IMA Journal of Numerical Analysis (2015) Page 1 of 29
doi:10.1093/imanum/drv021

A high-order time-parallel scheme for solving wave propagation problems via the direct construction of an approximate time-evolution operator

T. S. Haut*
Center For Nonlinear Studies (CNLS), Los Alamos National Laboratory, Los Alamos, USA
*Corresponding author: terryhaut@lanl.gov terryhaut@gmail.com

T. Babb and P. G. Martinsson
Department of Applied Mathematics, University of Colorado at Boulder, Boulder, USA

and

B. A. Wingate
Department of Mathematics, University of Exeter, Devon, UK

[Received on 16 February 2014; revised on 19 April 2015]

The manuscript presents a technique for efficiently solving the classical wave equation, the shallow water equations, and, more generally, equations of the form $\partial u/\partial t = \mathcal{L}u$, where $\mathcal{L}$ is a skew-Hermitian differential operator. The idea is to explicitly construct an approximation to the time-evolution operator $\exp(\tau\mathcal{L})$ for a relatively large time-step τ. Recently developed techniques for approximating oscillatory scalar functions by rational functions, and accelerated algorithms for computing functions of discretized differential operators are exploited. Principal advantages of the proposed method include: stability even for large time-steps, the possibility to parallelize in time over many characteristic wavelengths and large speed-ups over existing methods in situations where simulation over long times are required. Numerical examples involving the 2D rotating shallow water equations and the 2D wave equation in an inhomogenous medium are presented, and the method is compared to the 4th order Runge–Kutta (RK4) method and to the use of Chebyshev polynomials. The new method achieved high accuracy over long-time intervals, and with speeds that are orders of magnitude faster than both RK4 and the use of Chebyshev polynomials.

Keywords: time-stepping methods; optimal rational approximations; parallel-in-time; direct solver.

1. Introduction

1.1 *Problem formulation*

We present a technique for solving a class of linear hyperbolic problems

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t}(\mathbf{x},t) = \mathcal{L}\mathbf{u}(\mathbf{x},t), \quad \mathbf{x}\in\Omega,\ t>0, \\ \mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}) \quad \mathbf{x}\in\Omega. \end{cases} \tag{1.1}$$

Here $\mathbf{u} \in: L^2(\Omega)$ is a possibly vector-valued function, $\mathcal{L} : L^2(\Omega) \to L^2(\Omega)$ is a skew-Hermitian differential operator (see the end of this section for the method's scope), and $\exp(\tau\mathcal{L}) : L^2(\Omega) \to L^2(\Omega)$ denotes the propagator associated with (1.1). The technique is demonstrated on the 2D rotating shallow water equations, as well as the variable coefficient wave equation.

The basic approach is classical, and involves the construction of a rational approximation to the time evolution operator $\exp(\tau\mathcal{L})$ in the form

$$\exp(\tau\mathcal{L}) \approx \sum_{m=-M}^{M} b_m(\tau\mathcal{L} - \alpha_m)^{-1},$$

where the time-step τ is fixed in advance and M scales linearly in τ. Once the time-step τ has been fixed, an approximate solution at times $\tau, 2\tau, 3\tau, \ldots$ can be obtained via repeated application of the approximate time-stepping operator, since $\exp(n\tau\mathcal{L}) = (\exp(\tau\mathcal{L}))^n$. The computational profile of the method is that it takes a moderate amount of work to construct the initial approximation to $\exp(\tau\mathcal{L})$, but once it has been built, it can be applied very rapidly, even for large τ.

The efficiency of the proposed scheme is enabled by (i) a modified version of Damle *et al.* (2013) for constructing near optimal rational approximations to oscillatory functions such as e^{ix} over arbitrarily long intervals, and by (ii) the development (Martinsson, 2013) of a high-order accurate and stable method for pre-computing approximations to operators of the form $(\tau\mathcal{L} - \alpha_m)^{-1}$. The near optimality of the rational approximations ensures that the number $2M + 1$ of terms needed for a given accuracy is typically much smaller than standard methods that rely on polynomial or rational approximations of $\mathcal{L}$.

Although our main contribution is in combining the techniques in Damle *et al.* (2013) and Martinsson (2013) in order to yield an efficient and time-parallel means of applying the operator exponential, we also develop in this paper several key technical advances in each of the component algorithms. First, the sub-optimal rational approximations developed here require substantially fewer poles than in Damle *et al.* (2013) (and, in fact, are very close to optimal in the L^∞ norm). Secondly, the direct solver in Martinsson (2013) is modified in order to allow body loads.

The proposed scheme has several advantages over typical methods, including the apparent absence of stability constraints on the time step τ in relation to the spatial discretization, the ability to parallelize in the time variable over many characteristic wavelengths (in addition to any spatial parallelization), and great acceleration when integrating equation (1.1) for long times or for multiple initial conditions (e.g., when employing an exponential integrator on a nonlinear evolution equation, cf. Section 5). A drawback of the scheme is that it is more memory intensive than standard techniques. Another potential limitation is that, since the proposed technique relies on a spectral element discretization, the initial condition needs to be sufficiently smooth $\mathbf{u}(\mathbf{x}, 0)$ in order to achieve spectral accuracy with respect to p-refinement.

The ability to solve (1.1) in a time-parallel manner can be used to construct efficient parallel-in-time schemes for the fully nonlinear evolution equations in the presence of time scale separation (see Haut & Wingate, 2014). This extra source of parallelism can be useful if the speedup due to spatial parallelization saturates. In addition, even when the speedup from spatial parallelization is not saturated, the ability to parallelize with space-time blocks can have efficiency benefits. In fact, it is expected that the cost of communication versus computation decreases with a space-time decomposition relative to a purely spatial decomposition (heuristically, the 'surface-to-volume' ratio of a domain decomposition decreases in higher dimensions). As far as parallelization of the direct solver, the algorithm involves applying, at a number of levels that is logarithmic in the spatial grid size, small dense matrices that can all be applied independently of each other. Since applying many small dense matrices in parallel is amenable to efficient parallelization, we expect good parallel scaling for the direct solver; however, this issue needs to be explored further.

We restrict the scope of this paper to when the application $(\tau\mathcal{L} - \alpha_m)^{-1}\mathbf{u}_0$ can be reduced to the solution of an elliptic-type partial differential equation (PDE) for one of the unknown variables. This situation arises in geophysical fluid applications (among others), including the rotating primitive equations

that are used for climate simulations. However, the direct solver presented in Section 2 is quite general, and in principle can be extended to first-order linear systems of hyperbolic PDEs with little modification (though such an extension is speculative and, in particular, has not been tried).

