Adaptive mesh refinement strategies in isogeometric analysis - a computational comparison

Paul Hennig, Markus Kästner, Philipp Morgenstern, Daniel Peterseim

Angaben zur Veröffentlichung / Publication details:

Hennig, Paul, Markus Kästner, Philipp Morgenstern, and Daniel Peterseim. 2017. "Adaptive mesh refinement strategies in isogeometric analysis - a computational comparison." *Computer Methods in Applied Mechanics and Engineering* 316: 424–48. https://doi.org/10.1016/j.cma.2016.07.029.

Parallel multistep methods for linear evolution problems

LEHEL BANJAI
Max-Planck-Institute for Mathematics in the Sciences, Inselstrasse 22–26, 04103 Leipzig, Germany
banjai@mis.mpg.de

AND

DANIEL PETERSEIM*
Humboldt-Universität zu Berlin, Department of Mathematics, Unter den Linden 6, 10099, Berlin, Germany
*Corresponding author: peterseim@math.hu-berlin.de

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 problem 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 semidiscrete 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; here N is the number of time steps computed in parallel. The complex wave numbers 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) 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 Sheen *et al.* (2003), López-Fernández *et al.* (2005). 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 (Bini, 1984). The original algorithm is restricted to accuracy $\sqrt{\text{eps}}$, where eps is the machine precision; eps $\sim 10^{-16}$ for double precision computations. We describe a correction term that can be computed efficiently 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 generalized minimal residual (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 backward difference formula (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 (TR).

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 a high-order $A(\theta)$-stable discretization of the heat equation and an 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 also need to be solved in methods which rest on the computation of inverse Laplace transforms; see Sheen *et al.* (2003) and López-Fernández *et al.* (2005).

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 (1984) and has a number of

times been re interpreted (and re invented); see Lin *et al.* (2004), Banjai & Sauter (2008). In fact Bini's method is also closely related to the earlier work by Schönhage (1982) 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 $\mathbf{c} = (c_0, c_1, \ldots, c_N)^{\mathrm{T}}$, $T_{\mathrm{U}}(\mathbf{c})$ is the upper triangular Toeplitz matrix with $\mathbf{c}$ as the first row, similarly $T_{\mathrm{L}}(\mathbf{c})$ 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.,

$$T_{\mathrm{U}}(\mathbf{c}) = \begin{pmatrix} c_0 & c_1 & \cdots & c_{N-1} & c_N \\ 0 & c_0 & \cdots & c_{N-2} & c_{N-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & c_0 & c_1 \\ 0 & 0 & \cdots & 0 & c_0 \end{pmatrix}, \quad T_{\mathrm{L}}(\mathbf{c}) = T_{\mathrm{U}}(\mathbf{c})^{\mathrm{T}}$$

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

We refer to Gray (2005) 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

$$P\mathbf{c} = (0, c_N, c_{N-1}, \ldots, c_1)^{\mathrm{T}} \quad \text{for all } \mathbf{c} = (c_0, c_1, \ldots, c_N)^{\mathrm{T}}. \tag{2.1}$$

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

$$T_{\mathrm{L}}(\mathbf{c}) = \Lambda^{-1} C(\mathbf{c}_\lambda)\Lambda - \lambda^{N+1} T_{\mathrm{U}}(P\mathbf{c}), \quad \text{where } \mathbf{c}_\lambda = \Lambda\mathbf{c}.$$

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

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

which shows that

$$T_{\mathrm{L}}(\mathbf{c}) = \Lambda^{-1} T_{\mathrm{L}}(\mathbf{c}_\lambda)\Lambda = \Lambda^{-1} C(\mathbf{c}_\lambda)\Lambda - \Lambda^{-1} T_{\mathrm{U}}(P\mathbf{c}_\lambda)\Lambda.$$

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

The circulant matrix $C(\mathbf{c}_\lambda)$ is diagonalized by the discrete Fourier change of basis (Gray, 2005), 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 $T_{\mathrm{L}}(\mathbf{c})$, the error matrix $\Lambda^{-1} T_{\mathrm{U}}(P\mathbf{c}_\lambda)\Lambda = \lambda^{N+1} T_{\mathrm{U}}(P\mathbf{c})$ hasentries of order $\mathcal{O}(\lambda^{N+1})$, nevertheless in

finite precision arithmetic arbitrary accuracy cannot be obtained as the matrix A is highly ill conditioned for small λ. If eps is the machine precision, then multiplication by A^{-1} increases the rounding errors to λ^{-N} eps, hence in order to make the combined error $\lambda^{-N}\,\mathrm{eps}+\lambda^{N+1}$ as small as possible the optimal choice of the parameter is $\lambda = \mathrm{eps}^{\frac{1}{2N+1}} \approx \sqrt{\mathrm{eps}}^{\frac{1}{N}}$ giving accuracy $\mathrm{eps}^{\frac{N+1}{2N+1}} \approx \sqrt{\mathrm{eps}}$. In double precision arithmetic 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 $\mathbf{c} = (c_0, c_1, \ldots, c_N)^{\mathrm{T}}$, 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

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

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

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

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

$$C(\mathbf{c}_\lambda) \approx \mathcal{F}_{N+1}^{-1}\,\mathrm{diag}\left(c(\lambda), c\bigl(\lambda\zeta_{N+1}^{1}\bigr), \ldots, c\bigl(\lambda\zeta_{N+1}^{N}\bigr)\right)\mathcal{F}_{N+1}. \tag{2.2}$$

If $c(\zeta)$ is a polynomial of degree k, then for $N \geqslant k$, $c(\lambda\zeta_{N+1}^{n}) = (\mathcal{F}_{N+1}\mathbf{c}_\lambda)_n$ and no error is committed in (2.2), otherwise

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

which finally gives an $\mathcal{O}(\lambda^{N+1})$ error in the approximation (2.2). The approximation proposed in Lemma 2.1 together with (2.2) is equivalent to approximating the c_j by the trapezoidal quadrature of the Cauchy integral representation; see Schönhage (1982), Banjai & Sauter (2008).

Let us make one final remark here. If

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

then $T_{\mathrm{L}}(\mathbf{c})^{-1} = T_{\mathrm{L}}(\boldsymbol{d})$. If $c_0 \neq 0$ and $|c_j| \leqslant p^j$, 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 $T_{\mathrm{L}}(\mathbf{c})^{-1}$ by $A^{-1}C(\mathbf{c}_\lambda)^{-1}A$ 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

$$\partial_t^l u + \mathcal{L}u = f, \quad t \in [0, T], \tag{3.1}$$

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 $u(t)$, $f(t) \in X$ for some Banach space X and $t \in [0, T]$.

