All-at-once solution of linear wave equations

Federico Danieli | Andrew J. Wathen

Mathematical Institute, University of Oxford, Oxford, UK

Correspondence
Federico Danieli, Mathematical Institute, Radcliffe Observatory, Andrew Wiles Building, Woodstock Rd, Oxford OX2 6GG, UK.
Email: danieli@maths.ox.ac.uk

Funding information
Culham Centre for Fusion Energy;
Engineering and Physical Sciences Research Council, Grant/Award Number: EP/L015803/1

Abstract
Efficient parallel-in-time methods for hyperbolic partial differential equation problems remain scarce. Here we investigate an approach based on circulant preconditioned generalised minimal residual (GMRES) for the monolithic block Toeplitz equations which arise from constant time-step discretizations. We present theoretical results which guarantee convergence in a number of iterations independent of the number of time-steps and demonstrate the potential utility of the approach with numerical results employing several different finite difference schemes of varying orders of accuracy.

KEYWORDS
circulant preconditioning, GMRES, hyperbolic equations, parallel-in-time integration

1 | INTRODUCTION

Several techniques now exist for parallel-in-time computation, that is computational approaches which reduce wall-clock time for evolutionary problems by employing computations acting in parallel along the temporal domain: we point the interested reader to the excellent review on the topic written by Gander,\(^1\) as well as to the more recent one by Ong and Schroder.\(^2\) In particular, for problems arising from the discretization of parabolic partial differential equations (PDEs), a number of schemes have been suggested, the principle ones being based on the Parareal\(^3\) or PFASST\(^4\) algorithms, and their acceleration using multigrid ideas.\(^5-7\) Examples of successful applications of these methods to hyperbolic problems are however still scarce, with some noticeable exceptions,\(^8-10\) although research in this direction is underway.\(^11-13\)

Other suggested approaches have employed a so-called monolithic or all-at-once approximation, where equations that define solutions at many or all time levels are considered; the contribution from this article falls into that category. Monolithic systems arising from the approximation of constant-coefficient, time-evolving differential equations present a clear structure when discretized using fixed time steps: their constituting blocks, in fact, are the same down each diagonal. When dealing with periodic problems, this gives rise to a block circulant matrix, whose structure can be directly employed for time-parallelization,\(^14\) since it can be diagonalized via fast Fourier transform (FFT). When solving an initial value problem, instead, the circulant structure is broken, but we retain a block Toeplitz monolithic system: in this case, we can still exploit a circulant approximation in order to accelerate the iterative solution of such system,\(^15,16\) which is the approach we pursue in this article. The literature provides theoretical results which indicate the potential of this approach for parallel-in-time computation,\(^16\) and preliminary results from simple parallel implementations bear out the optimal parallel efficiency of this approach.\(^17\) Intriguingly, this efficiency still seems to hold even for a particular discretization of the wave equation.\(^17\) Given that regular time-stepping might be considered more appropriate for wave problems, we further investigate this and related approaches in this article. We show how a result of Chan, Potts, and Steidl on circulant preconditioning\(^18\) applies to this situation and establishes in several cases that only a small number of...
generalised minimal residual (GMRES) iterations is required for convergence/termination of the method. This number is independent of the number of time steps. An alternative approach due to Gander et al.\textsuperscript{19,20} is applicable when irregular time steps are chosen; by contrast to the methods described in this article, such an approach might be particularly relevant for parabolic problems.

In Section 2, we describe the all-at-once preconditioned Krylov iteration method, as well as the main theorem employed in our work,\textsuperscript{18} together with its consequences. In Section 3 we give numerical results for hyperbolic problems. We conclude this article in Section 4.

2 CIRCULANT PRECONDITIONING FOR TOEPLITZ SYSTEMS

In the literature, a variety of circulant preconditioners for Toeplitz matrices have been proposed,\textsuperscript{21} and indeed, for the simple test cases considered, some of these approaches coincide. In this article we exploit some of the results from Chan, Potts, and Steidl:\textsuperscript{18} we decide then to follow most of the notation used in their article, and describe the \( \omega \)-circulant preconditioner therein defined.

2.1 \( \omega \)-Circulant preconditioner

Let us introduce a \( 2\pi \)-periodic generating function \( f(x) \), with the associated Toeplitz matrix \( A_N(f) \in \mathbb{R}^{N \times N} \), that is:

\[
[A_N(f)]_{m,n} = a_{m-n}(f), \quad \text{with} \quad a_k(f) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x)e^{-ikx} \, dx,
\]

where the \( a_k(f) \) are the Fourier coefficients of \( f(x) \). In our work we are mostly concerned with banded Toeplitz matrices with a small band, which have trigonometric polynomials as generating functions, in the form

\[
f(x) = \sum_{k=-s_2}^{s_1} a_k e^{ikx}, \quad \text{for some} \quad a_k \in \mathbb{R}, \quad s_1, s_2 \in \mathbb{N}, \quad 0 < s_1 + s_2 \ll N.\tag{2}
\]

In this case, the Fourier coefficients (1) are well defined, and coincide with the coefficients of the polynomial (2); moreover, the summation extrema \( s_1 \) and \( s_2 \) in (2) directly denote the number of sub- and superdiagonals of \( A_N(f) \), respectively, so that this matrix has the following structure:

\[
A_N(f) = \begin{bmatrix}
    a_0 & a_{-1} & \ldots & a_{-s_2} \\
    a_{1} & \ddots & \ddots & \ddots \\
    \vdots & \ddots & \ddots & \ddots \\
    a_{s_1} & \ddots & \ddots & a_{-1} \\
    a_{s_1} & \ldots & a_1 & a_0
\end{bmatrix}.
\]

We further request that \( f(x) \) has no zeroes on the \( N \) equispaced nodes:

\[
f(x_{N,l}) \neq 0 \quad \forall x_{N,l} = w + \frac{2\pi l}{N}, \quad \text{with} \quad l = 0, \ldots, N-1,
\]

where \( w \in [0, 2\pi / N) \) is a free parameter which gives some flexibility in ensuring that we steer away from zeroes of the generating function: the \( \omega \) in the name of the original preconditioner\textsuperscript{18} is given by \( \omega = e^{iNw} \). To simplify our notation, in this section we limit ourselves to the choice \( w = 0 \) (i.e., \( \omega = 1 \)), but the treatment would remain largely unchanged otherwise. We also introduce the diagonal matrix \( D_N(f) \in \mathbb{R}^{N \times N} \), which contains evaluations of \( f \) on the nodes \( x_{N,l} \) defined in (4):

\[
D_N(f) = \text{diag} \left( f(x_{N,l}) \right),
\]

(5)
as well as the discrete Fourier transform operator,

\[ [F_N]_{m,n} = \frac{1}{\sqrt{N}} e^{-\frac{2\pi i}{N} mn}, \quad m, n = 0, \ldots, N-1. \] (6)

The \( \omega \)-circulant matrix \( M_N(f) \) associated with \( A_N(f) \) is then defined as:

\[ M_N(f) = F_N \quad D_N(f) \quad F_N^*, \] (7)

