearly $\mathcal{O}(M)$ time, the overall complexity of Algorithm 1(a) is $\mathcal{O}(kNM)$. In the decoupled version 1(b) we get $\mathcal{O}(NM \log N)$ (neglecting parallelism for the moment). The dependence on the step number k disappears at the price of an additional logarithmic factor. Consequently, the decoupling

Algorithm 1 Classical k-step algorithm and decoupled multistep method.

Require: Data: $\mathbf{f} \in \mathbb{R}^{M\times(N+1)}$, Generating polynomials: α, β, Parameters: Δt, ε.

(a) Time stepping loop

for $n = 0 : N$ **do**
  $\boldsymbol{r} := \beta_0^{(l)} \mathcal{B} f_n$;
  for $j = 1 : k$ **do**
    $\boldsymbol{r} := \boldsymbol{r} + \beta_j^{(l)}(\mathcal{B} f_{n-j} - \mathcal{A} u_{n-j}) - \frac{\alpha_j^{(l)}}{\Delta t^l}\mathcal{B} u_{n-j}$;
  end for
  Solve $(\beta_0 \mathcal{A} + \frac{\alpha_0^{(l)}}{\Delta t^l}\mathcal{B}) u_n = \boldsymbol{r}$;
end for

(b) Parallelized

$\lambda := \varepsilon^{1/N}$;
$\Lambda := \mathrm{diag}(1, \lambda, \lambda^2, \ldots, \lambda^N)$;
$z := \lambda \exp(2\pi \mathrm{i}/(N+1))$;
parfor $j = 1 : M$ **do**
  $(\hat{\mathbf{f}}^T)_j := \mathcal{F}_{N+1}((\Lambda \mathbf{f}^T)_j)$;
end for
parfor $n = 0 : N$ **do**
  Solve $(\beta^l(z^n)\mathcal{A} + \frac{\alpha^l(z^n)}{\Delta t^l}\mathcal{B})\hat{\mathbf{u}}_n = \beta^l(z^n)\hat{\mathbf{f}}_n$;
end for
parfor $j = 1 : M$ **do**
  $(\mathbf{u}^T)_j := \Lambda^{-1}\mathcal{F}_{N+1}^{-1}((\hat{\mathbf{u}}^T)_j)$;
end for

return u.

provides huge savings of computational time compared to forward elimination if the Toeplitz system is densely populated. However, in the context of linear multistep methods where we are confronted with band limited systems the effort for both algorithms is comparable in a sequential computational environment.

The discussion of storage complexity depends strongly on the needs of the underlying application. If the solution is required at all time steps, then both algorithms belong to the class $\mathcal{O}(MN)$. If only the solution at final time is of interest then the time stepping strategy is superior because it requires only $\mathcal{O}(kM)$ data to be stored simultaneously while Algorithm 1(b) needs access to the data at all time steps at once.

A key property of modern algorithms is parallelism, which is the greatest advantage of the decoupling procedure. Indicated by the use of keyword parfor, the loops in Algorithm 1(b) can be executed in parallel. While the loops computing the discrete Fourier transformations can be parallelized with respect to the space variables, the loop for the solution of the decoupled systems is parallel with respect to the time discretization leading to $\mathcal{O}(\frac{MN\log N}{P})$ time complexity if $P \leq \min\{M, N\}$ processors are available. Since within each loop *no* communication between the individual processes is necessary, algorithm 1(b) is embarrassingly parallel. All internodal communication is concentrated between the loops where the data needs to be redistributed. This allows its simple use of distributed computing networks. On the contrary, in the time-stepping algorithm 1(a), parallelization is restricted to the individual time steps, i.e., to vector and matrix operations. The biggest drawback of such a parallelization is that communication between the working units is necessary throughout the computation. Therefore optimal upscaling with respect to the number of parallel machines is hard to achieve for time stepping in practice.

There is a difference in the linear systems to be solved in the algorithms. While in the time-stepping algorithm real symmetric positive definite systems have to be solved, we are confronted with complex and possibly indefinite Helmholtz problems in the decoupled case. The disadvantage of Algorithm 1(b) of requiring complex arithmetic is alleviated by the fact that only half of the systems need to be solved since the linear systems occur in complex-conjugate pairs. The availability of fast solvers for the indefinite linear systems will be discussed in Section 6.2.

An important aspect of the parallel algorithm is that it can easily be combined with the time-stepping algorithm. That is, we suggest a block time-stepping algorithm where triangular blocks of size P are solved by the parallel Fourier techniques. The optimal choice of P would be governed by the number of nodes in the parallel cluster, the available memory, and the difficulty of solving the Helmholtz problems. The latter point is due to the fact, explained in the coming sections, that generally the Helmholtz problems become more difficult to solve for increasing P.

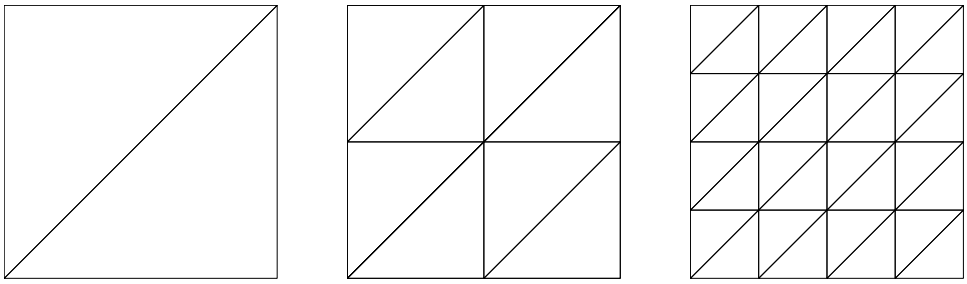
6.1. **Numerical experiments.** We continue the discussion with some numerical experiments aiming for the practical investigation of the error bound from Theorem 4.3 for some of the implicit linear multistep time discretizations listed in Table 1. All the computations are done in Matlab. For most of the experiments, we restrict ourselves to the more challenging case of the wave equation and consider two simple model situations that allow the computation of exact errors. Let