We discretize (3.1) 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

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

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 an 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 also need higher powers of generating polynomials for which we will use the following notation:

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

Discretizing (3.1) in time by the linear multistep discretization transforms it into the discrete convolutional system which at time $t_N = N\Delta t$ has the form

$$\frac{1}{(\Delta t)^l} \sum_{j=0}^{N} \alpha_{N-j}^{(l)} u_j + \mathcal{L} \sum_{j=0}^{N} \beta_{N-j}^{(l)} u_j = \sum_{j=0}^{N} \beta_{N-j}^{(l)} f_j + \widetilde{f}_N, \tag{3.2}$$

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

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
TR	$1 - \zeta$	$\frac{1}{2} + \frac{1}{2}\zeta$	1	Implicit	2
Explicit Euler	$1 - \zeta$	ζ	1	Explicit	1
Leapfrog formula (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

where the sub indices of functions indicate evaluation at the corresponding time step, i.e., for $u\colon \mathbb{R} \to X$, $u_j := u(j\,\Delta t)$, $j \in \mathbb{Z}$. The initial data $u_{-lk}, \dots, u_{-1}$ and $f_{-lk}, \dots, f_{-1}$ are assumed to be known and to have been moved to the right-hand side; see the definition of $\widetilde{\boldsymbol{f}}$ in (3.3).

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 (3.1), e.g., for $l = 2$,

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

discretize it by the linear multistep formula and then recover the scalar semidiscrete system (3.2).

Writing (3.2) in matrix notation we obtain

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

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

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

$$\widetilde{\boldsymbol{f}} := T_{\mathrm{U}}\big(P\boldsymbol{\beta}^l\big) \otimes \mathcal{I}\boldsymbol{f}_{\mathrm{in}} - T_{\mathrm{U}}\big(P\boldsymbol{\alpha}^l/\Delta t^l\big) \otimes \mathcal{I}\boldsymbol{u}_{\mathrm{in}} - T_{\mathrm{U}}\big(P\boldsymbol{\beta}^l\big) \otimes \mathcal{L}\boldsymbol{u}_{\mathrm{in}}, \tag{3.3}$$

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

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

$$C\big(\boldsymbol{\alpha}_\lambda^l/\Delta t^l\big) \otimes \mathcal{I}\boldsymbol{u}_\lambda + C\big(\boldsymbol{\beta}_\lambda^l\big) \otimes \mathcal{L}\boldsymbol{u}_\lambda = C\big(\boldsymbol{\beta}_\lambda^l\big) \otimes \mathcal{I}\boldsymbol{f}_\lambda + \widetilde{\boldsymbol{f}}_\lambda, \tag{3.4}$$

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

$$\begin{aligned} &T_{\mathrm{L}}\big(\boldsymbol{\alpha}^l/\Delta t^l\big) \otimes \mathcal{I}\boldsymbol{e} + T_{\mathrm{L}}\big(\boldsymbol{\beta}^l\big) \otimes \mathcal{L}\boldsymbol{e} \\ &\qquad = \lambda^{N+1}\big(-T_{\mathrm{U}}\big(P\boldsymbol{\beta}^l\big) \otimes \mathcal{I}\boldsymbol{f} + T_{\mathrm{U}}\big(P\boldsymbol{\alpha}^l/\Delta t^l\big) \otimes \mathcal{I}\widetilde{\boldsymbol{u}} + T_{\mathrm{U}}\big(P\boldsymbol{\beta}^l\big) \otimes \mathcal{L}\widetilde{\boldsymbol{u}}\big). \end{aligned} \tag{3.5}$$

Comparing the right-hand side in the above equation with (3.3) 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 (3.5) 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 $\sqrt{\mathrm{eps}}$. For example, choosing $\lambda^{N+1} = \mathrm{eps}^{1/3}$ after one correction gives an accuracy of $\mathrm{eps}^{2/3}$; for two corrections the optimal choice would be $\lambda^{N+1} = \mathrm{eps}^{1/4}$. The correction strategy is widely known as iterative refinement.

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

$$\frac{1}{(\Delta t)^l}\,\Sigma_1 \otimes \mathcal{I}\widehat{\boldsymbol{u}} + \Sigma_2 \otimes \mathcal{L}\widehat{\boldsymbol{u}} = \Sigma_2 \otimes \mathcal{I}\widehat{\boldsymbol{f}} + \widehat{\widetilde{\boldsymbol{f}}}, \tag{3.6}$$

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 $(\Sigma_2)_{mm} = (\mathcal{F}_{N+1}\boldsymbol{\beta}_\lambda^l)_m$, $m = 0, 1, \ldots, N$. Note that if $N \geqslant lk$, then

$$(\Sigma_1)_{mm} = \alpha^l\big(\lambda\zeta_{N+1}^m\big) \text{ and } (\Sigma_2)_{mm} = \beta^l\big(\lambda\zeta_{N+1}^m\big), \quad m = 0, 1, \ldots, N, \tag{3.7}$$

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

Multiplying (3.6) by $\Sigma_2^{-1} \otimes \mathcal{I}$ we see that we indeed obtain $N + 1$ decoupled Helmholtz-like problems of the type

$$\Big(\frac{s_m}{\Delta t}\Big)^l \widehat{u}_m + \mathcal{L}\widehat{u}_m = \widehat{f}_m, \quad s_m = \delta(\lambda\zeta_{N+1}^m) = \frac{\alpha\big(\lambda\zeta_{N+1}^m\big)}{\beta\big(\lambda\zeta_{N+1}^m\big)}, \tag{3.8}$$

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

4. Semidiscretized 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 $u(\cdot, t) \in H^1(\Omega)$ such that

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

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

Wave equation. Find $u(\cdot, t) \in H^1(\Omega)$ such that

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

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 introduced 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:

$$\begin{aligned} \mathcal{L}\hat{u}_n + \omega_n \hat{u}_n &= \hat{f}_n, \\ \hat{u}_n|_\Gamma &= \hat{g}_n, \end{aligned} \quad n = 0, 1, \ldots, N. \tag{4.1}$$

These Helmholtz problems are uniquely solvable if the linear multistep method is $A(\theta)$-stable for $(\mathcal{P})$ and if it is A-stable for $(\mathcal{H})$; for more details and for explicit methods see Section 4.1.

An approximate solution of (4.1) 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 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 ordinary differential equation.

4.1 *Finite element discretization in space*

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

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

$$a(u,v) := \int_\Omega \nabla u \cdot \overline{\nabla v}, \quad b(u,v) := \int_\Omega u\overline{v}, \quad c(u,v) := \int_\Gamma u\overline{v}. \tag{4.2}$$

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

$$\mathcal{A}u = a(u,\cdot), \quad \mathcal{B}u = b(u,\cdot). \tag{4.3}$$

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

4.2 *The fully discrete equations*

Let us assume that $g_j^* \in S$ are given such that

$$c(g_j^*, v) = c(g_j, v) \quad \text{for all } v \in S, \quad j = 0, 1, \dots, N, \tag{4.4}$$

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

$$T_{\mathrm{L}}\big(\boldsymbol{\alpha}^l/\Delta t^l\big) \otimes \mathcal{B}\boldsymbol{u}^* + T_{\mathrm{L}}\big(\boldsymbol{\beta}^l\big) \otimes \mathcal{A}\boldsymbol{u}^* = T_{\mathrm{L}}\big(\boldsymbol{\beta}^l\big) \otimes \mathcal{B}\boldsymbol{f} - T_{\mathrm{L}}\big(\boldsymbol{\alpha}^l/\Delta t^l\big) \otimes \mathcal{B}\boldsymbol{q}^* - T_{\mathrm{L}}\big(\boldsymbol{\beta}^l\big) \otimes \mathcal{A}\boldsymbol{q}^* + \widetilde{\boldsymbol{f}}, \tag{4.5}$$

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

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

$$C\big(\boldsymbol{\alpha}_\lambda^l/\Delta t^l\big) \otimes \mathcal{B}\boldsymbol{u}_\lambda^* + C\big(\boldsymbol{\beta}_\lambda^l\big) \otimes \mathcal{A}\boldsymbol{u}_\lambda^* = C\big(\boldsymbol{\beta}_\lambda^l\big) \otimes \mathcal{B}\boldsymbol{f}_\lambda - C\big(\boldsymbol{\alpha}_\lambda^l/\Delta t^l\big) \otimes \mathcal{B}\boldsymbol{q}_\lambda^* - C\big(\boldsymbol{\beta}_\lambda^l\big) \otimes \mathcal{A}\boldsymbol{q}_\lambda^* + \widetilde{\boldsymbol{f}}_\lambda. \tag{4.6}$$

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 Section 4.3.