1.2 *Time discretization*

In order to time-discretize (1.1), we fix a time-step τ (the choice of which is discussed shortly), a requested precision $0 < \delta < 1$, and 'band-width' $\Lambda \in (0, \infty)$ which specifies the spatial resolution (in effect, the scheme will accurately capture eigenmodes of $\mathcal{L}$ whose eigenvalues λ satisfy $|\lambda| \leqslant \Lambda$). We then use an improved version of the scheme of Damle *et al.* (2013) to construct a rational function,

$$R_M(\mathrm{i}x) = \sum_{m=-M}^{M} \frac{b_m}{(\mathrm{i}x - \alpha_m)}, \tag{1.2}$$

such that

$$|e^{\mathrm{i}x} - R_M(\mathrm{i}x)| \leqslant \delta, \quad x \in [-\tau\Lambda, \tau\Lambda], \tag{1.3}$$

and

$$|R_M(\mathrm{i}x)| \leqslant 1, \quad x \in \mathbb{R}. \tag{1.4}$$

It now follows from (1.3) and (1.4) that if we approximate $\exp(t\mathcal{L})$ by $R_M(\tau\mathcal{L})$, the approximation error satisfies

$$\left\| e^{\tau\mathcal{L}}\mathbf{u}_0 - \sum_{m=-M}^{M} b_m(\tau\mathcal{L} - \alpha_m)^{-1}\mathbf{u}_0 \right\|_2 \leqslant \delta\|\mathbf{u}_0\|_2 + 2\|\mathbf{u}_0 - \mathcal{P}_\Lambda\mathbf{u}_0\|_2, \tag{1.5}$$

where $\mathcal{P}_\Lambda$ projects functions onto the subspace spanned by eigenvectors of $\mathcal{L}$ with modulus at most Λ. Here the only property of $\mathcal{L}$ that we use is that $\mathcal{L}$ is skew-Hermitian, and hence has a complete spectral decomposition with a purely imaginary spectrum.

In the absence of spatial discretization errors, the bound (1.4) ensures that the repeated application of $R_M(\tau\mathcal{L})$ is stable on the entire imaginary axis. It also turns out that the number $2M + 1$ of terms needed in the rational approximation in (1.3) is close to optimally small (for the given accuracy δ). It is important to point out that, when the above temporal discretization is coupled with the spatial discretization discussed in Section 2, stability analysis of the time-stepping method requires an error analysis for the direct solver discusses in Section 2.3, which is nontrivial and beyond the scope of this paper. We simply note that the stability of the time-stepping scheme is substantiated via extensive numerical experiments and using a broad range of different time steps τ. In all of these numerical experiments, we have not observed any instabilities, including when we choose large τ (resulting in hundreds of terms in (1.2)) and nonsmooth initial conditions.

The scheme described above allows a great deal of freedom in the choice of the time step τ. While classical methods typically require the time step to be a small fraction of the characteristic wavelength, we have freedom to let τ cover a large number of characteristic wavelengths. Therefore, the scheme is well suited to parallelization in time, since all the inverse operators in the approximation of the operator exponential can be applied independently. In fact, the only constraint on the size of τ is on the memory available to store the representations of the inverse operators (as explained in Section 1.3, the memory required for each inverse scales linearly in the number of spatial discretization parameters, up to a logarithmic factor).

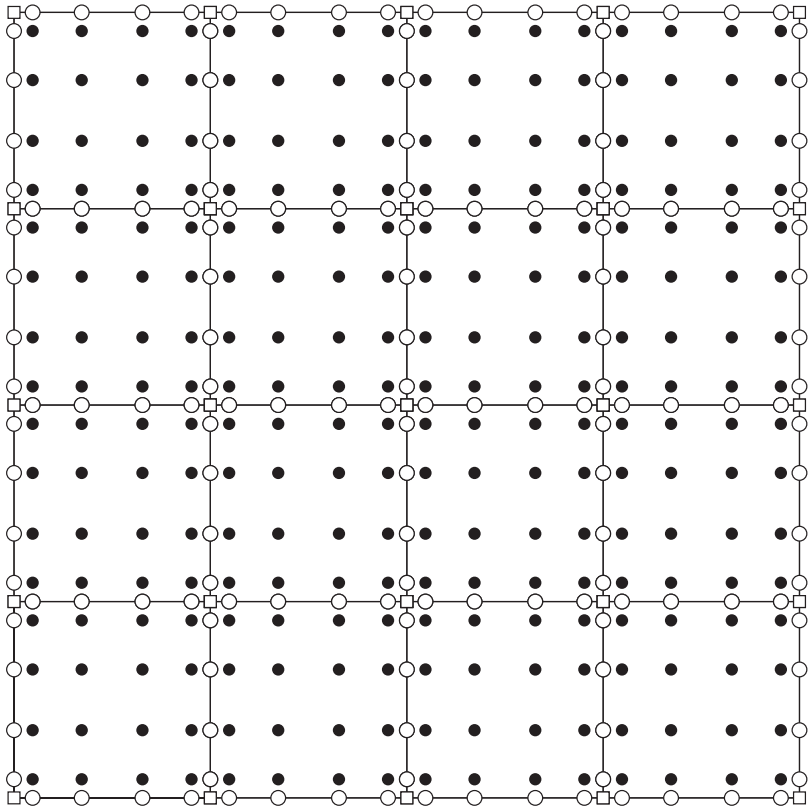

FIG. 1. Illustration of the grid of points $\{\mathbf{x}_j\}_{j=1}^N$ introduced to discretize (2.12) in Section 2.2. The figure shows a simplified case involving 4×4 squares, each holding a 6×6 local tensor product grid of Chebyshev nodes. The PDE (2.12) is enforced via collocation using spectral differentiation on each small square at all solid ('internal') nodes. At the hollow ('boundary') nodes, continuity of normal fluxes is enforced.

In serial, the efficiency of this approach is relatively insensitive to the number M of terms used, since the number of poles per wavelength in (1.5) is constant (and just larger than the lower bound imposed Nyquist constraint). However, since the M solution operators can be applied independently of one another, this approach allows the ability to parallelize the computation in time over a large number of characteristic wavelengths. In contrast, standard methods for applying the operator exponential are inherently serial in the time variable.

1.3 *Pre-computation of rational functions of* $\mathcal{L}$

The time discretization technique described in Section 1.2 requires us to build explicit approximations to differential operators on the domain Ω such as $(\tau\mathcal{L} - \alpha_m)^{-1}$. We do this using a variation of the technique described in Martinsson (2013). A variety of different domains can be handled, but for simplicity, suppose that Ω is a rectangle. The idea is to tessellate Ω into a collection of smaller rectangles, and to put down a tensor product grid of Chebyshev nodes on each rectangle, as shown in Fig. 1. A function is

represented via tabulation on the nodes, and then $\mathcal{L}$ is discretized via standard spectral collocation techniques on each patch. The patches are glued together by enforcing continuity of both function values and normal derivatives. This discretization results in a block sparse coefficient matrix, which can rapidly be inverted via a procedure very similar to the classical nested dissection technique of George (1973). The resulting inverse is dense but 'data-sparse,' which is to say that it has internal structure that allows us to store and apply it efficiently.

In order to describe the computational cost of the direct solver, let N denote the number of nodes in the spatial discretization. For a problem in two dimensions, the 'build stage' of the proposed scheme constructs $2M + 1$ data-sparse matrices $\{\mathsf{A}_m\}_{m=-M}^{M}$ of size $N \times N$, where each A_m approximates $(\tau\mathcal{L} - \alpha_m)^{-1}$. The build stage has asymptotic cost $\mathcal{O}(MN^{1.5})$, and storing the matrices requires $O(MN\log(N))$ memory. The cost of applying a matrix A_m is $\mathcal{O}(N\log(N))$. (We remark that the cost of building the matrices $\{\mathsf{A}_m\}_{m=-M}^{M}$ can often be accelerated to optimal $\mathcal{O}(MN)$ complexity (Gillman & Martinsson, 2014), but since the pre-factor in the $\mathcal{O}(MN^{1.5})$ bound is quite small, such acceleration would have negligible benefit for the problem sizes under consideration here.) Section 2 describes the inversion procedure in more detail.