$$w_{\boldsymbol{a},b} : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}, \qquad \boldsymbol{x} \mapsto \cos(b(t - \langle \boldsymbol{a}, \boldsymbol{x}\rangle)e^{-10(t-\langle \boldsymbol{a},\boldsymbol{x}\rangle-3)^2}, \tag{21}$$

where $\boldsymbol{a} \in \mathbb{R}^d$, $\|\boldsymbol{a}\| = 1$, $b \in \mathbb{R}$. It is easy to see that $w_{\boldsymbol{a},b}$ fulfills the homogeneous wave equation. Moreover, every linear combination of these functions is a solution to the wave equation which will be exploited in the first model problem. Let $T = 6$, $\Omega = [-0.5, 0.5]^2$. Consider

$$\begin{aligned} \partial_t^2 u(x,t) - \Delta u(x,t) &= 0, && (x,t) \in \Omega \times [0,T], \\ u(x,0) &= 0, && x \in \Omega, \\ \partial_t u(x,0) &= 0, && x \in \Omega, \\ u(x,t) &= w_{[1,0],1}(x) + w_{[-1,1],2}(x) && x \in \partial\Omega. \end{aligned} \tag{22}$$

The results of the related computations are listed in Table 2. They are based on a conforming P1 Finite Element space discretization with respect to uniform refinements of the initial triangulation depicted in Figure 1. The level parameter ref $= 2, 3, \dots, 8$ denotes the number of uniform refinements applied to $\mathcal{T}_0$. It further denotes the number of uniform refinements of the time interval $[0, 6]$. The arising linear systems are solved by preconditioned GMRES to accuracy 10^{-8} as described in the subsequent section. Note that, the initial time discretizations are far too coarse to resolve the solution. This explains the slow convergence at the lower levels. As expected the trapezoidal rule performs best. For the BDF2 it takes quite a while to reach the predicted second-order convergence. For both methods the difference between the time stepping and the decoupling result is of order $\varepsilon = 10^{-6}$ which matches the prediction since we chose the optimal value for the parameter $\lambda = \varepsilon^{\frac{1}{N}}$, where $N = 2^{\mathrm{ref}}$ denotes the number of time steps.



(a) $\mathcal{T}_0$. (b) $\mathcal{T}_1$. (c) $\mathcal{T}_2$.

FIGURE 1. Initial triangulation $\mathcal{T}_0$ of $[-0.5, 0.5]^2$ and uniform refinements $\mathcal{T}_{\text{ref}}$, ref $= 1, 2$.

For hyperbolic problems we are restricted to the use of A-stable or explicit methods with sufficiently fine time discretization. Nevertheless, some accuracy can be obtained by using $A(\theta)$-stable methods and our decoupling procedure gives an interesting viewpoint on the development of instabilities. Next, we have a closer look at the behavior of simulations based on the BDF3 time discretization. Let $T = 10$, $\Omega = [0, 1]^2$. Consider

$$\begin{aligned} \partial_t^2 u(x,t) - \Delta u(x,t) &= 0, \qquad (x,t) \in \Omega \times [0,T], \\ u(x,0) &= \sin(2\pi x), \qquad x \in \Omega, \\ \partial_t u(x,0) &= 0, \qquad x \in \Omega, \\ u(x,t) &= 0, \qquad x \in \partial\Omega. \end{aligned} \tag{23}$$

We use the same space discretization as before but, due to the larger time interval, we use slightly larger time steps. If h indicates the mesh width of $\mathcal{T}_{\text{ref}}$ and Δt the time step, then $\Delta t = 10h$ is chosen on all levels. The BDF3 method is not A-stable and therefore expected to fail after a certain number of time steps. This situation has not been reached after 256 time steps; see Table 3(b). This is theoretically justified by the fact that the related curve $\mathfrak{C}_{\mathcal{H}} := \left\{ \left(\alpha_{\text{BDF3}}(10^{-6/N}\zeta)\right)^2 / \Delta t^2 \mid |\zeta| = 1 \right\}$, for $N = 256$, does not intersect the negative real axis. Doubling the number of time steps makes the above curve cross the negative real axis and (keeping space resolution) gives $\|e^{ts}\|_{L^\infty([0,T],L^2(\Omega))} = 16.51$ and $\|e^{par}\|_{L^\infty([0,T],L^2(\Omega))} = 0.389$. Decreasing λ would again make the curve avoid the negative real axis, see Figure 2, but would make the matrix Λ^{-1} more singular thereby creating numerical difficulties.

6.2. **A few remarks on the efficient solution of the linear Helmholtz-type systems.** The application of the parallel Algorithm 1(b) requires the solution of discrete Helmholtz problems of the form

$$H(\omega) := Au + \omega Bu = f, \quad A, B \in \mathbb{R}^{M\times M},\ f \in \mathbb{C}^M,\ \omega \in \mathfrak{C}(\lambda) \subset \mathbb{C}. \tag{24}$$

where $\mathfrak{C}(\lambda) = \mathfrak{C}_{\mathcal{P}}$ or $\mathfrak{C}(\lambda) = \mathfrak{C}_{\mathcal{H}}$. We have highlighted the dependence on the parameter $\lambda \in (0, 1)$ explicitly. In Figure 2 such curves are depicted for several implicit and explicit linear multistep methods.

ref	$\|e^{ts}_{\mathrm{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	$\|e^{par}_{\mathrm{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	difference	rate
2	0.52449640	0.52449633	4.1e-07	1.05
3	1.08740731	1.08740752	1.1e-06	-1.05
4	0.91896229	0.91896275	9.1e-07	0.23
5	0.52220239	0.52220257	5.0e-07	0.82
6	0.15753554	0.15753554	1.8e-07	1.70
7	0.04117602	0.04117600	7.6e-08	1.96
8	0.01061508	0.01061506	1.1e-07	2.03

(a) Trapezoidal rule.

ref	$\|e^{ts}_{\mathrm{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	$\|e^{par}_{\mathrm{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	difference	rate
2	0.41693631	0.41693631	1.9e-08	-
3	0.54310906	0.54310906	8.5e-08	-0.40
4	0.64303205	0.64303206	1.5e-07	-0.25
5	0.62529911	0.62529916	2.4e-07	0.02
6	0.36945373	0.36945381	2.2e-07	0.77
7	0.14528735	0.14528733	1.3e-07	1.40
8	0.04089906	0.04089903	7.2e-08	1.86

(b) BDF2.