4.3 *Analysis of the discrete Helmholtz problems*

The system (4.6) is equivalent to $N+1$ problems of the following type: find $\hat{u}^* \in X$ such that

$$\mathcal{A}\hat{u}^* + \omega\mathcal{B}\hat{u}^* = \mathcal{B}\hat{f} - \mathcal{A}\hat{g}^* - \omega\mathcal{B}\hat{g}^*. \tag{4.7}$$

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

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

$$\text{there exists } u_j \in X \setminus \{0\} \quad \text{s.t. } \mathcal{A}u_j + \mu_j \mathcal{B}u_j = 0. \tag{4.8}$$

If $\omega \notin \{\mu_j\}$, then equation (4.7) has a unique solution. It is well known that $\mu_j < 0$ for all j and that $|\mu_j| \leqslant Ch^{-2}$, where h is the diameter of the smallest element in the space discretization. The frequencies ω lie on the curve

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

for the parabolic model problem and are on

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

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

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

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

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 LF $\delta(\zeta) = \frac{1-\zeta^2}{2\zeta}$, the curve cuts the negative real line at $\frac{(1+\lambda^2)^2}{4\lambda^2 \Delta t^2}$; the usual CFL condition $\Delta t = \mathcal{O}(h)$ ensures solvability.

For A-stable linear multistep methods it holds that $\mathrm{Re}\,\delta(\zeta) > 0$ for $|\zeta| < 1$, therefore, $\mathfrak{C}_{\mathcal{H}}$ misses the negative real line and (4.7) has a unique solution for all Δt. For example, for the Backward Euler (BDF1), $\delta(\zeta) = 1 - \zeta$, BDF of order 2 (BDF2), $\delta(\zeta) = \frac{3}{2} - 2\zeta + \frac{1}{2}\zeta^2$ and the second-order TR, $\delta(\zeta) = 2(1-\zeta)/(1+\zeta)$, we have that

$$\mathrm{Re}\,\delta(\lambda\zeta) \geqslant (1-\lambda)/\Delta t > 0. \tag{4.9}$$

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

$$\mu_j \notin \mathfrak{C}_{\mathcal{P}}, \text{ for } (\mathcal{P}), \text{ and } \mu_j \notin \mathfrak{C}_{\mathcal{H}}, \text{ for } (\mathcal{H}), \quad j = 0, 1, \ldots, \tag{4.10}$$

where μ_j are the eigenvalues of the discrete Laplacian as defined in (4.8), then the solution $\boldsymbol{u}^*_\lambda$ of (4.6) exists and is unique.

In particular the condition (4.10) is satisfied if the underlying linear multistep method is $A(\theta)$-stable, for $(\mathcal{P})$ and 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 (4.5) and (4.6) respectively, and let $\boldsymbol{u} = \boldsymbol{u}^* + \boldsymbol{q}^*$, $\boldsymbol{u}_\lambda = \boldsymbol{u}_\lambda^* + \boldsymbol{q}_\lambda^*$ and $\widetilde{\boldsymbol{u}} = \mathit{\Lambda}^{-1} \otimes \mathcal{I}\boldsymbol{u}_\lambda$. The error $\boldsymbol{e} = \boldsymbol{u} - \widetilde{\boldsymbol{u}}$ is then the solution of the discrete evolution problem

$$T_{\mathrm{L}}\big(\boldsymbol{\alpha}^l/\mathit{\Delta}t^l\big) \otimes \mathcal{B}\boldsymbol{e} + T_{\mathrm{L}}\big(\boldsymbol{\beta}^l\big) \otimes \mathcal{A}\boldsymbol{e} = \lambda^{N+1}\Big(T_{\mathrm{U}}\big(P\boldsymbol{\alpha}^l/\mathit{\Delta}t^l\big) \otimes \mathcal{B}\widetilde{\boldsymbol{u}} + T_{\mathrm{U}}\big(\boldsymbol{\beta}^l\big) \otimes \mathcal{A}\widetilde{\boldsymbol{u}} - T_{\mathrm{U}}\big(\boldsymbol{\beta}^l\big) \otimes \mathcal{B}\boldsymbol{f}\Big).$$

Proof. Using the results from the previous section, see (3.5), we immediately obtain that $\boldsymbol{e}^* = \boldsymbol{u}^* - \mathit{\Lambda}^{-1} \otimes \mathcal{I}\boldsymbol{u}_\lambda^*$ satisfies

$$\begin{aligned} T_{\mathrm{L}}\big(\boldsymbol{\alpha}^l/\mathit{\Delta}t^l\big) \otimes \mathcal{B}\boldsymbol{e}^* + T_{\mathrm{L}}\big(\boldsymbol{\beta}^l\big) \otimes \mathcal{A}\boldsymbol{e}^* = \\ \lambda^{N+1}\Big(T_{\mathrm{U}}\big(P\boldsymbol{\alpha}^l/\mathit{\Delta}t^l\big) \otimes \mathcal{B}(\boldsymbol{q}^* + \widetilde{\boldsymbol{u}}^*) + T_{\mathrm{U}}\big(P\boldsymbol{\beta}^l\big) \otimes \mathcal{A}(\boldsymbol{q}^* + \widetilde{\boldsymbol{u}}^*) - T_{\mathrm{U}}\big(P\boldsymbol{\beta}^l\big) \otimes \mathcal{B}\boldsymbol{f}\Big), \end{aligned}$$

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

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 $\mathit{\Lambda}^{-1}$. 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 $\partial_t^2 u + k\partial_t u - \mathit{\Delta}u = f$ the only change to the code would be to substitute the wave number $\delta(\zeta)/\mathit{\Delta}t$ in (3.8) by the dissipative equivalent $\sqrt{(\delta(\zeta)/\mathit{\Delta}t)^2 + k\delta(\zeta)/\mathit{\Delta}t}$.

Further, other time discretization methods based on equally spaced time steps produce lower triangular Toeplitz semidiscrete 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 Banjai (2010).