In our case, characterizing \( M_N(f) \) is particularly simple:

\[
[M_N(f)]_{m,n} = \begin{cases} 
  a_{m-n} & \text{if } m = n \\
  a_{m-n} + a_{(m-n)-N} & \text{if } m > n \\
  a_{m-n} + a_{(m-n)+N} & \text{if } m < n 
\end{cases} \Rightarrow M_N(f) = \begin{bmatrix} 
  a_0 & (a_{-1} + a_{N-1}) & (a_{-2} + a_{N-2}) & \cdots \\
  (a_1 + a_{-(N-1)}) & \ddots & \ddots & \ddots \\
  (a_2 + a_{-(N-2)}) & \ddots & \ddots & \ddots \\
  \vdots & \ddots & \ddots & \ddots 
\end{bmatrix};
\] (8)

that is, in order to recover \( M_N(f) \), the off-diagonal elements of \( A_N(f) \) are summed to those \( N \) diagonals away. Notice that in our particular setting, where the band of \( A_N(f) \) is small (specifically, smaller than \( N/2 \)), and \( \omega = 1 \), the \( \omega \)-preconditioner coincides with the Strang circulant preconditioner.\(^{15}\)

Formula (7) exposes a particularly favorable property of circulant matrices: they can be trivially diagonalized via Fourier transform. Inverting \( M_N(f) \), then, amounts to applying a fast Fourier transform (FFT), inverting a diagonal matrix, and applying the inverse FFT: a task of overall complexity \( \mathcal{O}(N \log N) \). This argument can be similarly extended to block circulant matrices, in which case we would achieve block-diagonalization. Therein also lies the concurrency of the algorithm: solving a (block) diagonal system is a task which can be trivially parallelized. Being simple to invert, then, (7) definitely satisfies the first requirement of a preconditioner. As a second requirement, we would also hope for it to effectively accelerate the convergence of iterative methods. We address this in the following.

### 2.2 Convergence of circulant-preconditioned Krylov methods

There is a choice of iterative methods one can use, paired with the preconditioner (7) (or slight modifications thereof), to solve a system of equations involving (3). For example,\(^{16}\) one could symmetrize the Toeplitz system via a Hankel matrix, and use MINRES with the absolute value of (7) as preconditioner. An alternative\(^{18}\) consists in solving the normal equation using conjugate gradient, preconditioned with \( M_N(|f|^2) \). In both cases, strong theoretical bounds guarantee fast convergence of the methods used. We are mostly concerned with convergence of circulant preconditioners applied to GMRES. While convergence results are harder to come by in this case,\(^{22}\) some experiments\(^{23,24}\) hint at it being a more effective iterative solver with respect to other choices available.

In this article, we use a result from Chen, Potts, and Steidl\(^{18}\) which deals with the application of (7) as a right-preconditioner for GMRES iterations of a system involving (3). Adapted to our case, the theorem states the following:

**Theorem 1.** Let \( f \) be a trigonometric polynomial such as (2), satisfying (4), and generating the Toeplitz matrix \( A_N(f) \), with associated \( \omega \)-circulant preconditioner \( M_N(f) \). Then,

\[ A_N(f)M_N(f)^{-1} = I_N + R_N(s_1 + s_2), \] (9)

where \( I_N \in \mathbb{R}^{N \times N} \) is the identity matrix, while \( R_N(s) \in \mathbb{R}^{N \times N} \) identifies a matrix of low rank \( s \).

**Proof.** In our situation, the proof simplifies significantly. We can split \( A_N(f) \) as \( M_N(f) - B_N(f) \), where

\[
B_N(f) = \begin{bmatrix}
  a_{s_1} & a_{s_1-1} & \cdots & a_1 \\
  \vdots & \ddots & \ddots & \vdots \\
  a_{-s_2} & \ddots & a_{s_1-1} & \cdots \\
  a_{-s_2+1} & \ddots & \ddots & a_{s_1} \\
  \vdots & \ddots & \ddots & \ddots \\
  a_{-1} & \cdots & a_{-s_2+1} & a_{-s_2} 
\end{bmatrix},
\] (10)
so that

\[ A_N(f)M_N(f)^{-1} = I_N(f) - B_N(f)M_N(f)^{-1}, \] (11)

from which it follows that \( B_N(f)M_N(f)^{-1} = I_N(f) - B_N(f) \), since \( B_N(f) \) only has nonzero elements in the top \( s_1 \) and bottom \( s_2 \) rows.

\[ \text{Remark 1.} \] Notice that the theorem allows for some flexibility in breaking the Toeplitz structure of \( A_N(f) \). If we consider a perturbation \( \tilde{A}_N(f) \) to \( A_N(f) \), we can still decompose \( A_N(f) + \tilde{A}_N(f) = M_N(f) - \tilde{B}_N(f) \), where \( \tilde{B}_N(f) = B_N(f) - \tilde{A}_N(f) \). Provided that the entries in \( \tilde{A}_N(f) \) which do not adhere to the Toeplitz structure are only limited to the first \( s_1 \) and last \( s_2 \) rows, then, Theorem 1 still holds. This is particularly useful if we consider that we are dealing with the solution of discrete space–time systems. In this framework, in fact, we are often forced to modify the equations corresponding to the first few temporal nodes, be it in order to accommodate for initial conditions on the derivative, and/or in order to kick-start high-order discretizations in time: this has the effect of perturbing the Toeplitz structure of the system, but not to an extent that would break the result of Theorem 1, as described more in detail in Section 3.

As a note, the original theorem\(^\text{18} \) is more general, and deals rather with \textit{rational} generating functions:

\[ f(x) = \frac{p(x)}{q(x)} = \frac{\sum_{k=0}^{d_1} p_k e^{ikx}}{\sum_{k=0}^{d_2} q_k e^{ikx}}, \quad \text{with} \quad d_1, d_2 \in \mathbb{N}, \quad d_1d_2 \neq 0. \] (12)

In that case, the original theorem states that the rank of the resulting perturbation of the identity is equal to \( \max\{d_1, d_2\} \). It is possible to recover our version of Theorem 1 by noticing that we can rewrite (2) as \( f(x) = (\sum_{k=0}^{d_1+d_2} a_{-s_2+k} e^{ikx}) / e^{is_2x} \).

Noticeably, Theorem 1 gives us a theoretical guarantee that GMRES with \( M_N(f) \) as a right-preconditioner will converge to the solution after at most \( s_1 + s_2 + 1 \) iterations, at least in exact precision: if the band of \( A_N(f) \) is small enough, then (7) provides an excellent preconditioner. Examples of its effectiveness, as well as a discussion of the cases in which this theorem fails to come to our aid, are described in Section 3.

3 | ALL-AT-ONCE SOLUTION OF LINEAR DIFFERENTIAL EQUATIONS

In this section, we illustrate how simple systems arising from the discretization of differential equations are (close-to-) Toeplitz in nature, which makes them amenable to be preconditioned using (7). We also introduce a few test cases, and the methods used in our experiments to recover their numerical solution. The setup follows the one presented by Goddard and Wathen.\(^\text{17} \)

3.1 | An ordinary differential equation

As a proof of concept, useful in introducing some of the notation employed, we propose the simplest linear second-order differential equation:

\[
\begin{align*}
\frac{d^2u}{dt^2} - \lambda u &= 0 \quad t \in (0, T] \\
u(0) &= \tilde{u}^0 \\
\frac{du}{dt}(0) &= \tilde{u}^1
\end{align*}
\] (13)

where \( \tilde{u}^0 \) and \( \tilde{u}^1 \in \mathbb{R} \) represent the initial values of the solution itself and its derivative, respectively.