We remark that the spatial discretization procedure we use does not explicitly enforce that the discrete operator is exactly skew-Hermitian. The fact that the spatial discretization is done to very high accuracy means that it is in practice very nearly so, and numerical experiments also indicate that the scheme as a whole is stable in every regime where it was tested. However, a rigorous investigation of the numerical stability of the scheme is currently lacking and is the subject of future investigation.

1.4 *Comparison to existing approaches*

The approach of using proper rational approximations for applying matrix exponentials has a long history. In the context of operators with negative spectrum (e.g., for parabolic-type PDEs), many authors have discussed how to compute efficient rational approximations to the decaying exponential e^{-x}, including using Cauchy's integral formula coupled with Talbot quadrature (cf. Schmelzer & Trefethen, 2007b), and optimal rational approximations via the Carathéodory–Fejer method (cf. Schmelzer & Trefethen, 2007b) or the Remez algorithm (Cody *et al.*, 1969). However, such methods are less effective (or not applicable) when applied to approximating oscillatory functions such as $e^{\mathrm{i}x}$ over long intervals. For computing functions of parabolic-type linear operators, the approach of combining rational approximations and compressed representations of the solution operators using so-called $\mathcal{H}$-matrices has been proposed in Gavrilyuk *et al.* (2005).

Common approaches for applying the exponential of skew-Hermitian operators include high-order time-stepping methods, scaling-and-squaring coupled with Padé approximations (cf. Higham, 2005) or Chebyshev polynomials (cf. Bergamaschi & Vianello, 2000) and polynomial or rational Krylov methods (cf. Hochbruck & Lubich, 1997; Güttel, 2013). Krylov methods, in particular, have enjoyed enormous success due to their ability to handle very large problem sizes and their favourable approximation properties (see the review article Hochbruck & Ostermann, 2010). We note that rational Krylov methods also exhibit near optimal approximation properties (see Güttel, 2013). Note that all these methods iteratively build up rational or polynomial approximations to the operator exponential, and correspondingly approximate the spectrum $e^{\mathrm{i}\omega_n\tau}$ of $e^{\tau\mathcal{L}}$ with polynomials or rationals. Therefore, the near optimality of (1.2) and the speed of applying the inverse operators in (1.5) will generally translate into high efficiency relative to standard methods (or, in the case of rational Krylov methods, comparable efficiency). Comparing the proposed method with these more standard approaches, one major advantage is that

the method can be trivially parallelized in time over many characteristic wavelengths. The ability to parallelize in the time variable is of particular relevance to large-scale simulations in geophysical fluid applications, where the speedup from spatial parallelization alone is beginning to saturate.

In addition to approaches that rely on polynomial or rational approximations, let us mention two alternative approaches for time-stepping on wave propagation problems. The authors in Beylkin & Sandberg (2005) combine separated representations of multi-dimensional operators, partitioned low rank compressions of matrices, and (near) optimal quadrature nodes for band-limited functions, in order to compute compressed representations of the operator exponential over $1-2$ characteristic wavelengths. Along different lines, the authors in Demanet & Ying (2009) use wave atoms to construct compressed representations of the (short time) operator exponential, and in particular can bypass the CFL constraint.

1.5 *Outline of manuscript*

The paper is organized as follows. In Section 2, we briefly describe the direct solver in Martinsson (2013). We then discuss in Section 3 a technique for constructing efficient rational approximations of general functions, and specialize to the case of approximating the exponential e^{ix} and the phi-functions for exponential integrators (Cox & Matthews, 2002). In Section 4, we present applications of the method for both the 2D rotating shallow water equations and the 2D wave equation in inhomogenous medium. In particular, we compare the accuracy and efficiency of this approach against 4th order Runge–Kutta (RK4) and the Chebyshev polynomial method (in our comparisons, we use the same spectral element discretization). Finally, Appendix A contains error bounds for the rational approximations constructed here.

2. Spectral element discretization

This section describes how to efficiently compute a highly accurate approximation to the inverse operator $(\mathcal{L}-\alpha)^{-1}$, where $\mathcal{L}$ is a skew-Hermitian operator. As mentioned in the introduction, we restrict our discussion to environments where application of the inverse can be reformulated as a scalar elliptic problem. This reformulation procedure is illustrated for the classical wave equation and for the shallow water equations in Section 2.1. Section 2.2 describes a high-order multidomain spectral discretization procedure for the elliptic equation. Section 2.3 describes a direct solver for the system of linear equations arising upon discretization.

2.1 *Reformulation as an elliptic problem*

In many situations of practical interest, the task of solving a hyperbolic equation $(\mathcal{L}-\alpha)u=f$, where $\mathcal{L}$ is a skew-Hermitian operator, can be reformulated as an associated elliptic problem. In this section, we illustrate the idea via two representative examples. Example 1 is of particular relevance to geophysical fluid applications, which serve as a major motivation of this algorithm.

EXAMPLE 1 (the shallow water equation) We consider the rotating shallow water equations in the square domain $\mathbf{x}\in[0,1]\times[0,1]$ with periodic boundary conditions:

$$\begin{aligned}\mathbf{v}_t &= -fJ\mathbf{v}+\nabla\eta,\\ \eta_t &= \nabla\cdot\mathbf{v},\end{aligned} \tag{2.1}$$

where $\mathbf{v}(\mathbf{x},t) = (v_1(\mathbf{x},t), v_2(\mathbf{x},t))$ denotes the fluid velocity, $\eta(\mathbf{x},t)$ denotes perturbed surface elevation, f is the (constant) Coriolis frequency and

$$J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

For simplicity, we assume that the prescribed initial velocity components $v_j(\mathbf{x},0) \in L^2([0,1] \times [0,1])$ and the initial surface elevation $\eta(\mathbf{x},t) \in L^2([0,1])$ are continuous (so that pointwise evaluation on the spectral element grid is well-defined). This condition can likely be relaxed to piecewise smooth initial conditions, but is outside the scope of the current paper.

We write system (2.1) in the form

$$\mathbf{u}_t = \mathcal{L}\mathbf{u},$$

where

$$\mathcal{L}\begin{pmatrix} \mathbf{v} \\ \eta \end{pmatrix} = \begin{pmatrix} -fJ\mathbf{v} + \nabla\eta \\ \nabla \cdot \mathbf{v} \end{pmatrix}. \tag{2.2}$$

We note that the method generalizes to nonconstant coefficient f and more general domains and boundary conditions in a transparent manner, and is of particular relevance for a spectral element discretization on the cubed sphere. In fact, a spatial domain that is composed of a union of squares can be easily be handled by the direct solver. By smoothly mapping curvilinear patches near the boundary to square patches, in theory more general domains can be handled without much difficulty. Of particular relevance to geophysical fluid applications (which serve as a big motivation), we plan to explore the proposed method for the rotating shallow water equations on the cubed sphere. In this setup, six square patches are mapped to patches on the sphere. The direct solver algorithm remains essentially unchanged, except that the elliptic equation associated with the RSW equations now contain nonconstant coefficients that reflect the underlying geometry, which is also easily handled by the proposed scheme.