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 that correspond to forward elimination of the lower triangular Toeplitz system. The implementation of both methods is almost trivial. Nevertheless, in Algorithm 1 we give pseudocode for both methods in order to hghlight the differences. In the algorithm 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 $(\mathbf{u}^{\mathrm{T}})_j$ the jth row of $\mathbf{u}$. To simplify the presentation we consider the fully

discrete system (4.5) with homogeneous initial and boundary data written as a discrete convolutional system: find $\boldsymbol{u} \in X^{N+1}$, such that

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

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

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

Require: Data: $\boldsymbol{f} \in \mathbb{R}^{M\times(N+1)}$. 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 $\left(\beta_0\mathcal{A} + \frac{\alpha_0^{(l)}}{\Delta t^l}\mathcal{B}\right)u_n = \boldsymbol{r}$;
end for

(b) Parallelized

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

return u.

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 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 the 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 the same time.

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 $\mathcal{O}\big(\frac{MN\log N}{P}\big)$ time complexity if $P \leqslant \min\{M, N\}$ 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 need 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 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 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

$$w_{\boldsymbol{a},b}\colon \mathbb{R}\times\mathbb{R}^d \to \mathbb{R}, \quad \boldsymbol{x} \mapsto \cos(b(t-\langle \boldsymbol{a},\boldsymbol{x}\rangle))\exp\Big(-10(t-\langle \boldsymbol{a},\boldsymbol{x}\rangle - 3)^2\Big), \tag{6.1}$$

where $\boldsymbol{a} \in \mathbb{R}^d$, $\|\boldsymbol{a}\| = 1$, $b \in \mathbb{R}$. It is easy to see that $w_{\boldsymbol{a},b}$ fulfils 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

$$\begin{aligned} \partial_t^2 u(x,t) - \Delta u(x,t) &= 0, \quad (x,t) \in \Omega\times[0,T],\\ u(x,0) &= 0, \quad x\in\Omega,\\ \partial_t u(x,0) &= 0, \quad x\in\Omega,\\ u(x,t) &= w_{[1,0],1}(x) + w_{[-1,1],2}(x), \quad x\in\partial\Omega. \end{aligned} \tag{6.2}$$

The results of the related computations are listed in Table 2. They are based on a conforming piecewise affine finite element space discretization with respect to uniform refinements of the initial triangulation depicted in Fig. 1. The level parameter ref $= 2, 3, \ldots, 8$ denotes the number of uniform refinements

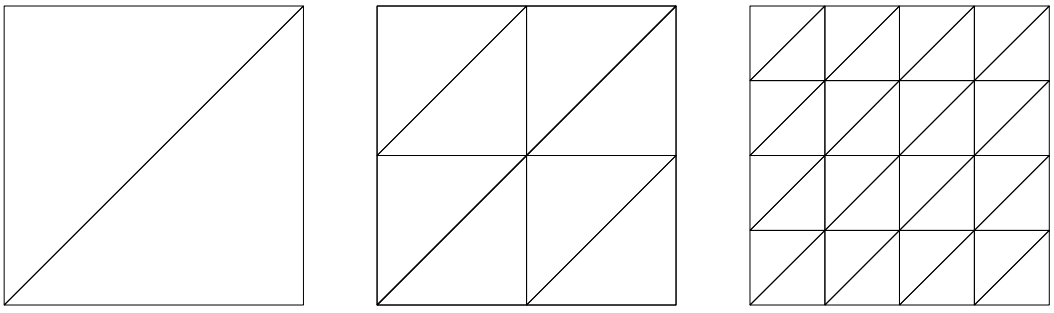
TABLE 2 *Convergence history for model problem* (6.2)*:* $e^{\mathrm{ts}}_{\mathrm{ref}}$ *and* $e^{\mathrm{par}}_{\mathrm{ref}}$ *denote the errors of Algorithm* 1*(a) and (b), respectively, on refinement level* ref*; the difference column gives relative distances between the two approximations measured in the* $\|\cdot\|_{L^\infty([0,T],L^2(\Omega))}$ *norm*

(a) TR

ref	$\|e^{\mathrm{ts}}_{\mathrm{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	$\|e^{\mathrm{par}}_{\mathrm{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	Difference	Rate
2	0.52449640	0.52449633	4.1×10^{-07}	1.05
3	1.08740731	1.08740752	1.1×10^{-06}	−1.05
4	0.91896229	0.91896275	9.1×10^{-07}	0.23
5	0.52220239	0.52220257	5.0×10^{-07}	0.82
6	0.15753554	0.15753554	1.8×10^{-07}	1.70
7	0.04117602	0.04117600	7.6×10^{-08}	1.96
8	0.01061508	0.01061506	1.1×10^{-07}	2.03

(b) BDF2

ref	$\|e^{\mathrm{ts}}_{\mathrm{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	$\|e^{\mathrm{par}}_{\mathrm{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	Difference	Rate
2	0.41693631	0.41693631	1.9×10^{-08}	—
3	0.54310906	0.54310906	8.5×10^{-08}	−0.40
4	0.64303205	0.64303206	1.5×10^{-07}	−0.25
5	0.62529911	0.62529916	2.4×10^{-07}	0.02
6	0.36945373	0.36945381	2.2×10^{-07}	0.77
7	0.14528735	0.14528733	1.3×10^{-07}	1.40
8	0.04089906	0.04089903	7.2×10^{-08}	1.86



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

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

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 TR 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.

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 behaviour of simulations based on the BDF3 time discretization. Let $T = 10$, $\Omega = [0, 1]^2$. Consider

$$\begin{aligned} \partial_t^2 u(x,t) - \Delta u(x,t) &= 0, \quad (x,t) \in \Omega \times [0,T], \\ u(x,0) &= \sin(2\pi x), \quad x \in \Omega, \\ \partial_t u(x,0) &= 0, \quad x \in \Omega, \\ u(x,t) &= 0, \quad x \in \partial\Omega. \end{aligned} \tag{6.3}$$

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 $\mathcal{T}_{\mathrm{ref}}$ 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 and the error is even lower than that of the BDF2 method with the same time step; see Table 3(a) and Table 3(b). This is theoretically justified by the fact that the related curve $\mathfrak{C}_{\mathcal{H}} := \{(\alpha_{\mathrm{BDF3}}(10^{-6/N}\zeta))^2/\Delta t^2 \,|\, |\zeta| = 1\}$ for $N = 256$, does not intersect the negative real axis. Doubling the number of time steps makes the above curve cross the negative

TABLE 3 *Convergence history for model problem* (6.3)*:* $e^{\mathrm{ts}}_{\mathrm{ref}}$ *and* $e^{\mathrm{par}}_{\mathrm{ref}}$ *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*