3.1.1 | Discretization aspects

To discretize our temporal domain, we choose a uniform grid with step size \( \Delta t = T/(N_t - 1) \). Furthermore, to approximate (13) we consider \textit{Störmer–Verlet} schemes (SV) of various orders: this is a family of multistep methods specifically tailored.
for the solution of second-order differential equations [25, chap. III-10]. The general formulation for a kth order SV scheme is given by

$$u^n - 2u^{n-1} + u^{n-2} = \Delta t^2 \sum_{j=0}^{k-1} a_j \nabla^j f^n,$$  

(14)

where the right-hand side of the system $f^n$ simply corresponds to $\lambda u^n$ in our case; $\nabla$ identifies the backward difference operator

$$\nabla f^n = \nabla f^{n-1} - \nabla f^{n-2},$$  

(15)

while the coefficients $a_j$ are picked in such a way to ensure the desired level of accuracy [25, table.10.2]: the values for the schemes used in this article are

$$[a_j]_{j=0}^8 = [1, -1, \frac{1}{12}, 0, -\frac{1}{240}, -\frac{1}{240}, -\frac{221}{60480}, -\frac{19}{60480}, -\frac{9829}{3628800}].$$  

(16)

Notice that for all cases but $k = 2$, (14) defines an implicit method. For ease of notation, we identify discretization schemes of different accuracies by reporting their order as a subscript to their acronym. We can see that choosing SV$_1$ corresponds to approximating the second-order derivative using a first-order backward difference formula, which produces the following recurrence relation:

$$\left\{ \begin{array}{l} \frac{u^n - 2u^{n-1} + u^{n-2}}{\Delta t^2} - \lambda u^n = 0 \quad n = 2, \ldots, N_t - 1 \\ 2u^2 - 2u^1 - \lambda u^1 = \frac{2}{\Delta t} \bar{u}^1 \\ u^0 = \bar{u}^0 \end{array} \right..$$  

(17)

Notice that we use a ghost-node approach to include the initial condition on the derivative at $t = 0$; that is, we introduce a fictitious node $u^{-1}$, approximate the first derivative using the central difference (CD) formula $u^1 - u^{-1} = 2\Delta t \bar{u}^1$, and substitute in the first equation of (17) with $n = 1$ to recover the second equation. We can gather the Equation (17) together and build the system

$$\left( \Phi^{SV_1} - \lambda \Delta t^2 \Phi^r_{SV_1} \right) u = g^{SV_1},$$  

(18)

where $u = [u^0, u^1, \ldots, u^{N_t-1}]^T \in \mathbb{R}^{N_t}$ collects the values of the solution at each temporal node. The operator $\Phi^{SV_1} \in \mathbb{R}^{N_t \times N_t}$ represents the discretization of the time derivative (including the perturbation from the ghost-node approach), and has the following structure:

$$\Phi^{SV_1} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ -2 & -2 & -2 & \cdots & -2 \\ 2 & 1 & -2 & \cdots & 1 \\ & 2 & 1 & \cdots & 1 \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -2 & 1 \end{bmatrix} \in \mathbb{R}^{N_t \times N_t}.$$  

(19)

Operator $\Phi^r_{SV_1}$ stems from the evaluation of the right-hand side in (14), and it reduces to $\Phi^r_{SV_1} = I_{N_t} \in \mathbb{R}^{N_t \times N_t}$, the identity matrix, for SV$_1$. The effect of the initial conditions is included in the right-hand side:

$$g^{SV_1} = [1 - \lambda \Delta t^2] \bar{u}^0, 2\Delta t \bar{u}^1, 0, \ldots, 0]^T \in \mathbb{R}^{N_t}.$$  

(20)

Notice how imposing the initial conditions breaks the Toeplitz structure of the matrix (19), and hence that of the system $A^{SV_1}$: the second element in the second row is different from the others on the diagonal. However, we can still
assemble the circulant preconditioner disregarding this perturbation, as such:

\[ M^{SV_1} = \Phi^{SV_1} - \lambda \Delta t^2 \Phi_r^{SV_1}, \]  

(21)

where \( \Phi^{SV_1} \) is our circulant approximation to \( \Phi^{SV_1} \), built as follows:

\[
\Phi^{SV_1} = \begin{bmatrix}
1 & 1 & -2 \\
-2 & 1 & 1 \\
1 & -2 & 1 \\
& & & \ddots & \ddots \\
& & & 1 & -2 & 1
\end{bmatrix} \in \mathbb{R}^{N_t \times N_t}.
\]  

(22)

As hinted in Remark 1, the statement of Theorem 1 still remains valid using \( M^{SV_1} \) as a preconditioner for \( M^{SV_1} \), even if the system is not strictly Toeplitz: in fact, we only perturbed the second row on a matrix with a band of size 2.

We also experiment using higher order discretizations. For orders > 2, this requires recovering the values of the solution at the first few time steps in some alternative way, in order to kick-start the multistep method [26, chap. 5.9.3]. This is due both to the stencil generally becoming larger as the order increases, and also to the fact that the ghost-node approach used in (17) is only second-order accurate, and will end up polluting the global accuracy of higher order schemes. There are a number of ways in which the kick-starting procedure might be conducted (generally resorting to self-starting schemes to solve for the first few unknowns), which end up disrupting the Toeplitz structure of the space–time system. For simplicity, in our experiments we directly evaluate the analytical solution for the necessary nodes; however, in order to mimic the impact the perturbation stemming from a kick-starting procedure might have on the performance of the circulant preconditioner, we also modify the first few equations in the system by leaving only the contribution from the main diagonal. For example, using \( SV_5 \) gives rise to the space–time system:

\[
\begin{bmatrix}
\Phi^{SV_5} - \lambda \Delta t^2 \Phi_r^{SV_5}
\end{bmatrix} u = g^{SV_5},
\]  

(23)

where \( \Phi^{SV_5} \) and \( \Phi_r^{SV_5} \) play the same roles as \( \Phi^{SV_1} \) and \( \Phi_r^{SV_1} \) in (19), but both have their first four rows perturbed, as such:

\[
\Phi^{SV_5} = \begin{bmatrix}
1 & 1 & -2 \\
0 & 1 & 1 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 1 & -2 & 1 \\
& & & & \ddots & \ddots \\
& & & & 1 & -2 & 1
\end{bmatrix} \in \mathbb{R}^{N_t \times N_t}, \quad \text{and} \quad \Phi_r^{SV_5} = \frac{1}{240} \begin{bmatrix}
19 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
& & & & -1 & 4 & 14 & 204 & 19 \\
& & & & \ddots & \ddots & \ddots & \ddots & \ddots
\end{bmatrix} \in \mathbb{R}^{N_t \times N_t}.
\]  

(24)

The circulant preconditioner is then assembled as in (21), starting from the circulant approximations to the operators in (24), that is,

\[ M^{SV_5} = \Phi^{SV_5} - \lambda \Delta t^2 \Phi_r^{SV_5}, \]  

(25)

where \( \Phi^{SV_5} \) is the same as \( \Phi^{SV_1} \), while \( \Phi_r^{SV_5} \) is built disregarding the perturbations from the underlying Toeplitz structure of \( \Phi_r^{SV_1} \). Finally, the right-hand side of the space–time system is built as

\[
[g^{SV_5}]_j = \begin{cases}
[A^{SV_5}]_{jj} \mu^j & \text{if } j \leq 3 \\
0 & \text{if } j > 3
\end{cases},
\]  