TABLE 2. Convergence history for model problem (22): e^{ts}_{ref} and e^{par}_{ref} denote the errors of Algorithm 1(a) and (b) respectively on refinement level ref; the difference column gives relative distances between the two approximations measured in the $\|\cdot\|_{L^\infty([0,T],L^2(\Omega))}$ norm.

In contrast to the linear systems arising from time-stepping algorithms problem (24) is in general indefinite. The availability of efficient solvers strongly depends on the shape of the curves $\mathfrak{C}(\lambda)$, but not as much on the scaling, i.e., on Δt, as we will see later. If the problem is positive definite we may use multigrid techniques to solve (24). It is known that the linear problems (24) can be solved by means of multigrid or multigrid preconditioned iterative solvers in optimal time complexity $\mathcal{O}(M)$ provided $\operatorname{Re}\omega \geq 0$ and $\operatorname{Im}\omega = 0$; see [17]. We could not find corresponding multigrid convergence results for the case $\operatorname{Re}\omega \geq 0$ and $\operatorname{Im}\omega \neq 0$, but in experiments the same kind of behavior is seen.

We will now briefly discuss the availability of efficient solvers for indefinite Helmholtz problems that appear whenever the curves related to the linear multistep method intersect the left half complex plane. This will happen using $A(\theta)$-stable methods for parabolic problems and A-stable methods for hyperbolic problems for values of λ close to one. Let us recall that the choice of the parameter $\lambda = \varepsilon^{\frac{1}{N}} > \sqrt{\mathrm{eps}}^{\frac{1}{N}}$ strongly depends on N as well as the number of time steps performed in parallel; see Remark 2.2. The parameter λ tends to 1 as $N \to \infty$.

ref	$\|e^{ts}_{\text{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	$\|e^{par}_{\text{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	difference	rate
2	0.48881127	0.48881127	9.8e-10	-
3	0.48783902	0.48783902	4.4e-09	0
4	0.91685161	0.91685161	2.0e-07	-0.94
5	0.75215079	0.75215078	2.8e-07	0.29
6	0.69648890	0.69648886	4.8e-07	-0.10
7	0.59814940	0.59814966	1.1e-06	0.22
8	0.20057960	0.20057953	2.6e-06	1.56

(a) BDF2.

ref	$\|e^{ts}_{\text{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	$\|e^{par}_{\text{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	difference	rate
2	0.51198670	0.51198670	3.0e-09	-
3	0.49146349	0.49146349	7.43-09	0.06
4	1.14187459	1.14187447	2.2e-07	-1.21
5	3.12117657	3.12118521	1.1e-05	-1.45
6	2.71643601	2.71644405	1.2e-05	0.20
7	0.26507886	0.26507961	5.1e-06	3.38
8	0.02912403	0.02911598	2.3e-04	3.16

(b) BDF3.

TABLE 3. Convergence history for model problem (23): e^{ts}_{ref} and e^{par}_{ref} denote the errors of Algorithm 1(a) and (b) respectively on level ref; the difference column gives relative distances between the approximations measured in the $\|\cdot\|_{L^\infty([0,T],L^2(\Omega))}$ norm.

The matrices A and B from (24) are matrix representations of the operators $\mathcal{A}$ and $\mathcal{B}$ defined in (13). While A is positive semi definite, B is assumed to be positive definite. In the setting of the previous section A and B are also symmetric and sparse.

Some linear system solvers are not affected by the indefiniteness of $H(\omega)$, e.g. sparse direct solvers will be equally fast in all cases since the sparsity pattern of the matrix is independent of the wavenumber; we refer to the textbook [4] for an overview on fast direct solvers for sparse linear systems. Furthermore, so called Fast Poisson Solvers (see [23, Section 3.5] for an introduction) can be applied if the underlying domain has a tensorized structure, the appearing coefficients are constant, and the $H(\omega)$ is based on a finite difference discretization or finite elements on uniform grids. In the latter situation the eigenvalues and eigenvectors of $H(\omega)$ are known explicitly and can be exploited to solve (24) by means of fast Fourier transforms in almost optimal complexity $\mathcal{O}(M \log M)$. In the described situation the method of cyclic reduction (see [12]) is a further option.

In more general situations we want to make use of iterative solvers. However, their use is not straightforward compared to the case of positive definite systems. There are some multigrid techniques available for indefinite systems (see e.g. [5, 7, 6]). The key problem is that the spectral properties of the operator changes with respect to the refinement level. This demands either the coarsest level to resolve the frequency or the careful use of smoothing operators depending on the actual refinement level.

The GMRES (G*eneralized* M*inimal* Res*idual*) method introduced in [20] is one of the most effective methods for solving large sparse systems of equations if equipped with a suitable preconditioner. Starting from an initial guess $x^0 \in \mathbb{C}^M$ and the associated residual $r^0 := f - Hx^0$, the algorithm computes a sequence of iterates $x^1, x^2, \dots$ such that the $m-$th residual $r^m := f - Hx^m$ satisfies

$$\|r^m\|_2 = \min_{p\in\mathbb{P}_m : p(0)=1} \|p(H)r^0\|_2, \quad m = 0, 1, \dots, M-1, \tag{25}$$

where $\|\cdot\|_2$ denotes the Euclidean norm in $\mathbb{C}^M$. If H is diagonalizable, i.e., there exists an invertible matrix $X \in \mathbb{C}^{M\times M}$ and a diagonal matrix $D = \operatorname{diag}(d)$, $d \in \mathbb{C}^M$, such that $H = XDX^{-1}$, then $p(H) = Xp(D)X^{-1}$ and the residual has been proved in [20, Proposition 4] to be bounded as follows:

$$\|r^m\|_2 \le \kappa(X) \underbrace{\left(\min_{p\in\mathbb{P}_m : p(0)=1} \max_{j=1,\dots,M} |p(d_j)| \right)}_{=:\varepsilon_m} \|r^0\|_2. \tag{26}$$