In order to apply the method in this paper, we use the standard fact (cf. Paldor & Sigalov, 2011) that if

$$(\mathcal{L} - \alpha)\begin{pmatrix} \mathbf{v} \\ \eta \end{pmatrix} = \begin{pmatrix} \mathbf{v}_0 \\ \eta_0 \end{pmatrix}, \tag{2.3}$$

then η satisfies the elliptic equation

$$\nabla \cdot (\mathcal{A}_\alpha \nabla\eta) - \alpha\eta = \eta_0 + \nabla \cdot \mathcal{A}_\alpha \mathbf{v}_0. \tag{2.4}$$

Here $\mathcal{A}_\alpha$ is defined by

$$\mathcal{A}_\alpha = \frac{1}{\alpha^2 + f^2}\begin{pmatrix} \alpha & -f \\ f & \alpha \end{pmatrix}.$$

Once η is computed, $\mathbf{v}$ can be obtained directly,

$$\mathbf{v} = -\mathcal{A}_\alpha \mathbf{v}_0 + \mathcal{A}_\alpha \nabla\eta. \tag{2.5}$$

When f is constant, equation (2.4) reduces to

$$(\Delta - (\alpha^2 + f^2))\eta = \frac{\alpha^2 + f^2}{\alpha}(\eta_0 + \nabla \cdot (\mathcal{A}_\alpha \mathbf{v}_0)). \tag{2.6}$$

EXAMPLE 2 (the wave equation) Consider the wave propagation problem

$$u_{tt} = \kappa \Delta u, \quad \mathbf{x} \in [0,1] \times [0,1], \tag{2.7}$$

where $\kappa(\mathbf{x}) \geqslant \kappa_0 > 0$ is a smooth uniformly positive function, the initial conditions $u(\mathbf{x},0)$ and $u_t(\mathbf{x},0)$ are prescribed square integrable and continuous functions, and periodic boundary conditions are used.

In order to apply the method in this paper, we reformulate (2.7) as a first-order system in both time and space by defining $v = u_t$, $w = u_x$ and $z = u_y$. Then we have that

$$\begin{pmatrix} w_t \\ z_t \\ v_t \end{pmatrix} = \begin{pmatrix} 0 & 0 & \partial_x \\ 0 & 0 & \partial_y \\ \kappa\partial_x & \kappa\partial_y & 0 \end{pmatrix} \begin{pmatrix} w \\ z \\ v \end{pmatrix}, \tag{2.8}$$

with initial conditions

$$v(\mathbf{x},0) = u_0(\mathbf{x}), \quad w(\mathbf{x},0) = \frac{\partial u_0}{\partial x}(\mathbf{x}), \quad z(\mathbf{x},0) = \frac{\partial u_0}{\partial y}(\mathbf{x}).$$

Here the scalar function u to the original system (2.7) can be recovered after the final time step by solving the elliptic equation $\Delta u = w_x + z_y$.

To apply the method in this paper, we compute the solution to

$$(\mathcal{L} - \alpha) \begin{pmatrix} w \\ z \\ v \end{pmatrix} = \begin{pmatrix} v_x - \alpha w \\ v_y - \alpha z \\ \kappa(w_x + z_y) - \alpha v \end{pmatrix} = \begin{pmatrix} w_0 \\ z_0 \\ v_0 \end{pmatrix} \tag{2.9}$$

as follows. First, solving for w and z in terms of v,

$$w = \frac{1}{\alpha}(v_x - w_0), \quad z = \frac{1}{\alpha}(v_y - z_0), \tag{2.10}$$

it is straightfoward to show that

$$(\Delta - \alpha^2\kappa^{-1})v = \alpha\kappa^{-1}v_0 + \frac{\partial w_0}{\partial x} + \frac{\partial z_0}{\partial y}. \tag{2.11}$$

Once v is known, w and z can then be computed directly via (2.10).

2.2 *Discretization*

In this section, we describe a high-order accurate discretization scheme for elliptic boundary value problems (BVPs) such as (2.6) and (2.11) which arise in the solution of hyperbolic evolution equations. Specifically, we describe the solver for a BVP of the form

$$\mathcal{B}u(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{2.12}$$

where $\mathcal{B}$ is an elliptic differential operator. To keep things simple, we consider only square domains $\Omega = [0,1]^2$, but the solver can easily be generalized to other domains. The solver we use is described in detail in Martinsson (2015), our aim here is merely to give a high-level conceptual description.

The PDE (2.12) is discretized using a multidomain spectral collocation method. Specifically, we split the square Ω into a large number of smaller squares (or rectangles), and then put down a tensor product grid of $p \times p$ Chebyshev nodes on each small square, see Fig. 1. The parameter p is chosen so that dense computations involving matrices of size $p^2 \times p^2$ are cheap ($p = 20$ is often a good choice). Let $\{\mathbf{x}_j\}_{j=1}^N$ denote the total set of nodes. Our approximation to the solution u of (2.12) is then represented by a vector $\mathbf{u} \in \mathbb{C}^N$, where the j'th entry is simply an approximation to the function value at node $\mathbf{x}_j$, so that $\mathbf{u}_j \approx u(\mathbf{x}_j)$. The discrete approximation to (2.12) then takes the form

$$\mathbf{B}\mathbf{u} = \mathbf{f}, \tag{2.13}$$

where $\mathbf{B}$ is an $N \times N$ matrix. The j'th row of (2.13) is associated with a collocation condition for node $\mathbf{x}_j$. For all j for which $\mathbf{x}_j$ is a node in the *interior* of a small square (filled circles in Fig. 1), we directly enforce (2.12) by replacing all differentiation operators by spectral differentiation operators on the local $p \times p$ tensor product grid. For all j for which $\mathbf{x}_j$ lies on a *boundary* between two squares (hollow squares in Fig. 1), we enforce that normal fluxes across the boundary are continuous, where the fluxes from each side of the boundary are evaluated via spectral differentiation on the two patches (corner nodes need special treatment; see Martinsson, 2015).

2.3 *Direct solver*

The discrete linear system (2.13) arising from discretization of (2.12) is block-sparse. Since it has the typical sparsity pattern of a matrix discretizing a 2D differential operator, it is possible to compute its LU factorization in $O(N^{1.5})$ operations using a nested dissection ordering of the nodes (Duff *et al.*, 1986; George, 1973) that minimizes fill-in. Once the LU-factors have been computed, the cost of a linear solve is $O(N \log N)$. In the numerical computations presented in Section 4, we use a slight variation of the nested-dissection algorithm that was introduced in Martinsson (2013) for the case of homogeneous equations. The extension to the situation involving body loads is straightforward, see Martinsson (2015).

We note that by exploiting internal structure in the dense sub-matrices that appear in the factors of $\mathbf{B}$ as the factorization proceeds, the complexity of both the factorization and the solve stages can often be reduced to optimal $O(N)$ complexity (Gillman & Martinsson, 2014). However, for the problem sizes considered in this manuscript, there would be little practical gain to implementing this more complex algorithm.