(26)
TABLE 1  Number of iterations to convergence for GMRES, right-preconditioned with (7), applied to the solution of systems arising from the discretization of (13)

| $N_t$ | SV1 | SV2 | SV3 (=SV4) | SV5 | SV6 | SV7 | SV8 |
|-------|-----|-----|-----------|-----|-----|-----|-----|
| $10^2$ | 3   | 3   | 3         | 5   | 6   | 7   | 8   |
| $10^3$ | (3) | (3) | (3)       | (5) | (6) | (7) | (8) |
| $10^4$ | (3) | (3) | (3)       | (5) | (6) | (7) | (8) |
| $10^6$ | (3) | (3) | (3)       | (5) | (6) | (7) | (8) |
| $10^7$ | (3) | (3) | (3)       | (5) | (6) | (7) | (8) |
| $10^8$ | (3) | (3) | (3)       | (5) | (6) | (7) | (8) |
| $10^9$ | (3) | (3) | (3)       | (5) | (6) | (7) | (8) |
| $10^{10}$ | (3) | (3) | (3)       | (5) | (6) | (7) | (8) |

Note: Systems of different size $N_t$, and SV schemes of various orders of accuracy considered. The top row reports the theoretical upper bound from Theorem 1. The two columns of values refer to different tolerances for convergence: $\varepsilon = 10^{-7}$ (left, in brackets), $\varepsilon = 10^{-10}$ (right). Randomly chosen initial guess for GMRES (normal distribution). Problem parameters: $\bar{u}_0 = 1$, $\bar{u}_1 = -1$, $\lambda = -1$, $T = 10^3$.

assuming the values at the first few instants $u^j = u(j\Delta t)$ (with $j = 0, \ldots, 3$), are given. Notice that Remark 1 still applies, and that the result of Theorem 1 remains valid even in this case, since we perturbed the first four rows on a matrix where $s_1 = 4$ and $s_2 = 0$. Similar considerations hold for the systems corresponding to higher order discretizations, which are built following a similar procedure.

3.1.2 | Results

In Table 1 we collect the convergence results from the application of (7) as a right-preconditioner for GMRES, used to recover the solution of the systems (18), (23), and others built using SV discretizations of various orders. The theoretical bound provided in Theorem 1 is respected reasonably well: the discrepancies are attributable to round-off errors, and seem more pronounced for higher order schemes and more refined grids. The $\omega$-preconditioner is further tuned by picking the parameter $w$ in (4) as $w = \pi/N_t$, which (at least for the lowest order schemes) ensures that the roots of the corresponding generating function are the farthest away from 0; this in turns attains the desirable property of minimizing the condition number of the preconditioner. However, we deem relevant to point out that, while reviewing this article, a much more detailed analysis of the impact of this parameter on the effectiveness of the preconditioner has been presented.27

3.2 | A hyperbolic partial differential equation

Ultimately, our interest resides in applying this method to the solution of hyperbolic partial differential equations. As a test case of this class, we consider the wave equation, with Dirichlet boundary conditions:

\[
\begin{align*}
\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} &= 0 & (x, t) &\in (0, L) \times (0, T) \\
\frac{\partial u(x, 0)}{\partial t} &= \tilde{u}^0(x) & x &\in [0, L] \\
\frac{\partial u(x, 0)}{\partial x} &= \tilde{u}^1(x) \\
u(0, t) &= u(L, t) = 0 & t &\in (0, T)
\end{align*}
\]  

(27)

3.2.1 | Discretization aspects

Most of the notation used in Section 3.1 can be extended to the PDE case, but we need to introduce an appropriate discretization of the spatial operator. To this purpose, we choose once again a uniform grid spanning the interior of the
spatial domain, with spacing $\Delta x = L/(N_x + 1)$, and we consider a CD scheme for the second-order spatial derivative. Even in this case we make use of approximations of different orders of accuracy, to match the temporal discretizations. Furthermore, the Dirichlet boundary conditions in (27) are imposed naturally. This causes some inconvenience when using high-order schemes: in that case, in fact, we cannot use symmetric stencils anymore as we approach the boundaries, since we would require information from nodes outside the domain. We circumvent this issue by using, for those few nodes, finite difference of the same order of accuracy, but defined on a slanted stencil. \(^{28}\) For example, for second- and fourth-order-accurate discretizations, we obtain the following matrices:

\[
K^{\text{CD}_2} = \begin{bmatrix}
-2 & 1 & \cdot & \cdot & \cdot \\
1 & -2 & 1 & \cdot & \cdot \\
\cdot & \cdot & -2 & 1 & \cdot \\
\cdot & \cdot & \cdot & -2 & 1 \\
1 & \cdot & \cdot & \cdot & 1
\end{bmatrix} \in \mathbb{R}^{N_x \times N_x}, \quad \text{and} \quad K^{\text{CD}_4} = \frac{1}{12} \begin{bmatrix}
-6 & 14 & -15 & 10 \\
16 & -30 & 16 & 1 \\
-1 & 16 & -30 & 16 & -1 \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
-1 & 16 & -30 & 16 & -1 \\
10 & -15 & -4 & 14 & -6
\end{bmatrix} \in \mathbb{R}^{N_x \times N_x}.
\tag{28}
\]

We point out that the scheme stemming from choosing $\text{SV}_2$ for the temporal discretization and $\text{CD}_2$ for the spatial one is also known as the explicit leapfrog method \(^{26}\), chap. 10.2.2\), which is conditionally stable. This can be shown via Von Neumann analysis \(^{26}\), chap. 9.6\): assuming the solution is composed of a single Fourier mode $u^n = G^n e^{ikx}$, for a certain growth factor $G$ and frequency $k$, and plugging this into the scheme, we obtain the recurrence relation

\[
\frac{G^n - 2G^{n-1} + G^{n-2}}{\Delta t^2} e^{ikx} = \frac{e^{i\Delta x} + 2 + e^{-i\Delta x}}{\Delta x^2} G^{n-1} e^{ikx}. \tag{29}
\]

We are interested in the roots of the associated characteristic polynomial, given by the solutions of

\[
g^2 - 2(1 + r^2 c_2)g + 1 = 0, \tag{30}
\]

where we named $c_2 = \cos(k\Delta x) - 1 \in (-2, 0)$, and $r = \Delta t/\Delta x$, in order to reduce notation. Its roots, denoted as $g_{\pm}$, satisfy the relationships

\[
g_+ g_- = 1, \tag{31a}
\]

and

\[
g_+ + g_- = 2(1 + r^2 c_2). \tag{31b}
\]

We want to ensure that $|g_{\pm}| \leq 1 \forall k$, and that roots falling on the unit circle are simple. This can only be guaranteed if these are given as a complex conjugate pair of modulo 1 (otherwise they would break (31a)). Consequently, we can translate (31a) into requesting $|\Re(g_{\pm})| = |1 + r^2 c_2| \leq 1$, which only holds if $r^2 \leq 1$, given the range of $c_2$. For the fourth-order discretization, a similar reasoning can be applied. The roots of the characteristic polynomial associated with using $\text{SV}_4$ in time and $\text{CD}_4$ in space, are given by

\[
\left(1 - \frac{1}{12} r^2 c_4 \right) g^2 - 2 \left(1 + \frac{5}{12} r^2 c_4 \right) g + \left(1 - \frac{1}{12} r^2 c_4 \right) = 0
\]

\[
\Rightarrow \quad g^2 - 2 \left(\frac{12 + 5r^2 c_4}{12 - r^2 c_4} \right) g + 1 = 0, \tag{32}
\]

this time with $c_4 = (16 \cos(k\Delta x) - \cos(2k\Delta x) - 15)/12 \in (-8/3, 0)$. The conditions on its roots resemble those in (31a); for stability, we then need to impose

\[
|\Re(g_{\pm})| = \left|\frac{12 + 5r^2 c_4}{12 - r^2 c_4} \right| \leq 1 \quad \Rightarrow \quad r^2 \leq \frac{9}{4}. \tag{33}
\]
For higher order methods the analysis gets more complex, as increasing the size of the stencil increases the order of the associated characteristic polynomial, but they also require constraints on the size of \( r \) in order to ensure stability.