Here, $\kappa(X) := \|X\|\|X^{-1}\|$ denotes the condition number of X. Under the assumption that all eigenvalues of H are contained in a ball $\mathfrak{B}_R(q)$ with radius $R > 0$ centered at $q \in \mathbb{C} \setminus \{0\}$ Theorem 5 in [20] gives a bound on ε_m

$$\varepsilon_m \le \left(\min_{p\in\mathbb{P}_m : p(0)=1} \max_{z\in\mathfrak{B}_R(q)} |p(z)| \right) = \left(\frac{R}{|q|} \right)^m. \tag{27}$$

The equality in (27) is known as the Zorantonellos's lemma [18]. If $\frac{R}{|q|} \le c < 1$, then the GMRES iteration will produce a decreasing sequence of residuals that fulfill $\|r^m\| < tol$ for a given tolerance $tol > 0$ as soon as $m > \frac{\log(tol/\kappa(X))}{\log(c)}$.

In the following we will denote the generalized spectrum of A and B by $\sigma_B(A)$, i.e., $\mu \in \sigma_B(A)$ if there exists non-zero $u \in \mathbb{R}^M$ such that

$$Au = \mu Bu. \tag{28}$$

Then $(\omega + \mu, u) \in \mathbb{C} \times \mathbb{R}^M$ is a generalized eigenpair of

$$H(\omega) := A + \omega B$$

and problem (24) is uniquely solvable if and only if $\omega \neq -\mu$ for all solutions μ of (28). The spectrum of matrices $H(\omega)$ from (24) is in general not clustered in a ball that does not contain zero and the GMRES convergence bound (27) is not applicable.

A suitable preconditioner for the problem at hand is the shifted Laplacian preconditioner which has been introduced and analyzed in [10, 8, 9, 24]. The shifted Laplacian preconditioner is simply given by $P(z) = A + zB$, where the shift z is chosen in such a way that $P(z)$ can be inverted efficiently by means of multigrid techniques. The eigenvalues of the preconditioned operator $P(z)^{-1}H(\omega)$ are given by

$$\frac{\mu + \omega}{\mu + z}, \qquad \mu \in \mathbb{R}_{\ge 0} : \ Au = \mu Bu.$$

If we consider the related Möbius transform

$$M : \mathbb{C} \to \mathbb{C}, \quad M(\xi) := \frac{\xi + \omega}{\xi + z}, \tag{29}$$

then for the spectrum of the preconditioned operator it holds that

$$\sigma(P(z)^{-1}H(\omega)) \subset M(\mathbb{R}_{\geq 0}) \subset M(\mathbb{R}).$$

We refer to [16, Chapter 3.V] for a detailed description of Möbius transforms. The most important property of such a transform for our purposes is the preservation of circles including lines regarded as circles with infinite radius. As a consequence we can state that the eigenvalues of the preconditioned operator lie on the circle centered at $q_M := \frac{\omega-\bar{z}}{z-\bar{z}}$ with radius $R_M := \left|\frac{z-\omega}{z-\bar{z}}\right|$. Therefore GMRES convergence for the preconditioned system

$$P(z)^{-1}H(\omega)u = (A+zB)^{-1}(A+\omega B)u = P(z)^{-1}f. \tag{30}$$

can be bounded as follows

$$\varepsilon_m \leq \left(\frac{|\omega - z|}{|\omega - \bar{z}|}\right)^m.$$

This result recovers the convergence bound of the GMRES iteration for the preconditioned system (30) given in [24, Equation (32)].

It is left to determine the shift parameter $z = z(\omega)$ in such a way that the rate is minimized under the constraint that $P(z)$ can be inverted efficiently. In the case of positive or negative definite $H(\omega)$ ($\mathrm{Re}\ \omega \geq 0$, $\mathrm{Re}\ \omega < -\max(\sigma_B(A))$), we can choose $z = \omega$ since standard multigrid techniques will be efficient [5]. The critical setting is, that $H(\omega)$ is indefinite implying that $-\max(\sigma_B(A)) \leq \mathrm{Re}\ \omega < 0$. Assuming that $\mathrm{Im}\ \omega \neq 0$, i.e. $0 \notin M(\mathbb{R})$, we minimize the convergence bound

$$\min g_\omega(z) = \min \frac{|\omega - z|^2}{|\omega - \bar{z}|^2} \quad \text{subject to} \quad z \in \mathbb{C}_+ := \{z \in \mathbb{C} \ : \ \mathrm{Re}\ z \geq 0\}.$$

The constraint $z \in \mathbb{C}_+$ ensures that the preconditioner can be inverted efficiently by standard multigrid techniques. Obviously, if $\omega \in \mathbb{C}_+$, i.e. ω is admissible, the optimal shift

$$z_{\mathrm{opt}}(\omega) = \omega \in \mathbb{C}_+,$$

meaning that $P(z(\omega)) = H(\omega)^{-1}$. If $\omega \notin \mathbb{C}_+$, i.e. ω is not admissible, it is easy to see that

$$\frac{\partial g_\omega(z)}{\partial\, \mathrm{Re}\ z} \geq 0, \quad \forall z \in \mathbb{C}_+.$$

The minimum is attained at some point on the boundary of $\mathbb{C}_+$, that is to say on the imaginary axis. The remaining 1-dimensional minimization problem with respect to the imaginary part of the shift

$$\min_{z \in \partial\mathbb{C}_+} g_\omega(z) = \min_{s \in \mathbb{R}} g_\omega(0 + s \cdot i)$$

results in the following choice of z_{opt}

$$z_{\mathrm{opt}}(\omega) = \mathrm{sign}(\mathrm{Im}\ \omega)|\omega| i, \quad \omega \notin \mathbb{C}_+.$$

Thus, if $\omega \notin \mathbb{C}_+$, the convergence rate of GMRES applied to the preconditioned system

$$P(z_{\mathrm{opt}}(\omega))^{-1}H(\omega)u = P(z_{\mathrm{opt}}(\omega))^{-1}f \tag{31}$$

can be bounded by $\sqrt{g_\omega(z_{\text{opt}})}$, i.e. by

$$\sqrt{\frac{1-\sin(\theta_\omega)}{1+\sin(\theta_\omega)}}, \quad \text{where } \theta_\omega := \arcsin\left(\frac{|\operatorname{Im}\omega|}{|\omega|}\right) \in (0, \tfrac{\pi}{2}]. \tag{32}$$

Therefore the systems (24) arising from the decoupling procedure can be solved efficiently as long as θ_ω remains moderately bounded from below for $\omega \in \mathfrak{C}(\lambda)$.