3. Constructing rational approximations

We now discuss how to construct efficient rational approximations to smooth functions $f(x)$ defined on the real line. For concreteness, we consider approximating the phi-functions

$$\varphi_0(x) = e^{\mathrm{i}x}, \quad \varphi_1(x) = \frac{e^{\mathrm{i}x} - 1}{\mathrm{i}x}, \quad \varphi_2(x) = \frac{e^{\mathrm{i}x} - \mathrm{i}x - 1}{(\mathrm{i}x)^2},$$

that arise for high-order exponential integrators (cf. Schmelzer & Trefethen, 2007a and the review article Hochbruck & Ostermann, 2010). By considering the real and imaginary components separately, we assume that $f(x)$ is real-valued (it turns out that the poles in the approximation will be the same for the real and imaginary components, as explained shortly). The construction proceeds in two steps; the second step is actually a pre-computation and needs only be done once, but is presented last for clarity. First, we construct an approximation to $f(x)$ by sums of shifted Gaussians $\psi_h(x) = (4\pi)^{-1/2} e^{-x^2/(4h^2)}$

(see Section 3.1 for details),

$$\left| f(x) - \sum_{m=-M}^{M} b_m \psi_h(x + mh) \right| \leqslant \delta_1, \quad -\Lambda \leqslant x \leqslant \Lambda. \tag{3.1}$$

Here h is inversely proportional to the bandlimit of $f(x)$, and M controls the interval Λ over which the approximation is valid (roughly $|x| \lesssim Mh$). When $f(x) = e^{\mathrm{i}x}$, the coefficients are explicitly given by $b_m = (\hat{\psi}_h(1)/h)e^{-2\pi \mathrm{i} mh}$, and the approximation is remarkably accurate (see (3.6) for error bounds). Secondly, using the approach in Damle *et al.* (2013), a rational approximation to $\psi_1(x) = (4\pi)^{-1/2}e^{-x^2/4}$ is constructed over the real line (see Section 3.2 for details),

$$\left| \psi_1(x) - 2\,\mathrm{Re}\left(\sum_{j=-L}^{L} \frac{a_j}{\mathrm{i}x - (\mu + \mathrm{i}j)} \right) \right| \leqslant \delta_2, \quad x \in \mathbb{R}. \tag{3.2}$$

Note that the imaginary parts of the poles in the above approximation are integer multiples $j = 0, \pm 1, \ldots, \pm L$. For $L = 11$, we construct μ and coefficients a_j such that the L^∞ approximation error δ_2 satisfies $\delta_2 < 10^{-12}$ (see Table 1). Finally, combining (3.1) and (3.2), we obtain a rational approximation to $f(x)$,

$$\left| f(x) - 2\,\mathrm{Re}\left(\sum_{n=-M-L}^{M+L} \frac{c_n}{\mathrm{i}x - h(\mu + \mathrm{i}n)} \right) \right| \leqslant \delta_1 + 2(M + L)\delta_2. \tag{3.3}$$

Here the coefficients c_n are given by

$$c_n = h \sum_{k=L_1}^{L_2} a_k b_{n-k},$$

where

$$L_1(n) = \max(-L, n - M), \quad L_2(n) = \min(L, n + M).$$

Importantly, constructing the rational approximation (3.2) to $\psi(x)$ needs only be done once. In particular, once μ and the coefficients a_j are pre-computed, rational approximations to general functions $f(x)$ over arbitrarily long-spatial intervals can be obtained with minimal effort, as discussed in Section 3.1. We present μ, and the coefficients a_j, $j = -11, \ldots, 11$, in Table 1, which are sufficient to yield an L^∞ error $\delta_1 \approx 7 \times 10^{-13}$ in (3.2).

Using the reduction algorithm in Haut & Beylkin (2012), we find that the rational approximation constructed for $e^{\mathrm{i}x}$ is close to optimal in the L^∞ norm, for a given accuracy δ and spatial cutoff A. In fact, the construction in this paper uses only 1.2 times more poles than the near optimal rational approximation obtained from Haut & Beylkin (2012) when $\delta = 10^{-10}$ and $\Lambda = 56\pi$, which we use in our numerical experiments. We chose $\Lambda = 56\pi$ in the numerical experiments to demonstrate that the computation can in theory be parallelized over hundreds of characteristic wavelengths, but this choice for Λ is otherwise arbitrary and can be taken smaller or larger depending on the application. We note that the residues corresponding to this near optimal approximation can be very large and, for this reason, we prefer to use the sub-optimal approximation instead.

As clarified in Sections 3.1 and 3.2, the same poles can be used to approximate multiple functions with the same bandlimit. For example, we can use the same poles to approximate all functions $e^{2\pi \mathrm{i}tx}$, for $0 \leqslant t \leqslant 1$, since all these functions have bandlimit less than or equal to $e^{2\pi \mathrm{i}x}$; the dependence on

t is only through the coefficients, which are given explicitly by $b_m = (\hat{\psi}_h(t)/h)e^{-2\pi i n t h}$. In particular, the poles $\alpha_m = h(\mu + im)$ are independent of t and yield uniformly accurate approximations to e^{itx} on the same interval $[-\Lambda, \Lambda]$. This observation enables the efficient computation of multiple operator exponentials $e^{s_k \mathcal{L}}\mathbf{u}_0$, for $s_k = tk/L$, using the same computed solutions $(t\mathcal{L} - \alpha_m)^{-1}\mathbf{u}_0$, $m = 1, \ldots, M$. A similar comment applies to the phi-functions from exponential integrators.

Generally, any rational approximation to e^{ix} (or more general functions) must share the same number of zeros within the interval of interest; in particular, since the rational approximation can be expressed as a quotient of polynomials, it is therefore subject to the Nyquist constraint. However, one advantage of this approximation method is that it allows efficient rational approximations of functions that are spatially localized. In fact, since the approximation (3.1) involves highly localized Gaussians, the subsequent rational approximations are able to represent spatially localized functions as well as highly oscillatory functions using (perhaps a subset) of the same collection of poles. This allows the ability to take advantage of spectral gaps (e.g., from scale separation between fast and slow waves) and possibly bypass the Nyquist constraint under certain circumstances.

3.1 *Gaussian approximations to a general function*

We discuss how to construct the approximation (3.1). To do so, we choose h small enough that the Fourier transform $\hat{f}(\xi)$ is zero (or approximately so) outside the interval $[-1/(2h), 1/(2h)]$. Then we can expand $\hat{f}(\xi)/\hat{\psi}_h(\xi)$ in a Fourier series,

$$\frac{\hat{f}(\xi)}{\hat{\psi}_h(\xi)} = \sum_{m=-\infty}^{\infty} c_m e^{2\pi i m h \xi}, \tag{3.4}$$

where

$$c_m = h \int_{-1/(2h)}^{1/(2h)} e^{-2\pi i m h \xi} \frac{\hat{f}(\xi)}{\hat{\psi}_h(\xi)} \, d\xi.$$

Transforming (3.4) back to the spatial domain, we have that

$$f(x) = \sum_{m=-\infty}^{\infty} c_m \psi_h(x + mh).$$

Note that the functions $\psi_h(x + mh)$ are tightly localized in space, and truncating the above series from $-M$ to M yields accurate approximations for $-(M - b)hx < x < (M - b)hx$, where $b > 0$ is a small number that is related to the decay of $\psi_h(x)$. We remark that the authors in Maz'ya & Schmidt (1996) discuss a related method of constructing quasi-interpolating representations via sums of Gaussians (see Maz'ya & Schmidt, 2007 for a comprehensive survey).