To avoid these restrictions, we also introduce another discretization, namely an implicit version of the leapfrog method,\(^{29}\) which we consider only to second-order accuracy, and which we denote by CI\(_2\). The only difference with explicit leapfrog scheme lies in the fact that the spatial operator is averaged over three consecutive time steps, giving rise to the following recurrence relation:

\[
\frac{u^n - u^{n-1} + u^{n-2}}{\Delta t^2} = \frac{1}{4\Delta x^2}K_{\text{CD}_2} u^n + \frac{2}{4}u^{n-1} + \frac{1}{4}u^{n-2}.
\]  

(34)

Even in this case, Von Neumann analysis comes to our aid in establishing that the method is unconditionally stable. After some algebraic manipulations, we can show that the characteristic polynomial of interest this time is given by

\[
g^2 - 2\left(\frac{2 + r^2 c_2}{2 - r^2 c_2}\right) g + 1 = 0,
\]

(35)

with \( c_2 \in (-2, 0) \) defined as before. Similar considerations as for the case above show that its roots are a complex conjugate pair falling on the unit circle whenever \( |2 + r^2 c_2| < |2 - r^2 c_2| \); given \( c_2 \) is always negative, this holds \( \forall r^2 > 0 \).

We are finally ready to assemble the monolithic space–time system, by opportune combinations of the matrices introduced above. For CI\(_2\), or SV\(_k\) temporal discretizations of a given order \( k \), we do so in the following blockwise fashion:

\[
A^{\text{SV}_k} = \Phi^{\text{SV}_k} \otimes I^{N_t} - r^2 \left( \Phi_k^{\text{SV}_k} \otimes K^{\text{CD}_2} \right),
\]

and

\[
A^{\text{CI}_2} = \Phi^{\text{SV}_2} \otimes I^{N_t} - \frac{r^2}{4} \left( I^{N_t} + 2I^{N_t} + I^{N_t}_{-1} \right) \otimes K^{\text{CD}_2},
\]

(36)

where \( \otimes \) represents the Kronecker product, \( \lceil * \rceil \) is the ceiling function, while \( I^{N_t}_k \) is the lower shift matrix, containing only ones on the \( k \)th subdiagonal. Also, \( I^{N_t}_k \) is a slight modification of the identity matrix, containing a 2 in the second row: this is done to account for the initial conditions on the derivative. Notice how the part for \( A^{\text{CI}_2} \) referring to the spatial discretization (the term containing the Kronecker product with \( K^{\text{CD}_2} \)) is made of three adjacent blocks per block-row, corresponding to an averaging over three consecutive instants. The corresponding right-hand sides are assembled in a similar way as to what was done in Section 3.1:

\[
g^{\text{SV}_k} = \left[(I^{N_t}_k - r^2 K^{\text{CD}_2}) \bar{u}^0, 2\Delta t \bar{u}^1, 0, \ldots, 0\right]^T,
\]

\[
g^{\text{SV}_k} = \left[(A^{\text{SV}_k})_{j=0}^{N_t} u^0, \ldots, (A^{\text{SV}_k})_{j=(k-2)N_t}^{(k-1)N_t} u^{k-2}, 0, \ldots, 0\right]^T \quad \text{for } k \geq 3,
\]

(37)

and

\[
g^{\text{CI}_2} = \left[(I^{N_t}_k - \frac{r^2}{4} K^{\text{CD}_2}) \bar{u}^0, 2 \left(\Delta t I^{N_t}_k - \frac{r^2}{4} K^{\text{CD}_2}\right) \bar{u}^1, 0, \ldots, 0\right]^T.
\]

Notice that, since we defined the monolithic systems discretizing (27) as a Kronecker product with the (quasi) Toeplitz matrices in Section 3.1, we still retain a (quasi) block Toeplitz structure even for the PDE case. We can then build our block circulant preconditioner.

### 3.2.2 Block circulant preconditioning

Analogously to what was done in Section 3.1, the following block circulant preconditioners are built for the systems (36),

\[
M^{\text{CD}_2} = \Phi^{\text{SV}_2} \otimes I^{N_t} - r^2 \left( \Phi_k^{\text{SV}_2} \otimes K^{\text{CD}_2} \right),
\]

and

\[
M^{\text{CI}_2} = \Phi^{\text{SV}_2} \otimes I^{N_t} - \frac{r^2}{4} \left( I^{N_t} + 2I^{N_t} + I^{N_t}_{-1} \right) \otimes K^{\text{CD}_2},
\]

(38)

where \( I^{N_t}_{-1} \) is the circular lower shift matrix, that is the circulant counterpart to \( I^{N_t}_k \), containing ones both on the \( i \)th subdiagonal and the \((N_t - i)\)th superdiagonal. As described in Section 2, these preconditioners can be (block) diagonalized.
using a (block) FFT. For instance, for $SV_2$ we have:

$$M^{SV_2} = (F_{N_t} \otimes I_{N_x}) \begin{pmatrix} I_{N_t} \otimes I_{N_x} \\ \Lambda_{1}^{N_t} \otimes (2 I_{N_x} + r^2 K_{CD}) \\ + \Lambda_{2}^{N_t} \otimes I_{N_x} \end{pmatrix} (F_{N_t}^* \otimes I_{N_x})^*,$$  

(39)

where $\Lambda_{n}^{N_t}$ are the diagonal matrices containing the $N_t$th roots of unity, sampled at frequency $n$: more explicitly, we have $\Lambda_{n}^{N_t} = \text{diag}(\{e^{2\pi i nk/N_t}\}_{k=0}^{N_t-1}) = F_{N_t}^N_{n} F_{N_t}^{-N_{n}}$. In order to apply the inverse of (39) to a space–time vector $b \in \mathbb{R}^{N_x N_t}$, we then need to perform the following sequence of operations:

1. **Apply $F_{N_t}^* \otimes I_{N_x}$.** This corresponds to computing a total of $N_x$ different FFTs on the coefficients of $b$. Each of these FFTs can be performed independently. The signals that need to be Fourier transformed, denoted as $s_n \in \mathbb{R}^{N_x}$, are each of length $N_t$ and are recovered by collecting every $N_x$ coefficients of $b$, starting from the $n$th coefficient:

$$[s_n] = [b]_{N_x+n}, \quad \text{with} \quad n = 0, \ldots , N_x - 1, \quad \text{and} \quad i = 0, \ldots , N_t - 1. \quad (40)$$

2. **Invert $A^{SV_2}$.** As already pointed out, this matrix is block diagonal. As a consequence, this step requires solving for $N_t$ independent systems (the blocks on the diagonal), using $[s]_{i[N_x N_t]}^{N_t+1N_x-1}, n = 0, \ldots , N_t - 1$, as right-hand sides. Looking at (39) we can see how each of these solves requires inverting a complex-shifted discrete Laplacian operator.

3. **Apply $F_{N_t} \otimes I_{N_x}$.** The last step is equivalent to the first, except $N_x$ inverse FFTs (iFFT) must be performed.

### 3.2.3 Cost analysis of circulant preconditioning versus time-stepping

Canonically, approximate solutions to (27) are recovered via time-stepping, which corresponds to applying block forward substitution to space–time matrices such as (36). This is an inherently sequential procedure which, at least for implicit methods, requires inverting a shifted Laplacian at each time step. Assuming inverting the Laplacian operator has an associated cost of $O(N_s)$, the overall procedure has a complexity of $O(N_s C(N_s))$. Given the availability of optimal solvers for this kind of operators, we can expect $O(N_s)$ to scale linearly with $N_s$, with spatial parallelization eventually coming to our aid in reducing the required computational time.