(a) BDF2				
ref	$\|e^{\mathrm{ts}}_{\mathrm{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	$\|e^{\mathrm{par}}_{\mathrm{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	Difference	Rate
2	0.48881127	0.48881127	9.8×10^{-10}	—
3	0.48783902	0.48783902	4.4×10^{-09}	0
4	0.91685161	0.91685161	2.0×10^{-07}	−0.94
5	0.75215079	0.75215078	2.8×10^{-07}	0.29
6	0.69648890	0.69648886	4.8×10^{-07}	−0.10
7	0.59814940	0.59814966	1.1×10^{-06}	0.22
8	0.20057960	0.20057953	2.6×10^{-06}	1.56
(b) BDF3				
ref	$\|e^{\mathrm{ts}}_{\mathrm{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	$\|e^{\mathrm{par}}_{\mathrm{ref}}\|_{L^\infty([0,T],L^2(\Omega))}$	Difference	Rate
2	0.51198670	0.51198670	3.0×10^{-09}	—
3	0.49146349	0.49146349	7.43×10^{-09}	0.06
4	1.14187459	1.14187447	2.2×10^{-07}	−1.21
5	3.12117657	3.12118521	1.1×10^{-05}	−1.45
6	2.71643601	2.71644405	1.2×10^{-05}	0.20
7	0.26507886	0.26507961	5.1×10^{-06}	3.38
8	0.02912403	0.02911598	2.3×10^{-04}	3.16

real axis and (keeping space resolution) gives $\|e^{\mathrm{ts}}\|_{L^\infty([0,T],L^2(\Omega))} = 16.51$ and $\|e^{\mathrm{par}}\|_{L^\infty([0,T],L^2(\Omega))} = 0.389$. Decreasing λ would again make the curve avoid the negative real axis, see Fig. 2, but would make the matrix A^{-1} more singular, thereby creating numerical difficulties.

6.2 *A few remarks on the efficient solution of linear Helmholtz-type systems*

The application of the parallel Algorithm 1(b) requires the solution of discrete Helmholtz problems of the form

$$H(\omega) := Au + \omega Bu = f, \quad A, B \in \mathbb{R}^{M\times M}, \quad f \in \mathbb{C}^M, \quad \omega \in \mathfrak{C}(\lambda) \subset \mathbb{C} \tag{6.4}$$

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 Fig. 2 such curves are depicted for several implicit and explicit linear multistep methods.

In contrast to the linear systems arising from time-stepping algorithms, problem (6.4) 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 (6.4). It is known that the linear problems (6.4) can be solved by means of multigrid or multigrid preconditioned iterative solvers in optimal time complexity $\mathcal{O}(M)$ provided $\operatorname{Re}\omega \geqslant 0$ and $\operatorname{Im}\omega = 0$; see Olshanskii & Reusken (2000). We could not find corresponding multigrid convergence results for the case $\operatorname{Re}\omega \geqslant 0$ and $\operatorname{Im}\omega \neq 0$, but in experiments the same kind of behaviour is seen.

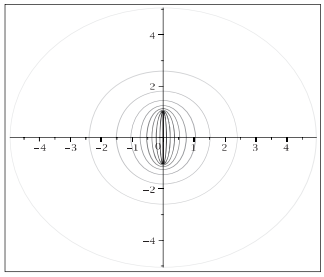
We will now briefly discuss the availability of efficient solvers for indefinite Helmholtz problems that appear whenever the curves related to the linear multistep method intersect the left-half complex plane. This will happen using $A(\theta)$-stable methods for parabolic problems and A-stable methods 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 $N \to \infty$.

The matrices A and B from (6.4) are matrix representations of the operators $\mathcal{A}$ and $\mathcal{B}$ defined in (4.3). While A is positive semidefinite, B is assumed to be positive definite. In the setting of the previous section, A and B are also symmetric and sparse.

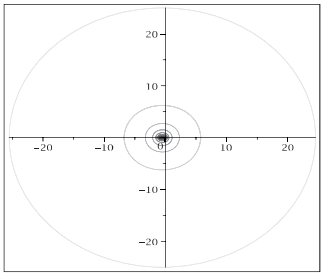
Some linear system solvers are not affected by the indefiniteness of $H(\omega)$, e.g., sparse direct solvers will be equally fast in all cases since the sparsity pattern of the matrix is independent of the wave number; we refer to the textbook Davis (2006) for an overview on fast direct solvers for sparse linear systems. Furthermore, so- called Fast Poisson Solvers (see Strang, 2007, 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 (6.4) 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 Gander & Golub, 1997) 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., Elman & O'Leary, 1999; Elman & Ernst, 2000; Elman *et al.*, 2001). The key problem is that the spectral properties of the operator change 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.

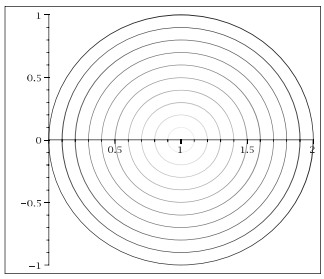
The GMRES method introduced in Saad & Schultz (1986) is one of the most effective methods for solving large sparse systems of equations if equipped with a suitable preconditioner. Starting from an



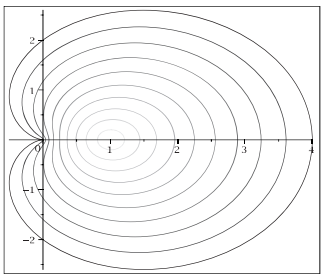
(a) Leapfrog.



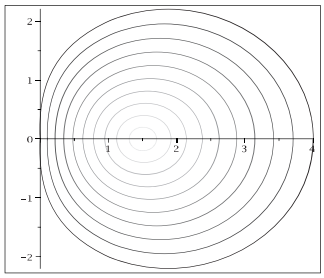
(b) Leapfrog.



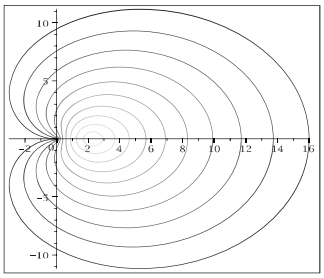
(c) BDF1 (Implicit Euler).



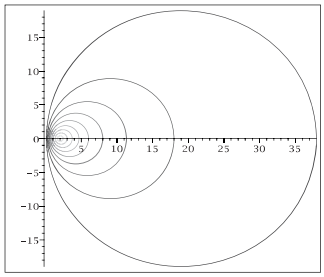
(d) BDF1 (Implicit Euler).



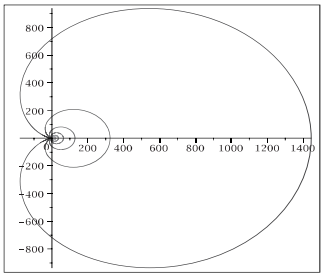
(e) BDF2.



(f) BDF2.

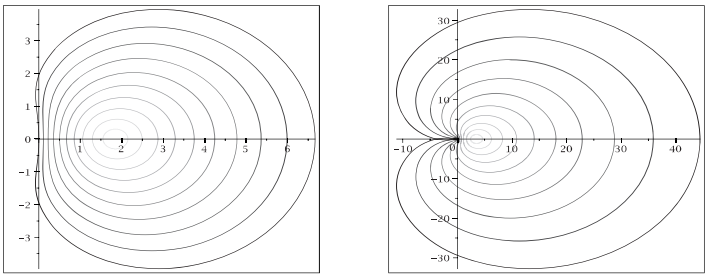


(g) TR.



(h) TR.

FIG. 2. (continued)



(i) BDF3. (j) BDF3.