Specializing to the case when $f(x) = e^{2\pi i x}$, we have that $\hat{f}(\xi) = \delta(\xi - 1)$, and so the coefficients c_m are given by

$$c_m = \frac{h}{\hat{\psi}_h(1)} e^{-2\pi i m h}. \tag{3.5}$$

Similarly, for functions $\varphi_1(x)$ and $\varphi_2(x)$, the coefficients c_m can be obtained numerically using the fact that

$$\hat{\varphi}_1(\xi) = \begin{cases} 2\pi, & -\dfrac{1}{2\pi} \leqslant \xi \leqslant 0, \\ 0, & \text{otherwise.} \end{cases}$$

and

$$\hat{\varphi}_2(\xi) = \begin{cases} (2\pi)^2 \left(\xi + \dfrac{1}{2\pi}\right), & -\dfrac{1}{2\pi} \leqslant \xi \leqslant 0, \\ 0, & \text{otherwise.} \end{cases}$$

For example, the coefficients c_m, e.g., $\phi_1(x)$ can be computed via discretization of the integral,

$$c_m = h \int_{-1/(2\pi)}^{0} e^{-2\pi \mathrm{i} m h \xi} \frac{e^{-2\pi \mathrm{i} m h \xi}}{\hat{\psi}_h(\xi)} \, \mathrm{d}\xi.$$

In Fig. 2, we plot the error,

$$\left| \varphi_j(x) - \sum_{m=-\infty}^{\infty} c_{m,j} \psi(x + mh) \right|,$$

for the phi-functions $\varphi_1(x)$ and $\varphi_2(x)$, where we choose $h = 1$ and $M = 200$; note that the choice of h corresponds to the bandlimit of $\varphi_j(x)$. As shown in Fig. 2, the error is smaller than $\approx 3 \times 10^{-13}$ for all $-191 \leqslant x \leqslant 191$, and is shown to begin to rise at the ends of the intervals, which are close to Mh. This behaviour can be understood by noting that

$$\left| \varphi_j(x) - \sum_{m=-M}^{M} c_{m,j} \psi_1(x + m) \right| \leqslant \sum_{|m|>M} |c_{m,j}| \psi_1(x + m),$$

where we used that the support of $\hat{\varphi}_j$ is contained in $[-\frac{1}{2}, 1.2]$. Since the functions $\psi_1(x + m)$ for $m > M$ decay rapidly away from $x = -m$, the error from truncation is negligible when $|x| \leqslant (M - m_0)$ and $m_0 = \mathcal{O}(1)$.

We remark that, for the function $e^{\mathrm{i}x}$, it can be shown (see Appendix) that the approximation for $e^{\mathrm{i}x}$ satisfies

$$\left| e^{\mathrm{i}x} - \sum_{m=-M}^{M} c_m \psi_h(x + mh) \right| \leqslant \frac{1}{\hat{\psi}_h(1)} \left(\sum_{k \neq 0} \hat{\psi}_h\left(\frac{k}{h}\right) + \sum_{|m|>M} \psi_h(x + mh) \right), \tag{3.6}$$

where c_m is defined in (3.5). We see that the first sum is negligible, e.g., $h \lesssim 1$, owing to the tight frequency localization of ψ_h. Similarly, the second sum is negligible when $|x| \leqslant (M - m_0)h$ and $m_0 = \mathcal{O}(1)$, owing to the tight spatial localization of ψ.

3.2 *Rational approximation to a Gaussian*

We now discuss how to construct the approximation (3.2).

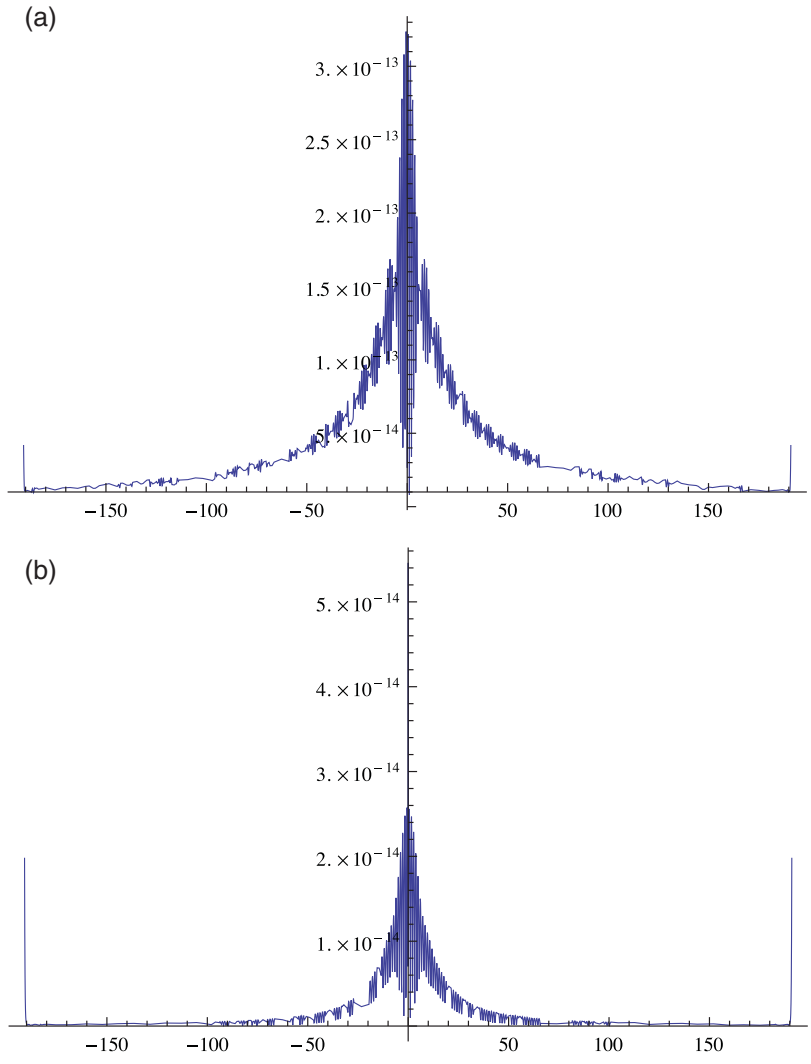

FIG. 2. The absolute error in the Gaussian approximations of $\varphi_j(x)$ for (a) $j = 1$ and (b) $j = 2$, using $h = 1$ and $M = 200$.

To do so, we first use Adamyan–Arov–Krein (AAK) theory (see Damle *et al.*, 2013 for details) to construct a near optimal rational approximation,

$$\left| \frac{1}{\sqrt{4\pi}} e^{-x^2/4} - \mathrm{Re}\left(\sum_{j=1}^{N} \frac{b_j}{\mathrm{i}x + \alpha_j} \right) \right| \leqslant \delta.$$

For an accuracy of $\delta \approx 10^{-13}$, 13 poles γ_j are required.