Conversely, our procedure increases dramatically the global cost required for solving systems with the coefficient matrix (36). For each GMRES iteration, in fact, we need to multiply the space–time residual by the matrix (36), an operation which by itself has a complexity of $O(N_s N_t)$. On top of this, applying the preconditioner involves $N_x$ independent FFT and iFFT, of overall complexity $O(N_s N_t \log(N_t))$, as well as inverting $N_t$ systems involving a complex-shifted Laplacian. The latter can be safely assumed to have a cost comparable to $O(N_s)$ (in fact, even for this operator optimal solvers are available), which makes the application of the preconditioner a procedure with an overall complexity of $O(N_s N_t \log(N_t) + N_t C(N_s))$. Unlike time-stepping, however, solution via GMRES with circulant preconditioning exposes parallelization along the temporal domain, as stated in Section 2.1. This can be exploited to reduce computational time at each step of the GMRES iteration. Firstly, when multiplying by the space–time matrix, if we have $N_t$ processors available, each assigned to a specific instant (or block-row in the matrix), then the computational time necessary is effectively reduced from $O(N_s N_t)$ to $O(N_s)$, if we assume that time-to-solution is directly proportional to complexity. Here we are neglecting the fact that, if each processor contains information only pertaining to a specific instant, a certain amount of communication must occur in order to perform multiplication by the space–time matrix. This overload is however...
limited, since the space–time matrix has a small band of size \( s_1 + s_2 \), implying that this exchange of information only involves \( s_1 + s_2 - 1 \) processors at most, and in particular its cost per processor is independent on \( N_t \). Secondly, when inverting the complex-shifted Laplacians, time similarly reduces form \( \mathcal{O}(N_s C(N_s)) \) to \( \mathcal{O}(C(N_s)) \), since again each system can be inverted independently (and no communication must occur among processors). Thirdly, a similar reasoning can be followed also when applying the Fourier transforms: not only the FFTs on each of the \( N_s \) signals are independent, meaning that any additional processor for spatial parallelization can still be employed to attack the \( N_s \) factor in the \( \mathcal{O}(N_s N_t \log(N_t)) \) complexity, but also parallelization over time is perfectly feasible. Most of the FFT algorithms available involve a step consisting of operations acting over the whole signal vector;\(^{33,34}\) the operations on each element are independent and can hence be parallelized,\(^{35,36}\) ideally dropping computational time from \( \mathcal{O}(N_s N_t \log(N_t)) \) to \( \mathcal{O}(N_s \log(N_t)) \). The actual performance of the parallel FFT depends heavily on how communication among processors is organized, and on the architecture of the system considered;\(^{37}\) since FFT represents the bottleneck in the application of the circulant preconditioner, any efficient parallel implementation will need to be fine-tuned in order to take this into account. This is however beyond the scope of this article, which rather wishes to focus on the efficacy of the preconditioner when applied to space–time systems.

To summarize, we have on the one hand time-stepping: a sure-fire direct procedure which recovers the solution to a space–time system in a number of operations scaling as \( \sim N_t C(N_s) \); on the other, we have an iterative procedure whose computational time per iteration scales as \( \sim C(N_s) + N_s \log(N_t) \). The difference between the two regimes allows some room for speed-up, provided that the number of necessary iterations remains bounded, and particularly that it does not scale with \( N_s \), nor with \( N_t \). Even in the PDE case, we can still invoke Theorem 1 to secure ourselves from the latter, since it confirms that the maximum theoretical number of iterations to convergence is independent of the number of time steps taken. Unfortunately, though, Theorem 1 does not shield us from the former, since in principle the number of iterations can grow with the size of the blocks composing the matrices (36), that is, the number of spatial nodes \( N_s \) considered in the discretization. In practice, however, and particularly under certain regimes, this seems not to be the case. Details on this are shown in the following.

### 3.2.4 Results

In this section, we report the results from the application of GMRES for the solution of systems involving (36) as coefficient matrix, using the corresponding (38) as right-preconditioners. In particular, and in light of the considerations made in the previous paragraph, we are interested in its performance in terms of number of iterations to convergence.

For all of our experiments, GMRES is set to a tolerance of \( 10^{-10} \), starting from a null initial guess. We consider the unit square, \( T = L = 1 \), as a domain for the PDE (27). As an initial condition, we choose a shifted gaussian:

\[
\tilde{u}^0(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left( -\frac{1}{2\sigma^2} \left( x - \frac{L}{2} \right)^2 \right) - c, 
\]

where \( c \) is picked so that \( \tilde{u}^0(0) = \tilde{u}^0(L) = 0 \), and \( \sigma^2 = 0.002 \); for the derivative, we simply take \( \tilde{u}^1(x) = 0 \). The solutions to the systems involving the complex-shifted Laplacian are recovered at each iteration using the \texttt{backslash} command in MATLAB. The code used for the experiments is publicly available in its Git repository.\(^{38}\) We test the two different schemes presented above, for a variety of orders of accuracy. Results are presented in Table 2 for SV discretizations of the temporal derivative, and in Table 3 for CI\(_2\) discretizations.

The actual number of iterations to convergence remains well below the limit indicated by Theorem 1 for all cases considered. We can notice however that this number is still far from being independent from the spatial mesh size. The convergence behavior when increasing \( N_t \) is somewhat hard to grasp, but shows a slight increase in iterations count as \( N_t \) gets higher. Increasing \( N_s \) has a similar impact, and also slows convergence. This is expected from the statement of Theorem 1: \( N_s \) determines the size of the blocks composing (7), and hence directly affects the rank of the perturbation (10). Also increasing the order of accuracy of the time discretizations used has a similar negative effect on performance. This is also expected from Theorem 1: in this case it is the actual number of blocks in (10), rather than their dimension, that is directly affected.

The SV\(_1\) scheme (top-left in Table 2) deserves a separate discussion: this scheme produces a solution which presents a noticeable degree of numerical diffusion, as reported in the literature\(^{17}\) and shown in Figure 1. There, we can see how the solution profile with SV\(_1\) (at the bottom) is visibly smeared out, contrasting with the result from leapfrog (top and middle):
### TABLE 2  Number of iterations to convergence for GMRES, right-preconditioned with (7), applied to the solution of systems arising from the discretization of (27)

| $N_t \rightarrow N_x$ | 120 | 240 | 480 | 960 | 1920 | 3840 | $(s_1 + s_2) \cdot N_x + 1$ |
|-----------------------|-----|-----|-----|-----|------|------|--------------------------|
| 80                    | 15  | 17  | 26  | 39  | 70   | 136  | 161                     |
| 160                   | 16  | 19  | 30  | 53  | 95   | 321  | 321                     |
| 320                   | 18  | 21  | 33  | 76  | 641  | 1281 | 1281                    |
| 640                   | 19  | 22  | 35  |     |      |      |                          |
| 1280                  | 21  | 23  |     |      |      |      |                          |
| 2560                  |     |     |      |      |      |      |                          |