FIG. 2. The curves $\mathfrak{C}(\lambda)$ for different time discretization methods (see Table 1) and parameters ranging over a greyscale 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 are on the right (i.e., $\mathfrak{C}(\lambda) = \mathfrak{C}_{\mathcal{H}}$).

initial guess $x^0 \in \mathbb{C}^M$ and the associated residual $r^0 := f - Hx^0$, the algorithm computes a sequence of iterates $x^1, x^2, \ldots$ such that the mth residual $r^m := f - Hx^m$ satisfies

$$\left\|r^m\right\|_2 = \min_{p\in\mathbb{P}_m:\ p(0)=1} \left\|p(H)r^0\right\|_2, \quad m = 0, 1, \ldots, M-1, \tag{6.5}$$

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 Saad & Schultz (1986, Proposition 4) to be bounded as follows:

$$\left\|r^m\right\|_2 \leqslant \kappa(X) \underbrace{\left(\min_{p\in\mathbb{P}_m:\ p(0)=1} \max_{j=1,\ldots,M} \left|p(d_j)\right|\right)}_{=:\varepsilon_m} \left\|r^0\right\|_2. \tag{6.6}$$

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 $R > 0$ centred at $q \in \mathbb{C}\setminus\{0\}$, Theorem 5 in Saad & Schultz (1986) gives a bound on ε_m

$$\varepsilon_m \leqslant \left(\min_{p\in\mathbb{P}_m: p(0)=1} \max_{z\in\mathfrak{B}_R(q)} |p(z)|\right) = \left(\frac{R}{|q|}\right)^m. \tag{6.7}$$

The equality in (6.7) is known as Zorantonellos's lemma (Rivlin, 1990). If $\frac{R}{|q|} \leqslant c < 1$, then the GMRES iteration will produce a decreasing sequence of residuals that fulfil $\|r^m\| < \text{tol}$, for a given tolerance $\text{tol} > 0$, as soon as $m > \frac{\log(\text{tol}/\kappa(X))}{\log(c)}$.

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 nonzero $u \in \mathbb{R}^M$ such that

$$Au = \mu Bu. \tag{6.8}$$

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

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

and problem (6.4) is uniquely solvable if and only if $\omega \neq -\mu$ for all solutions μ of (6.8). The spectrum of matrices $H(\omega)$ from (6.4) is in general not clustered in a ball that does not contain zero and the GMRES convergence bound (6.7) is not applicable.

A suitable preconditioner for the problem at hand is the shifted Laplacian preconditioner that was introduced and analysed in Erlangga *et al.* (2006a,b), Erlangga *et al.* (2004), van Gijzen *et al.* (2007). 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

$$\frac{\mu+\omega}{\mu+z}, \quad \mu \in \mathbb{R}_{\geqslant 0}\colon Au = \mu Bu.$$

If we consider the related Möbius transform

$$M\colon \mathbb{C} \to \mathbb{C}, \quad M(\xi) := \frac{\xi+\omega}{\xi+z}, \tag{6.9}$$

then for the spectrum of the preconditioned operator it holds that

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

We refer to Needham (1997, 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 which are regarded as circles with infinite radius. As a consequence we can state that the eigenvalues of the preconditioned operator lie on the circle centred 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

$$P(z)^{-1}H(\omega)u = (A+zB)^{-1}(A+\omega B)u = P(z)^{-1}f \tag{6.10}$$

can be bounded as follows:

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

This result recovers the convergence bound of the GMRES iteration for the preconditioned system (6.10) given in van Gijzen *et al.* (2007, 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 $H(\omega)$ ($\operatorname{Re}\omega \geqslant 0$, $\operatorname{Re}\omega < -\max(\sigma_B(A))$), we can choose $z = \omega$ since standard multigrid techniques will be efficient (Elman & Ernst, 2000). The critical setting is that $H(\omega)$ is indefinite, implying that $-\max(\sigma_B(A)) \leqslant \operatorname{Re}\omega < 0$. Assuming that $\operatorname{Im}\omega \neq 0$, i.e., $0 \notin M(\mathbb{R})$, we minimize the convergence bound

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

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 is

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

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

$$\frac{\partial g_\omega(z)}{\partial \operatorname{Re} z} \geqslant 0 \quad \text{for all } z \in \mathbb{C}_+.$$

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

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

results in the following choice of z_{opt}:

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

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

$$P(z_{\mathrm{opt}}(\omega))^{-1} H(\omega) u = P(z_{\mathrm{opt}}(\omega))^{-1} f \tag{6.11}$$

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

$$\sqrt{\frac{1 - \sin(\theta_\omega)}{1 + \sin(\theta_\omega)}}, \quad \text{where } \theta_\omega := \arcsin\left(\frac{|\operatorname{Im} \omega|}{|\omega|}\right) \in \left(0, \tfrac{\pi}{2}\right]. \tag{6.12}$$

Therefore, the systems (6.4) arising from the decoupling procedure can be solved efficiently as long as θ_ω remains moderately bounded from below for $\omega \in \mathfrak{C}(\lambda)$. In Figure 3 we plot the dependence of the number of GMRES steps on ω.

Considering $A(\theta)$-stable linear multistep methods for parabolic problems (see Fig. 2(c, e, g)) the minimal value of θ_ω is bounded below by the θ parameter implying that the convergence rate can be

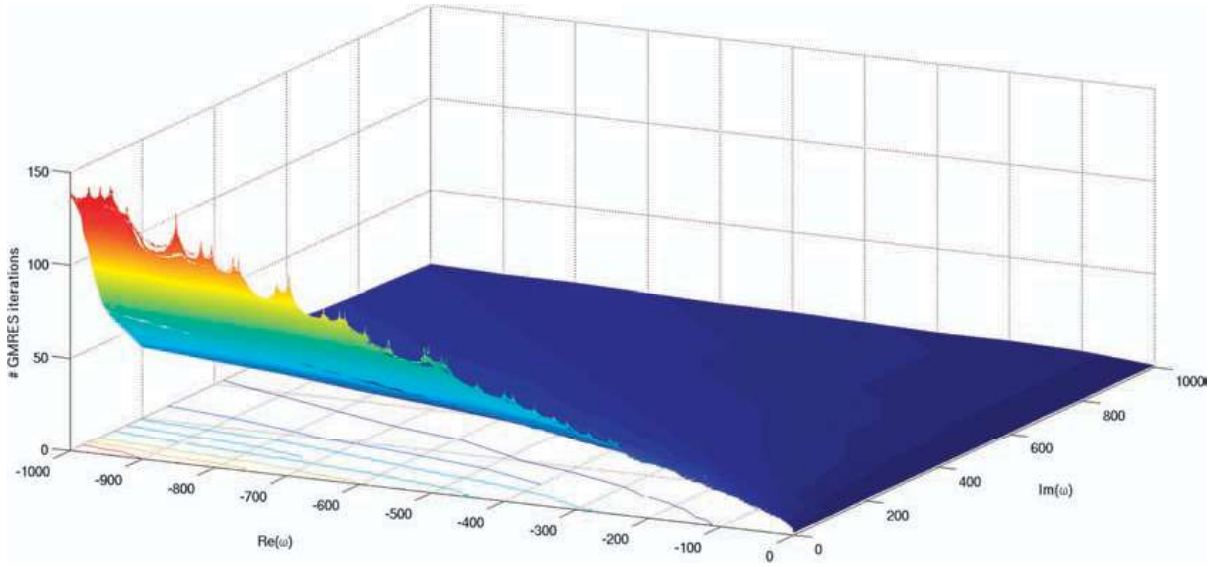


FIG. 3. Number of iterations for (exact) $|\omega|$i-shifted preconditioned GMRES iteration versus frequency $\omega \in \mathbb{C}$ (prescribed tolerance: 10^{-8}). The singular behaviour at the negative real axis reflects the eigenvalues of the discrete Laplacian.