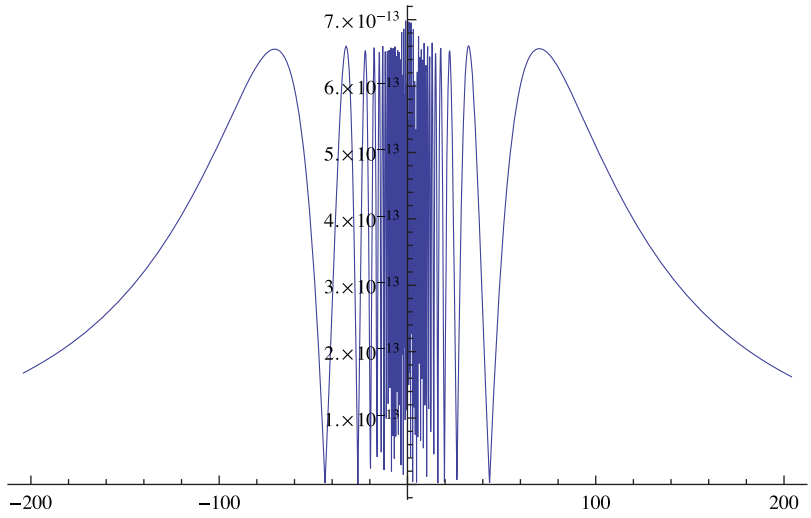

FIG. 3. Error in the rational approximation (3.2) to $e^{-x^2/4}$.

Setting $\mu = \min_j \mathrm{Re}(\alpha_j)$, we next look for a rational approximation to $(4\pi)^{-1/2}e^{-x^2/4}$ of the form

$$R(x) = \mathrm{Re}\left(\sum_{j=-L}^{L} \frac{a_j}{\mathrm{i}x + \mu + \mathrm{i}j}\right), \tag{3.7}$$

where we take $L = 11$. We find the coefficients a_j by minimizing the L^∞ error

$$\left\| \frac{1}{\sqrt{4\pi}} e^{-x^2/4} - \mathrm{Re}\left(\sum_{j=-L}^{L} \frac{a_j}{\mathrm{i}x_n + \mu + \mathrm{i}j}\right) \right\|_\infty,$$

where the points $x_n \in [-30, 30]$ are chosen to be more sparsely distributed outside the numerical support of $e^{-x^2/4}$; the interval $[-30, 30]$ is found experimentally to yield high accuracy for the approximation over the entire real line. Finding the coefficients a_j, $j = -L, \ldots, L$, that minimize the L^∞ error can be cast as a convex optimization problem, and a standard algorithm can be used (we use Mathematica). The resulting approximation error is shown in Fig. 3; the error remains less than $\approx 7 \times 10^{-13}$ for all $x \in \mathbb{R}$.

We display the real number μ, and the coefficients a_j, $j = 1, \ldots, 11$ in Table 1. In particular, these numbers are the only parameters that are needed in order to construct rational approximations to general functions on spatial intervals of any size.

In Fig. 4, we show the resulting rational approximations of $\cos(2\pi x)$ and $\sin(2\pi x)$, which use the same 172 complex–conjugate pairs of poles; the L^∞ error is seen to be $\approx 10^{-10}$ over the interval $-28 \leqslant x \leqslant 28$.

TABLE 1 *Coefficients $a_j, j = -11, \ldots, 11$ and number μ, in the rational approximation (3.7)*

$\mu = -4.315321510875024,$
$a_{-11} = (-1.0845749544592896 \times 10^{-7}, 2.77075431662228 \times 10^{-8}),$
$a_{-10} = (1.858753344202957 \times 10^{-8}, -9.105375434750162 \times 10^{-7}),$
$a_{-9} = (3.6743713227243024 \times 10^{-6}, 7.073284346322969 \times 10^{-7}),$
$a_{-8} = (-2.7990058083347696 \times 10^{-6}, 0.0000112564827639346),$
$a_{-7} = (0.000014918577548849352, -0.0000316278486761932),$
$a_{-6} = (-0.0010751767283285608, -0.00047282220513073084),$
$a_{-5} = (0.003816465653840016, 0.017839810396560574),$
$a_{-4} = (0.12124105653274578, -0.12327042473830248),$
$a_{-3} = (-0.9774980792734348, -0.1877130220537587),$
$a_{-2} = (1.3432866123333178, 3.2034715228495942),$
$a_{-1} = (4.072408546157305, -6.123755543580666),$
$a_0 = -9.442699917778205,$
$a_1 = (4.072408620272648, 6.123755841848161),$
$a_2 = (1.3432860877712938, -3.2034712658530275),$
$a_3 = (-0.9774985292598916, 0.18771238018072134),$
$a_4 = (0.1212417070363373, 0.12326987628935386),$
$a_5 = (0.003816972477033343, -0.017839242222443888),$
$a_6 = (-0.0010756025812659208, 0.0004731874917343858),$
$a_7 = (0.00001471375478909521 8, 0.000031358475831136815),$
$a_8 = (-2.659323898804944 \times 10^{-6}, -0.000011341571201752273),$
$a_9 = (3.6970377676364553 \times 10^{-6}, -6.517457477594937 \times 10^{-7}),$
$a_{10} = (3.883933649142257 \times 10^{-9}, 9.128496023863376 \times 10^{-7}),$
$a_{11} = (-1.0816457995911385 \times 10^{-7}, -2.954309729192276 \times 10^{-8})$

3.3 *Constructing rational approximation of modulus bounded by unity*

For our applications, it is important that the approximation to e^{ix} is bounded by unity on the real line. In particular, the Gaussian approximation for e^{ix} constructed in Section 3.1 has absolute value larger than one when $|x| \approx Mh$, and this can lead to instability in repeated applications of $e^{t\mathcal{L}}$.

The basic idea is to construct a rational function $S(\mathrm{i}x)$ that satisfies $S(\mathrm{i}x) \approx 1$ for $|x| \lesssim M_0 h$ and $S(\mathrm{i}x) \approx 0$ for $|x| \gtrsim M_0 h$. As long as M_0 is slightly less than M, the function $S(\mathrm{i}x)R_M(\mathrm{i}x)$ accurately approximates e^{ix} for $|x| \lesssim M_0 h$, and decays rapidly to zero for $|x| \gtrsim M_0 h$. Therefore, $|S(\mathrm{i}x)R_M(\mathrm{i}x)| \leqslant 1$ for all $x \in \mathbb{R}$, and repeated application of $S(t\mathcal{L})R_M(t\mathcal{L})\mathbf{u}_0$ is stable for all $t > 0$. In Fig. 5, we plot a rational filter that uses 33 complex–conjugate poles; we see that $|S(\mathrm{i}x) - 1| \approx 10^{-10}$ for $-28 \leqslant x \leqslant 28$.

Although the above approach results in a stable method, a slightly modified version can reduce the amount of computation by a factor of 2. This is motivated by the following simple observation: since $\mathbf{u}_0(\mathbf{x})$ is real-valued,

$$\overline{(t\mathcal{L} - \alpha)^{-1}\mathbf{u}_0} = (t\mathcal{L} - \bar{\alpha})^{-1}\mathbf{u}_0. \tag{3.8}$$

Recalling that the poles from Section 3.2 come in complex–conjugate pairs, only half the matrix inverses need to be pre-computed and applied if (3.8) is used. However, directly using (3.8) results in numerical instabilities, where small errors in the high frequencies are amplified after successive applications of $R_M(t\mathcal{L})\mathbf{u}_0$.