Considering $A(\theta)$-stable linear multistep methods for parabolic problems (see Figure 2(c,e,g)) the minimal value of θ_ω is bounded below by the θ parameter implying that the convergence rate can be bounded uniformly and independently of all other parameters, in particular Δt, h, and the number of time steps to be computed in parallel.

For A-stable linear multistep methods for hyperbolic problems (see Figure 2(d,f)) the situation is more complicated and we have to consider two regimes. The first regime is that of keeping the product $N\Delta t$ constant and letting N increase. For this case the convergence bound (32) tends to one as the number of time steps N increases. The second regime is that of fixing the time step Δt and then seeing how many steps can efficiently be performed in parallel.

Regime 1 ($N\Delta t = \text{const}$, $N \to \infty$): Here, see (19), we have that

$$\operatorname{Re}\frac{\delta(\lambda\zeta)}{\Delta t} \geq \operatorname{Re}\frac{1-10^{-6/N}}{T/N} \geq \frac{1}{T} =: \sigma_0.$$

Writing $x + \mathrm{i}y = \frac{\delta(\lambda\zeta)}{\Delta t}$, x is smallest, i.e. closest to σ_0, for $y = 0$. For the pth order BDF method x increases for increasing $|y|$ at the rate $c|y|^{p+1}$; the same statement can be made for a number of A-stable Runge-Kutta method, e.g., Radau IIA methods, only here we would be looking at the spectrum of the small matrices $\delta(\zeta)/\Delta t$. From these arguments we see that the critical values for ω in (24) are on the curve

$$\omega \in \{z^2 \mid z = \sigma_0 + \mathrm{i}y, |y| \leq \text{const} \cdot \Delta t^{-p/(p+1)}\}.$$

For this curve, the theory in this section would predict that the number of iterations needed for convergence of GMRES would increase as $N^{p/(p+1)}$ with increasing N, i.e., decreasing Δt. Numerical experiments for BDF1 and BDF2 (see Figure 5) show that this convergence analysis is sharp.

Regime 2 ($\Delta t = \text{const}$, $T \to \infty$): In this regime we fix the time-step appropriate for the particular problem we are solving and increase the number of time steps. In this regime the critical curve is $\omega \in \{z^2 \mid z = \delta(\zeta)/\Delta t, |\zeta| = 1\}$. In this case the number of iterations will be bounded for all A-stable methods that satisfy $\operatorname{Re}\delta(\zeta) > 0$ for $\zeta \neq 1$; this is the case for BDF methods. The trapezoidal rule does not satisfy this condition and we expect an increase in the iteration count as the number of time-steps is increased. We will investigate this case more carefully by numerical experiments described below.

We have also investigated the behavior of the multigrid solver by comparing the behavior of the exact preconditioner with the approximate preconditioner computed by one multigrid V-cycle with 2 Jacobi pre-/post-smoothings on each level. The results in Figure 4 tell us that typically one multigrid V-cycle is enough as it has also been observed in [24, 9, 8, 10, 19, 11]. Note that the gap between the iteration counts of the exact and the multigrid preconditioner could be reduced by using more sophisticated smoothers of Gauss-Seidel type instead of the Jacobi iteration. Nevertheless, for hyperbolic problems,

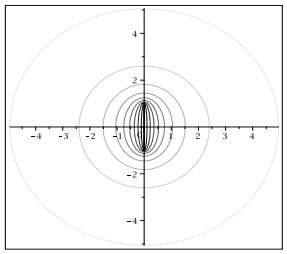
as we will see below, the main limitation to the number of time steps to be performed in parallel comes from the loss in the efficiency of the outer Krylov solver rather than the non-optimal performance of the smoother.

We now present the numerical experiments for the *Regime 2* described above. For different numbers of time steps we iteratively (GMRES with inexact multigrid-based shifted Laplacian preconditioner; parameters as in Figure 4) solve the discrete Helmholtz problems appearing in step 8 of Algorithm 1(b) for both, the heat and the wave equation, as well as several linear multistep methods from Table 1. We thereby fix the time step $\Delta t = 0.025$ and the tolerance $\varepsilon = 1e-6$. We use a conforming P1-FEM with respect to the mesh hierarchy $\{\mathcal{T}_{\text{ref}}\}_{\text{ref}=2}^{8}$ (see Figure 1) as before; the right hand side data is chosen randomly. Figure 6 plots maximal and mean iteration counts versus the number of time steps computed in parallel. For the heat equation we observe that the maximal number of iterations is limited independently from the number of time steps. As it has been proved before there is only a dependence on the order of the linear multistep method. This dependence reflects the decrease of the stability angle. The mean number of iterations is almost independent of the number of time steps and the choice of the linear multistep method. The results are only slightly worse, but still bounded, for the hyperbolic problem and A-stable BDF methods. However, as expected, the maximal number of iterations is not bounded for the Trapezoidal rule; see Figure 6(b). The increase seems to be linear with the number of time step computed in parallel.

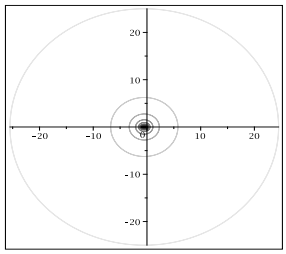
From these results we conclude that even for situations where there is a limit on the number of time-steps computed in parallel, this limit is not severe. The only case for which we have more reservations is the Trapezoidal discretization of the wave equation, where the lack of L-stability creates extra difficulties.