**Note:** Different number of spatial and temporal nodes considered ($N_x$ and $N_t$, respectively). Time derivative approximated using SV, and space derivative approximated using CD of matching orders of accuracy. The rightmost column in each table reports the theoretical upper bound from Theorem 1; a cross identifies a simulation which did not converge due to memory requirements becoming too severe. Values for $N_x > N_t$ are not reported, as they violate the CFL condition, and give rise to unstable solutions: convergence results are very poor in those cases. The different colors in the cells background provide an indication of the accuracy of the solution for the test case considered: the $L_2$ error in space–time with respect to the analytical solution decreases as we move to darker shades, becoming $<10^{-4}$, $<10^{-6}$, and $<10^{-8}$, respectively.

| $N_t \rightarrow N_x$ | 120 | 240 | 480 | 960 | 1920 | 3840 | $(s_1 + s_2) \cdot N_x + 1$ |
|-----------------------|-----|-----|-----|-----|------|------|--------------------------|
| 80                    | 49  | 67  | 91  | 105 | 114  | 118  | 161                     |
| 160                   | 46  | 55  | 81  | 110 | 153  | 321  | 321                     |
| 320                   | 30  | 31  | 33  | 46  | 641  | 1281 | 1281                    |
| 640                   | 25  | 25  | 27  |     |      |      |                          |
| 1280                  |     |     |     |      |      |      |                          |
| 2560                  |     |     |     |      |      |      |                          |

### TABLE 3  Number of iterations to convergence for GMRES, right-preconditioned with (7), applied to the solution of systems arising from the discretization of (27)

| $N_t \rightarrow N_x$ | 120 | 240 | 480 | 960 | 1920 | 3840 | $(s_1 + s_2) \cdot N_x + 1$ |
|-----------------------|-----|-----|-----|-----|------|------|--------------------------|
| 80                    | 181 | 229 | 205 | 212 | 222  | 225  | 401                     |
| 160                   | 248 | 244 | 186 | 169 | 164  | 801  | 801                     |
| 320                   | 263 | 348 | 264 | 253 |      | 1601 | 1601                    |
| 640                   | 310 | 465 | 368 |     |      |      | 3201                    |
| 1280                  |     |     |     |      |      |      | 6401                    |
| 2560                  |     |     |     |      |      |      | 12801                   |

**Note:** Different number of spatial and temporal nodes considered ($N_x$ and $N_t$, respectively). Time derivatives approximated using CI$_2$, space derivatives using CD with various accuracies. The rightmost column in each table reports the theoretical upper bound from Theorem 1. The different shades indicate the accuracy with respect to the analytical solution as in Table 2.
FIGURE 1  Surface plots and heatmaps of examples of numerical solutions recovered from the experiments. Top: SV₂ discretization in time, CD₂ in space (explicit Leapfrog); middle: implicit leapfrog, still with CD₂ space discretization; bottom: the pair SV₁ – CD₂ is used instead. \( N_t = 240, N_x = 160 \), and other parameters as in Tables 2 and 3. In the surface plots, time progresses moving toward the reader.
the difference is clear by comparing the shapes of the final negative peaks in the three surface plots, or by looking at how the heatmap at the bottom appears more blurred. The fact that diffusion is present seems to aid with the convergence of the scheme, a feature shared with other parallel-in-time integration methods.\(^{39,40}\) In fact, we can see from Table 2 that the number of iterations to convergence remains reasonably small, even when the meshes are refined. However, the recovered solution is of very poor quality: even with the finest meshes used, the error with respect to the analytical solution remains above \(10^{-4}\).

The background colors of the cells in Table 2 draw a clearer picture of the performances of each scheme, by providing an indication of the accuracy of the recovered approximation: moving to darker shades, the numerical solution reaches an approximation error of \(10^{-4}\), \(10^{-6}\), and \(10^{-8}\), respectively, although at around \(10^{-8}\) the convergence of high-order schemes tends to stall, likely because of round-off errors becoming relevant, which makes the results less informative for this last range. The first lesson we can extract from this is that there is no advantage in using a low-order scheme over a high-order one, so long as they share the same stencil: by comparing the tables relative to \(SV_2\) and \(SV_4\) (two schemes both with a stencil of size 3), we can see that \(SV_4\) achieves a much smaller error with respect to \(SV_2\) on the same meshes, and often even with a smaller number of iterations. The latter is possibly due to the negative impact of the extra perturbation introduced to the Toeplitz system when accounting for the initial conditions, since we used a ghost-node approach for \(SV_2\). As the order of the method increases, it becomes harder to make such strong statements, but if the stencil grows together with the order, then low-order, small-stencil schemes seem to be preferable. For example, \(SV_4\) reaches an error of \(10^{-6}\) with 25 iterations on a \(640 \times 960\) mesh, while \(SV_5\) needs 65 iterations (roughly 2.5 as many) on a mesh with 1/4 as many degrees of freedom, and \(SV_7\) uses 312 (or roughly 12.5 as many) iterations on a mesh with only 1/16 as many degrees of freedom. Given how the cost per iteration scales \(\text{linearly}\) in \(N_x\) but \(\text{logarithmically}\) in \(N_t\) (as per the analysis conducted in the previous paragraph), \(SV_5\) is of comparable cost with respect to \(SV_4\), but \(SV_7\) far exceeds it, and is hence not worth employing for the same target accuracy.

The nature of the initial conditions also has an impact on the speed of convergence. We test this by running a series of experiments where we progressively narrow the gaussian used as an initial condition. The number of iterations to convergence increases as the gaussian becomes steeper, regardless of the discretization used, as is shown in Table 4. This is not entirely surprising: with steeper profiles, the range of relevant modes in the solution extends to higher frequencies, and we can expect to require a richer Krylov subspace in order to properly approximate them. An extreme example of the converse can be shown by picking a single mode as an initial condition, of any frequency. This gives rise to a stationary wave, which is particularly simple to track, since the spatial and the temporal components of the solution factorize: the solution at each instant is nothing but the initial condition, opportunely rescaled. In this case, the circulant preconditioner is extremely effective, as it is capable of building a Krylov subspace which captures the degrees of freedom of the solution in very few iterations: only 2, for all schemes of order of accuracy 2 or smaller, as was reported already in the literature.\(^{17,24}\) This is however a peculiar feature of the circulant preconditioner, limited to the very specific combination of parameters employed: if we vary the type of spatial discretization used (or if we add a forcing term, or vary the initial guess for the GMRES algorithm), we cannot expect such a dramatic convergence anymore, as shown in the first row of Table 4.
4 | CONCLUSION

Parallel-in-time computational methods for evolutionary equations have been studied over a long period and there are now several successful schemes which are effective in speeding up the solution of parabolic PDEs. By contrast, hyperbolic PDE problems seem to present a bigger challenge for parallelization.

Here we have investigated a recently suggested approach based on circulant preconditioning for the block Toeplitz all-at-once (or monolithic) system that arises with the use of a constant time step for the discretization of the linear wave equation. The use of regular time steps would seem more appropriate in the context of such wave problems, rather than for dissipative problems. We have shown how a result of Chan, Potts, and Steidl guarantees convergence (termination) of GMRES in a small number of iterations—the number being independent of the number of time steps—in ideal cases with this preconditioning methodology. Parallel application of such iterative schemes has been well covered in the literature. We have also presented numerical results which bear out this theory and demonstrate its potential utility even where exact structures are broken, for example, by the application of initial conditions.

Further work will be required to generalize to variable coefficient and nonlinear hyperbolic equations. In both of these cases, the structure of the space–time system $A_N$ is bound to steer further away from that of a (block) Toeplitz matrix. A way for us to still make use of the circulant preconditioner (8), consists in designing an effective projection operator, so to trace $A_N$ back to the space of (block) Toeplitz matrices. Possibly the most straightforward way to achieve this is given by diagonally averaging the entries of $A_N$. It remains however not the only way by far, and indeed identifying the most apt one remains a matter for future research.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in the repository https://gitlab.com/fdanieli/all-at-once_code.