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 Fig. 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 (6.12) 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 (4.9), we have that

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

Writing $x + \mathrm{i}y = \frac{\delta(\lambda\zeta)}{\Delta t}$, x is smallest, i.e, closest to σ_0, for $y = 0$. For 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 methods, e.g., Radau IIA methods, except that 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 (6.4) are on the curve

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

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 Fig. 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 $\omega \in \{z^2 | z = \delta(\zeta)/\Delta t, |\zeta| = 1\}$. 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 TR 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 experiment described below.

We have also investigated the behaviour of the multigrid solver by comparing the behaviour 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 Fig. 4 tell us that typically one multigrid V-cycle is enough as has also been observed in van Gijzen *et al.* (2007), Erlangga *et al.* (2004), Erlangga *et al.* (2006a,b), Riyanti *et al.* (2007), Erlangga (2008). 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 efficiency of the outer Krylov solver rather than the nonoptimal performance of the smoother.

We now present the numerical experiments for *Regime 2* described above. For different numbers of time steps we iteratively (GMRES with inexact multigrid-based shifted Laplacian preconditioner; parameters as in Fig. 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 $\varepsilon = 1 \times 10^{-06}$. We use a conforming piecewise affine FEM with respect to the mesh hierarchy $\{\mathcal{T}_{\text{ref}}\}_{\text{ref}=2}^{8}$ (see Fig. 1) as before; the right-hand side data are 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 has been proved before there is only a dependence on

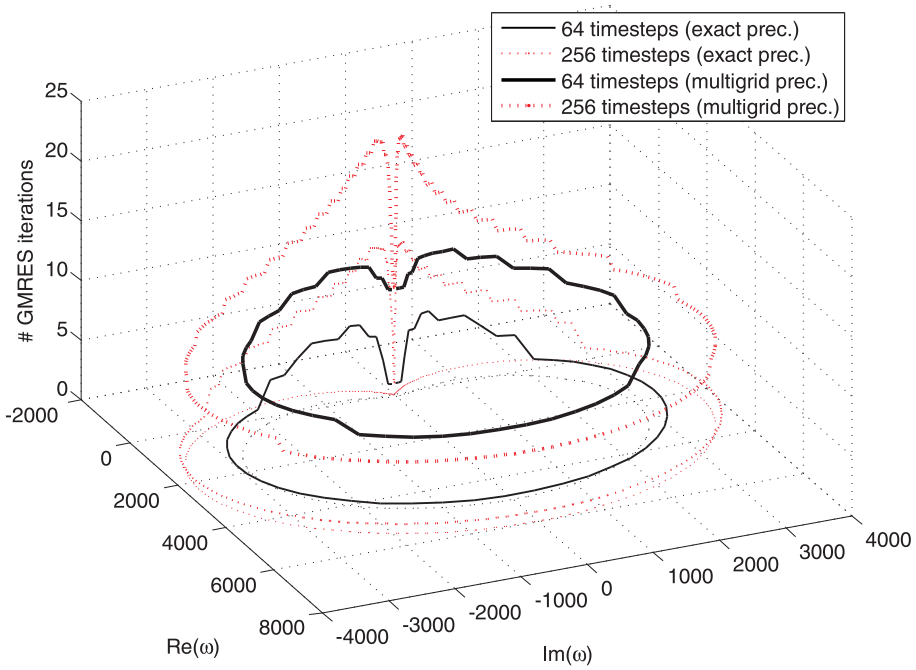


FIG. 4. Number of iterations for $|\omega|$i-shifted preconditioned GMRES iteration versus frequency $\omega \in \mathbb{C}$ resulting from the parallelized multistep discretization (see Algorithm 1(b), $\varepsilon = 10^{-6}$) of the wave equation. The preconditioner is computed exactly (thin lines)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

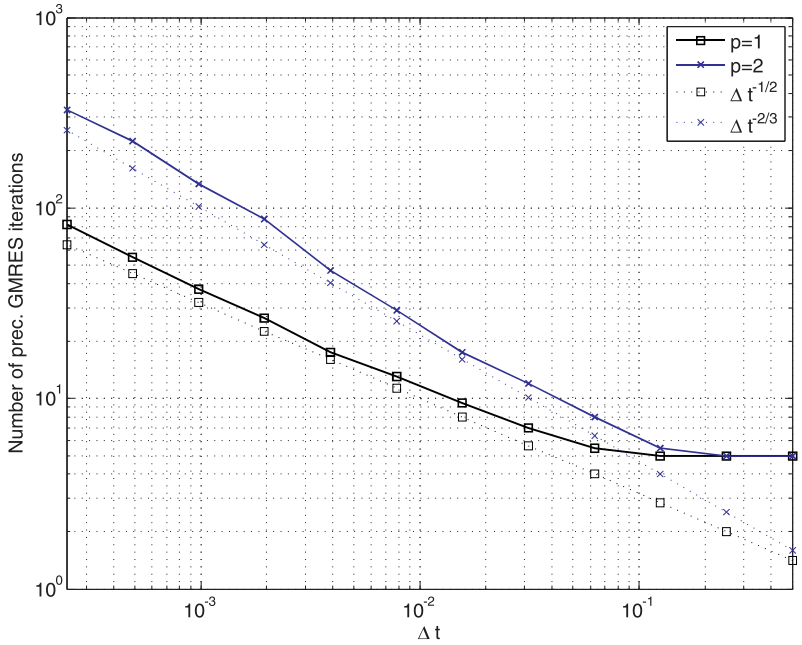
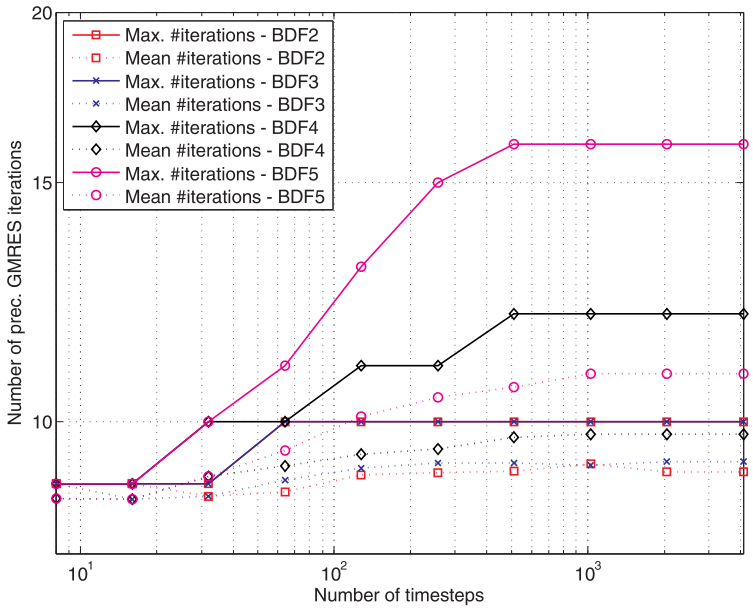
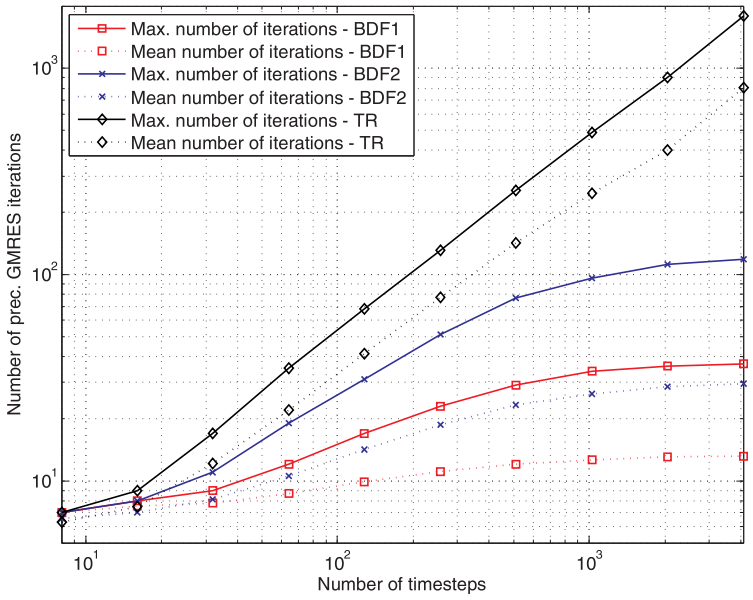