Acknowledgments

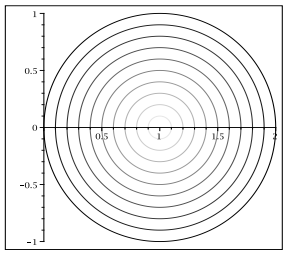
The authors gratefully acknowledge the helpful suggestions made by the anonymous referees, which greatly improved the presentation of the paper. Further, we thank Steffen Börm for pointing out useful references on multigrid methods.



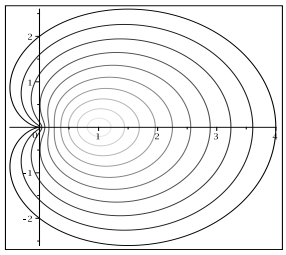
(a) Leapfrog.



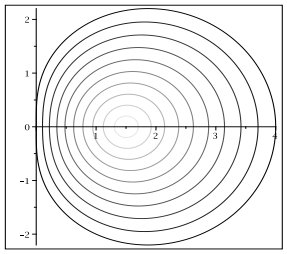
(b) Leapfrog.



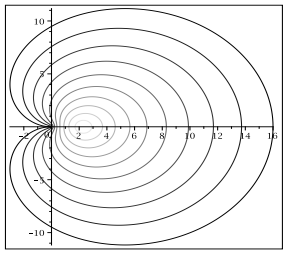
(c) BDF1 (Implicit Euler).



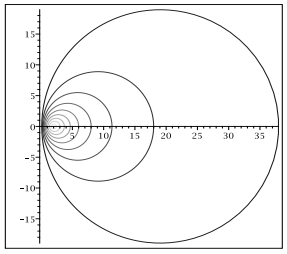
(d) BDF1 (Implicit Euler).



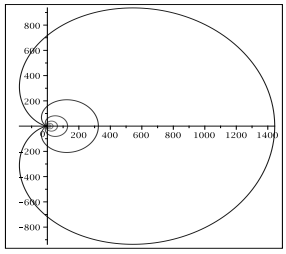
(e) BDF2.



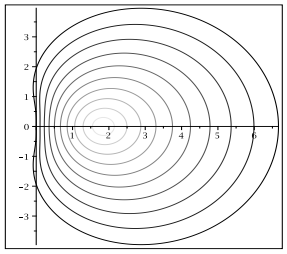
(f) BDF2.



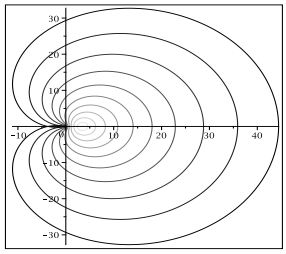
(g) TR.



(h) TR.



(i) BDF3.



(j) BDF3.

FIGURE 2. The curves $\mathfrak{C}(\lambda)$ for different time discretization methods (see Table 1) and parameters ranging from $\lambda = 0$ (white) to $\lambda = 1$ (black); $\Delta t = 1$. On the left the curves for first order time derivative discretizations are depicted (i.e., $\mathfrak{C}(\lambda) = \mathfrak{C}_{\mathcal{P}}$); the related curves for second order in time problems on the right (i.e., $\mathfrak{C}(\lambda) = \mathfrak{C}_{\mathcal{H}}$).

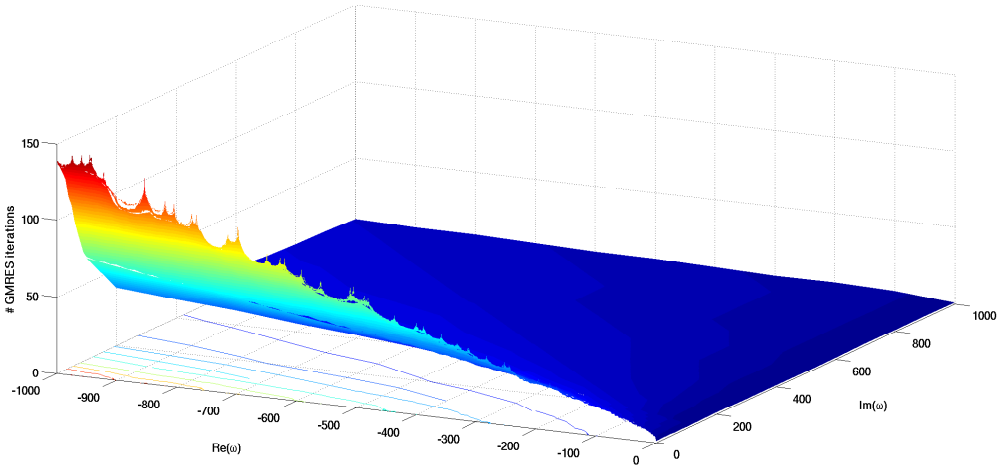


FIGURE 3. Number of iterations for (exact) $|\omega|$i-shifted preconditioned GMRES iteration versus frequency $\omega \in \mathbb{C}$ (prescribed tolerance: 10^{-8}). The singular behavior at the negative real axis reflects the eigenvalues of the discrete Laplacian.