ORCID

Federico Danieli https://orcid.org/0000-0002-0866-2485

REFERENCES

1. Gander MJ. 50 years of time parallel time integration. In: Carraro T, Geiger M, Körkel S, Rannacher R, editors. Multiple shooting and time domain decomposition methods. Contributions in Mathematical and Computational Sciences. Berlin, Germany: Springer International Publishing; 2014. p. 69–113.
2. Ong BW, Schroder JB. Applications of time parallelization. Comput Vis Sci. 2020;23(11).
3. Lions JL, Maday Y, Turinici G. Résolution d’EDP par un schéma en temps $\ll$pararéel$\gg$. C R Acad Sci I-Math. 2001;332(7):661–8.
4. Emmett M, Minion ML. Toward an efficient parallel in time method for partial differential equations. Comm Appl Math Comp Sci. 2012;7:105–32.
5. Falgout R, Friedhoff S, Kolev T, MacLachlan S, Schroder J. Parallel time integration with multigrid. SIAM J Sci Comput. 2014;36(6):C635–61.
6. Gander MJ, Neumüller M. Analysis of a new space-time parallel multigrid algorithm for parabolic problems. SIAM J Sci Comput. 2016;38(6):A2173–208.
7. Horton G, Vandewalle S. A space-time multigrid method for parabolic partial differential equations. SIAM J Sci Comput. 1995;16(4):848–64.
8. Dai X, Maday Y. Stable parareal in time method for first- and second-order hyperbolic systems. SIAM J Sci Comput. 2013;35(1):A52–78.
9. Howse A, Sterck H, Falgout R, MacLachlan S, Schroder J. Parallel-in-time multigrid with adaptive spatial coarsening for the linear advection and inviscid burgers equations. SIAM J Sci Comput. 2019;41(1):A538–65.
10. Nielsen AS, Brunner G, Hesthaven JS. Communication-aware adaptive parareal with application to a nonlinear hyperbolic system of partial differential equations. J Comput Phys. 2017;345:365–383.
11. Ruprecht D. Wave propagation characteristics of Parareal. Comput Vis Sci. 2018;19(1):1–17.
12. Southworth B, Mantueller T, McCormick S, Munzenmaier S, Ruge J. Nonsymmetric reduction-based algebraic multigrid. SIAM J Sci Comput. 2019;41(10):S242–68.
13. Danieli F, MacLachlan SP. Multigrid reduction in time for non-linear hyperbolic equations. Comput Vis Sci. 2019; Submitted for publication. https://arxiv.org/abs/2104.09404.
14. Kulchytska-Ruchka I, Schöps S. Efficient parallel-in-time solution of time-periodic problems using a multi-harmonic coarse grid correction. ArXiv; 2019. abs/1908.05245.
15. Strang G. A proposal for toeplitz matrix calculations. Stud Appl Math. 1986;74(2):171–6.
16. McDonald E, Pestana J, Wathen A. Preconditioning and iterative solution of all-at-once systems for evolutionary partial differential equations. SIAM J Sci Comput. 2018;40(2):A1012–33.
17. Goddard A, Wathen A. A note on parallel preconditioning for all-at-once evolutionary PDEs. Electron Trans Numer Anal. 2019;51:135–50.
18. Chan RH, Potts D, Steidl G. Preconditioners for non-hermitian toeplitz systems. Numer Linear Algebra Appl. 2001;8(2):83–98.
19. Gander MJ, Halpern L, Ryan J, Tran TTB. A direct solver for time parallelization. In: Dickopf T, Gander MJ, Halpern L, Krause R, Pavarino LF, editors. Domain decomposition methods in science and engineering. Lect Notes in Computer Science. Volume XXII. Cham, Switzerland: Springer International Publishing; 2016. p. 491–9.
20. Gander MJ, Halpern L. Time parallelization for nonlinear problems based on diagonalization. In: Lee CO, Cai XC, Keyes DE, Kim HH, Klawonn A, Park EJ, et al., editors. Domain decomposition methods in science and engineering. Lect Notes in Computer Science. Volume XXII. Cham, Switzerland: Springer International Publishing; 2017. p. 163–70.
21. Chan RHF, Jin XQ. An introduction to iterative toeplitz solvers. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2007.
22. Greenbaum A, Pták V, Strakous Z. Any nonincreasing convergence curve is possible for GMRES. SIAM J Matrix Anal Appl. 1996;07(17):465–9.
23. Antonucci G. Iterative solution of evolutionary PDEs via all-at-once methods. Oxford, UK: University of Oxford; 2017.
24. Goddard A. An implementation of the all-at-once method to evolutionary PDEs. Oxford, UK: University of Oxford; 2018.
25. Hairer E, Nørsett SP, Wanner G. Solving ordinary differential equations I: nonstiff problems. SPRs Company. Berlin/Heidelberg, Germany: Springer; 2008.
26. LeVeque RJ. Finite difference methods for ordinary and partial differential equations. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2007.
27. Liu J, Wu SL. A fast block α-circulant preconditioner for all-at-once system from wave equations. SIAM J Matrix Anal Appl. 2020;09.
28. Fornberg B. Generation of finite difference formulas on arbitrarily spaced grids. Math Comput. 1988;51(184):699–706.
29. Lippold G, Mitchell AR. The finite difference method in partial differential equations. Chichester, New York, Brisbane, Toronto: John Wiley & Sons; 1980.
30. Trottenberg U, Oosterlee CW, Schuller A. Multigrid. London: Elsevier Science; 2000.
31. MacLachlan S, Oosterlee K. Algebraic multigrid solvers for complex-valued matrices. SIAM J Sci Comput. 2008;jan;30:1548–71.
32. Erlangga YA, Oosterlee CW, Vuik C. A novel multigrid based preconditioner for heterogeneous Helmholtz problems. SIAM J Sci Comput. 2006;27(4):1471–92.
33. Cooley J, Tukey JW. An algorithm for the machine calculation of complex Fourier series. Math Comput. 1965;19:297–301.
34. Burrsus CS, Frigo M, Johnson GS. Fast Fourier transforms. London: Samurai Media Limited; 2018.
35. Swartztrauber PN. Multiprocessor FFTs. Parallel Comput. 1987;5(1):197–210. Proceedings of the International Conference on Vector and Parallel Computing - Issues in Applied Research and Development.
36. Jamieson L, Mueller PT, Siegel H. FFT algorithms for SIMD parallel processing systems. J Parallel Distr Com. 1986;3:48–71.
37. Frigo M, Johnson SG. The design and implementation of FFTW3. Proc IEEE. 2005;93(2):216–31.
38. Danieli F. Repository all-at-once_code; https://gitlab.com/fdanieli/all-at-once_code. Accessed on December 2020.
39. Dobrev V, Kolev T, Petersson N, Schroder J. Two-level convergence theory for multigrid reduction in time (MGRIT). SIAM J Sci Comput. 2017;39(3):S501–27.
40. Bal G. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In: Barth TJ, Griebel M, Keyes DE, Nieminen RM, Roose D, Schlick T, et al., editors. Domain decomposition methods in science and engineering. Lect Notes in Computer Science. Berlin/Heidelberg, Germany: Springer; 2005. p. 425–32.

How to cite this article: Danieli F, Wathen AJ. All-at-once solution of linear wave equations. Numer Linear Algebra Appl. 2021;e2386. https://doi.org/10.1002/nla.2386