FIG. 5. We show the increase in the number of iterations of the multigrid preconditioned GMRES when solving (6.4) with $\omega \in \{z^2 | z = \sigma_0 + \mathrm{i}y, |y| \leqslant \text{const.} \cdot \Delta t^{-p/(p+1)}\}$ and decreasing Δt.



(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).

FIG. 6. The maximum/mean number of iterations of the multigrid-based (see Fig. 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. (a) Number of preconditioned GMRES iterations in parallelized multistep methods for the heat equation (log-log plot) and (b) number of preconditioned GMRES iterations in parallelized multistep methods for the wave equation (log-log plot).

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 TR; see Fig. 6(b). The increase seems to be linear with the number of time steps 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.

Acknowledgements

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

Funding

The second author is supported by the Research Center MATHEON of the German Research Foundation DFG (Deutsche Forschungsgemeinschaft).

References

BANJAI, L. (2010) Multistep and multistage convolution quadrature for the wave equation: algorithms and experiments. *SIAM J. Sci. Comput.*, **32**, 2964–2994.

BANJAI, L. & SAUTER, S. (2008) Rapid solution of the wave equation in unbounded domains. *SIAM J. Numer. Anal.*, **47**, 227–249.

BINI, D. (1984) Parallel solution of certain Toeplitz linear systems. *SIAM J. Comput.*, **13**, 268–276.

DAVIS, T. A. (2006) *Direct Methods for Sparse Linear Systems*. *Fundamentals of Algorithms*, vol. 2. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).

ELMAN, H. C. & ERNST, O. G. (2000) Numerical experiences with a Krylov-enhanced multigrid solver for exterior Helmholtz problems. *Mathematical and Numerical Aspects of Wave Propagation* (Santiago de Compostela, 2000). Philadelphia, PA: SIAM, pp. 797–801.

ELMAN, H. C., ERNST, O. G. & O'LEARY, D. P. (2001) A multigrid method enhanced by Krylov subspace iteration for discrete Helmhotz equations. *SIAM J. Sci. Comput.*, **23**, 1291–1315 (electronic).

ELMAN, H. C. & O'LEARY, D. P. (1999) Eigenanalysis of some preconditioned Helmholtz problems. *Numer. Math.*, **83**, 231–257.

ERLANGGA, Y. A. (2008) Advances in iterative methods and preconditioners for the Helmholtz equation. *Arch. Comput. Methods Eng.*, **15**, 37–66.

ERLANGGA, Y. A., OOSTERLEE, C. W. & VUIK, C. (2006a) A novel multigrid based preconditioner for heterogeneous Helmholtz problems. *SIAM J. Sci. Comput.*, **27**, 1471–1492 (electronic).

ERLANGGA, Y. A., VUIK, C. & OOSTERLEE, C. W. (2004) On a class of preconditioners for solving the Helmholtz equation. *Appl. Numer. Math.*, **50**, 409–425.

ERLANGGA, Y. A., VUIK, C. & OOSTERLEE, C. W. (2006b) Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation. *Appl. Numer. Math.*, **56**, 648–666.

GANDER, W. & GOLUB, G. H. (1997) Cyclic reduction—history and applications. *Proceedings of the Workshop on Scientific Computing : 10–12 March, 1997, Hong Kong* (F. T. Luk & R. J. Plemmons eds). New York: Springer.

GRAY, R. M. (2005) Toeplitz and circulant matrices: a review. *Found. Trends Commun. Inf. Theory*, **2**, 155–239.

LIN, F.-R., CHING, W.-K. & NG, M. K. (2004) Fast inversion of triangular Toeplitz matrices. *Theor. Comput. Sci.*, **315**, 511–523.

LÓPEZ-FERNÁNDEZ, M., LUBICH, C., PALENCIA, C. & SCHÄDLE, A. (2005) Fast Runge–Kutta approximation of inhomogeneous parabolic equations. *Numer. Math.*, **102**, 277–291.

NEEDHAM, T. (1997) *Visual Complex Analysis*. New York: The Clarendon Press.

OLSHANSKII, M. A. & REUSKEN, A. (2000) On the convergence of a multigrid method for linear reaction–diffusion problems. *Computing*, **65**, 193–202.

RIVLIN, T. J. (1990) *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*. Pure and Applied Mathematics (New York), 2nd edn. New York: John Wiley.

RIYANTI, C. D., KONONOV, A., ERLANGGA, Y. A., VUIK, C., OOSTERLEE, C. W., PLESSIX R.-E. & MULDER, W. A. (2007) A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation. *J. Comput. Phys.*, **224**, 431–448.

SAAD, Y. & SCHULTZ, M. H. (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, **7**, 856–869.

SCHÖNHAGE, A. (1982) Asymptotically fast algorithms for the numerical multiplication and division of polynomials with complex coefficients. *Computer Algebra* (Marseille, 1982), vol. 144. Lecture Notes in Computer Science. Berlin: Springer, pp. 3–15.

SHEEN, D., SLOAN, I. H. & THOMÉE, V. (2003) A parallel method for time discretization of parabolic equations based on Laplace transformation and quadrature. *IMA J. Numer. Anal.*, **23**, 269–299.

STRANG, G. (2007) *Computational Science and Engineering*. Wellesley, MA: Wellesley-Cambridge Press.

VAN GIJZEN, M. B., ERLANGGA, Y. A. & VUIK, C. (2007) Spectral analysis of the discrete Helmholtz operator preconditioned with a shifted Laplacian. *SIAM J. Sci. Comput.*, **29**, 1942–1958 (electronic).