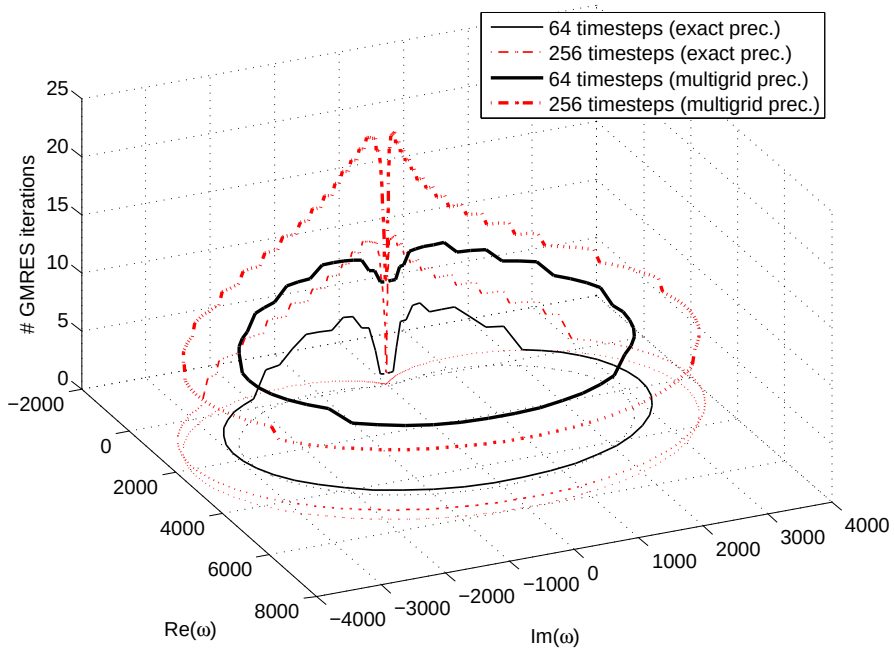


FIGURE 4. Number of iterations for $|\omega|$i-shifted preconditioned GMRES iteration versus frequencies $\omega \in \mathbb{C}$ resulting from the parallelized multistep discretization (see Algorithm 1(b), $\varepsilon = 10^{-6}$) of the wave equation. The preconditioner is computed exactly (thin lines) or approximately inverted by 1 multigrid V-cycle with 2 Jacobi pre-/post-smoothings on each level (prescribed tolerance: 10^{-8}, bold lines).

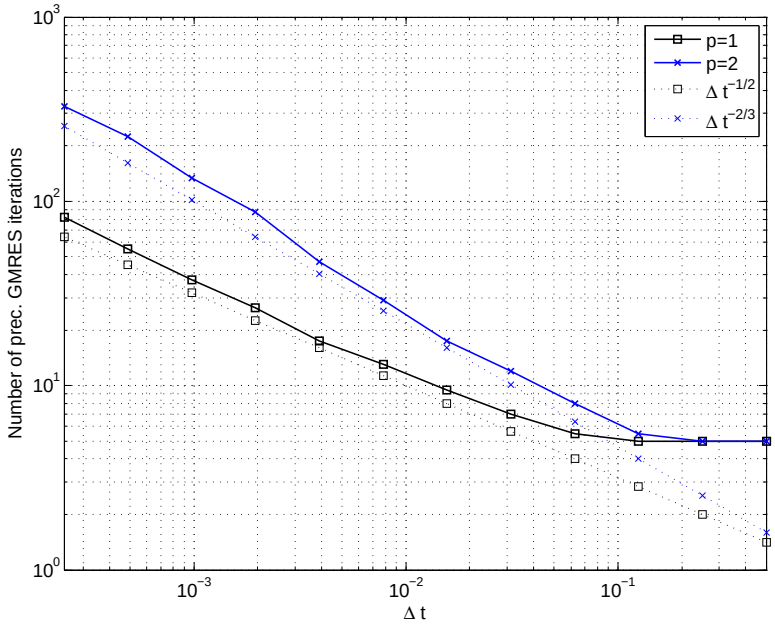
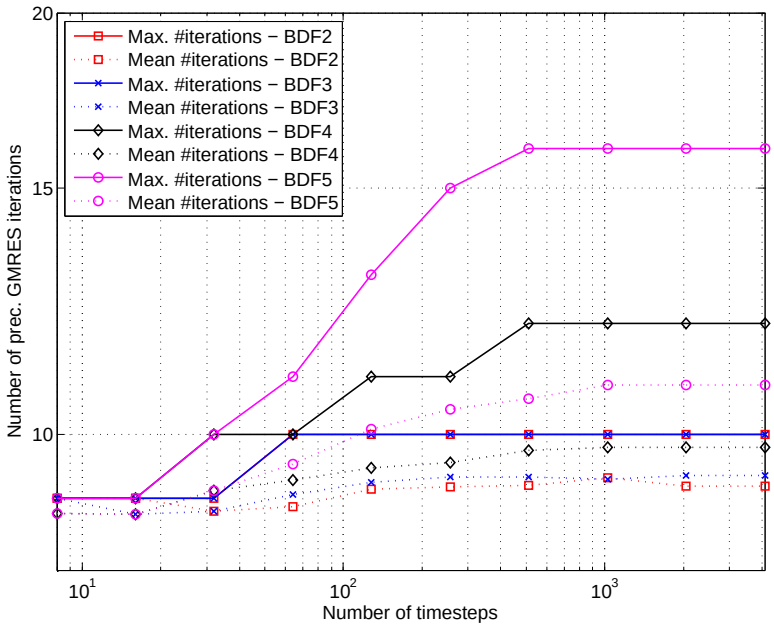
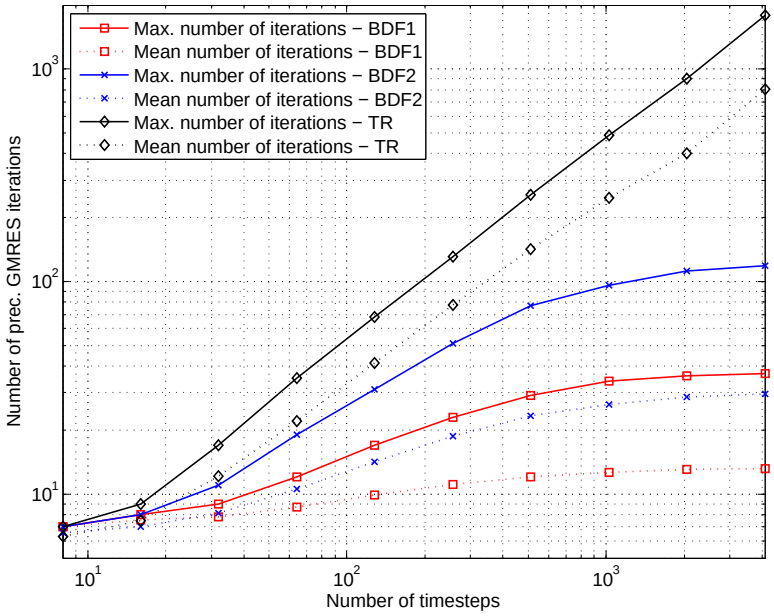


FIGURE 5. We show the increase in the number of iterations of the multigrid preconditioned GMRES when solving (24) with $\omega \in \{z^2 \mid z = \sigma_0 + \mathrm{i}y, |y| \leq \mathrm{const} \cdot \Delta t^{-p/(p+1)}\}$ and decreasing Δt.



(a) Number of preconditioned GMRES iterations in parallelized multistep methods for the heat equation (log-log plot).



(b) Number of preconditioned GMRES iterations in parallelized multistep methods for the wave equation (log-log plot).

FIGURE 6. The maximum/mean number of iterations of multigrid-based (see Figure 4) shifted Laplacian preconditioned GMRES method for the Helmholtz problems to be solved in the parallelized multistep discretization (see Algorithm 1(b)) of the heat and wave equation respectively.

References

[1] L. Banjai. Multistep and multistage convolution quadrature for the wave equation: Algorithms and experiments. *SIAM J. Sci. Comput.*, 32(5):2964-2994, 2010.

[2] L. Banjai and S. Sauter. Rapid solution of the wave equation in unbounded domains. *SIAM J. Numer. Anal.*, 47(1):227–249, 2008.

[3] Dario Bini. Parallel solution of certain Toeplitz linear systems. *SIAM J. Comput.*, 13(2):268–276, 1984.

[4] Timothy A. Davis. *Direct methods for sparse linear systems*, volume 2 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.

[5] Howard C. Elman and Oliver G. Ernst. Numerical experiences with a Krylov-enhanced multigrid solver for exterior Helmholtz problems. In *Mathematical and numerical aspects of wave propagation (Santiago de Compostela, 2000)*, pages 797–801. SIAM, Philadelphia, PA, 2000.

[6] Howard C. Elman, Oliver G. Ernst, and Dianne P. O'Leary. A multigrid method enhanced by Krylov subspace iteration for discrete Helmhotz equations. *SIAM J. Sci. Comput.*, 23(4):1291–1315 (electronic), 2001.

[7] Howard C. Elman and Dianne P. O'Leary. Eigenanalysis of some preconditioned Helmholtz problems. *Numer. Math.*, 83(2):231–257, 1999.

[8] Y. A. Erlangga, C. W. Oosterlee, and C. Vuik. A novel multigrid based preconditioner for heterogeneous Helmholtz problems. *SIAM J. Sci. Comput.*, 27(4):1471–1492 (electronic), 2006.

[9] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee. On a class of preconditioners for solving the Helmholtz equation. *Appl. Numer. Math.*, 50(3-4):409–425, 2004.

[10] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee. Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation. *Appl. Numer. Math.*, 56(5):648–666, 2006.

[11] Yogi A. Erlangga. Advances in iterative methods and preconditioners for the Helmholtz equation. *Arch. Comput. Methods Eng.*, 15(1):37–66, 2008.

[12] Walter Gander and Gene H. Golub. Cyclic reduction—history and applications. In *Scientific computing (Hong Kong, 1997)*, pages 73–85. Springer, Singapore, 1997.

[13] Robert M. Gray. Toeplitz and circulant matrices: A review. *Foundations and Trends in Communications and Information Theory*, 2(3), 2005.

[14] Fu-Rong Lin, Wai-Ki Ching, and Michael K. Ng. Fast inversion of triangular Toeplitz matrices. *Theoret. Comput. Sci.*, 315(2-3):511–523, 2004.

[15] María López-Fernández, Christian Lubich, Cesar Palencia, and Achim Schädle. Fast Runge-Kutta approximation of inhomogeneous parabolic equations. *Numer. Math.*, 102(2):277–291, 2005.

[16] Tristan Needham. *Visual complex analysis*. The Clarendon Press Oxford University Press, New York, 1997.

[17] M. A. Olshanskii and A. Reusken. On the convergence of a multigrid method for linear reaction-diffusion problems. *Computing*, 65(3):193–202, 2000.

[18] Theodore J. Rivlin. *Chebyshev polynomials*. Pure and Applied Mathematics (New York). John Wiley & Sons Inc., New York, second edition, 1990. From approximation theory to algebra and number theory.

[19] C. D. Riyanti, A. Kononov, Y. A. Erlangga, C. Vuik, C. W. Oosterlee, R.-E. Plessix, and W. A. Mulder. A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation. *J. Comput. Phys.*, 224(1):431–448, 2007.

[20] Youcef Saad and Martin H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7(3):856–869, 1986.

[21] Arnold Schönhage. Asymptotically fast algorithms for the numerical multiplication and division of polynomials with complex coefficients. In *Computer algebra (Marseille, 1982)*, volume 144 of *Lecture Notes in Comput. Sci.*, pages 3–15. Springer, Berlin, 1982.

[22] Dongwoo Sheen, Ian H. Sloan, and Vidar Thomée. A parallel method for time discretization of parabolic equations based on Laplace transformation and quadrature. *IMA J. Numer. Anal.*, 23(2):269–299, 2003.

[23] Gilbert Strang. *Computational Science and Engineering*. Wellesley-Cambridge Press, November 2007.

[24] M. B. van Gijzen, Y. A. Erlangga, and C. Vuik. Spectral analysis of the discrete Helmholtz operator preconditioned with a shifted Laplacian. *SIAM J. Sci. Comput.*, 29(5):1942–1958 (electronic), 2007.

(L. Banjai) Max-Planck-Institute for Mathematics in the Sciences, Inselstrasse 22-26, 04103 Leipzig, Germany

E-mail address: banjai@mis.mpg.de

(D. Peterseim) Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany

E-mail address: peterseim@math.hu-berlin.de