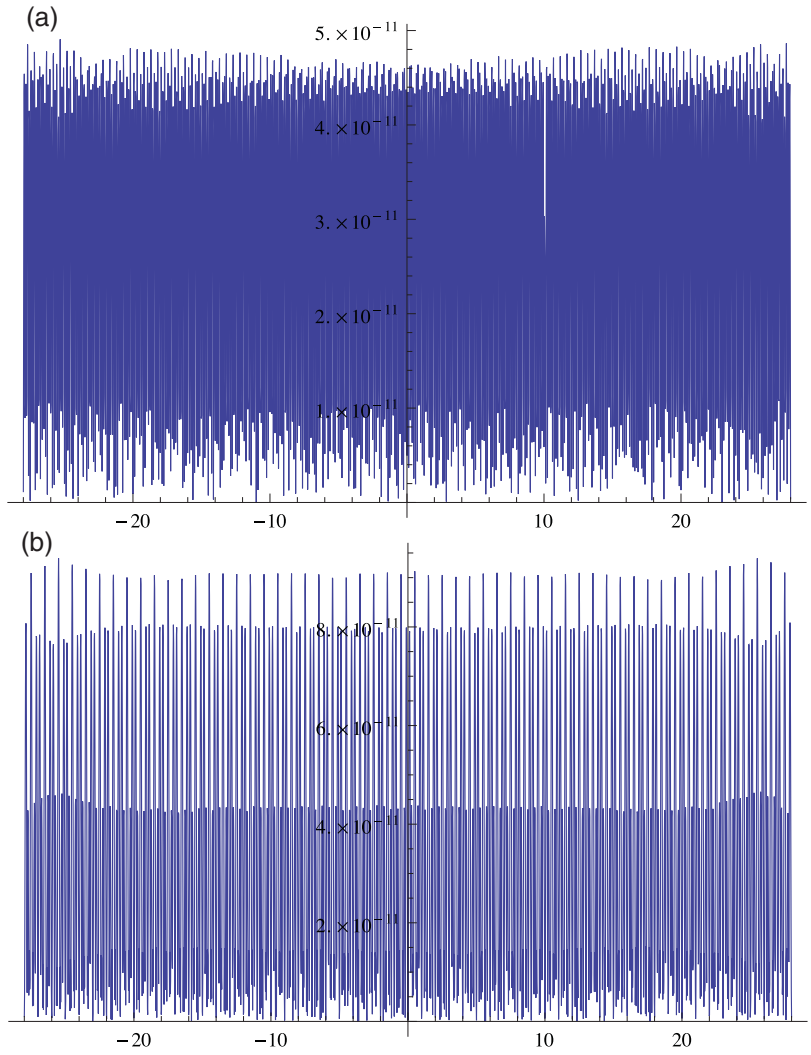

FIG. 4. Error in the rational approximations of $\sin(2\pi x)$ and $\cos(2\pi x)$ (plots (a) and (b)), for $-28 \leqslant x \leqslant 28$. These approximations use the same 172 pairs of complex–conjugate poles.

The fix is to eliminate the errors in the high frequency components by instead computing $S(k_0\Delta)R_M(t\mathcal{L})\mathbf{u}_0$, where the parameter k_0 is adjusted experimentally so that

(1) $\|S(k_0\Delta)\mathbf{u}_0 - \mathbf{u}_0\|_2$ is smaller than the desired approximation accuracy

(2) Decreasing k_0 by a factor of 2 results in the error $\|S(k_0\Delta)\mathbf{u}_0 - \mathbf{u}_0\|_2$ being larger than the desired approximation accuracy

The error and the stability of the time-stepping scheme appears relatively insensitive to the precise choice of k_0 and the above procedure has sufficed for all problems we have examined. We note that

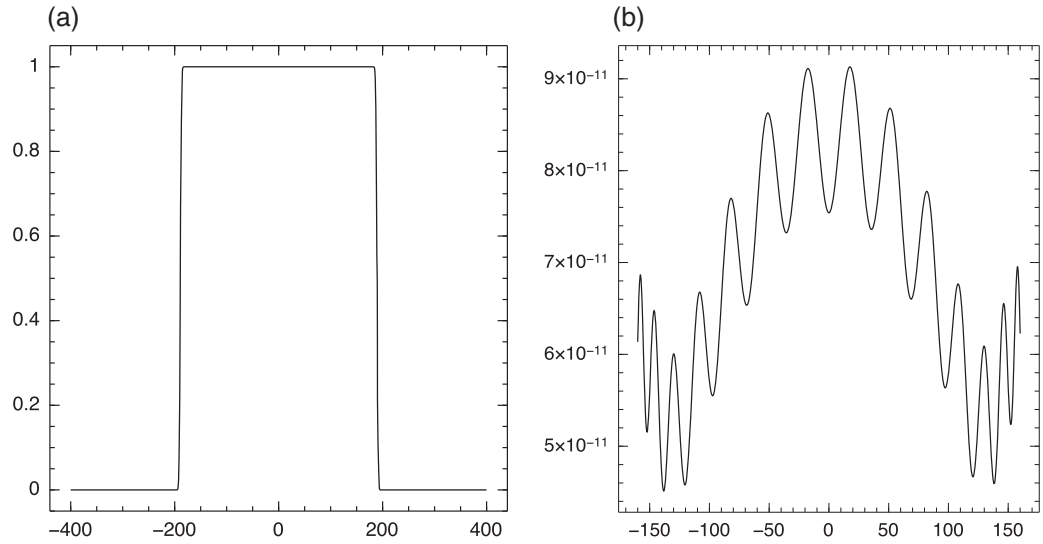

FIG. 5. (a) Plot of the rational filter function $S(\mathrm{i}x)$, for $-400 \leqslant x \leqslant 400$. (b) Plot of the difference $|S(\mathrm{i}x) - 1|$ for $-150 \leqslant x \leqslant 150$.

the transition region between $S(\mathrm{i}x) \approx 1$ and $S(\mathrm{i}x) \approx 0$ can be made arbitrarily small (see Fig. 5), and the rational function $S(\mathrm{i}x)$ uses a number of poles that scales logarithmically in the width of the transition region.

We now discuss how to construct the rational filter function $S(\mathrm{i}x)$:

$$S(\mathrm{i}x) = \mathrm{Re}\left(\sum_{j=1}^{M_1} \frac{d_j}{(\mathrm{i}x + \beta_j)}\right). \tag{3.9}$$

The poles β_m and residues d_m are explicitly given in Tables 2 and 3 for $M_1 = 33$ (see also Fig. 5).

To do so, we use that (see Müller & Varnhorn, 2007)

$$\left|\frac{1}{\hat{\psi}_h(1)} \sum_{m=-\infty}^{\infty} \psi_h(x + hm) - 1\right| \leqslant \frac{1}{h\hat{\psi}_h(1)} \sum_{k \neq 0} \hat{\psi}_h\left(\frac{k}{h}\right),$$

which follows from the Poisson summation formula. For $h \lesssim 1$, the right-hand side is negligible, owing to the tight frequency localization of $\hat{\psi}_h(\xi)$. Truncating the above sum and using the tight spatial localization of $\psi_h(x)$, we see that the function

$$\chi(x) = \sum_{m=-M_0}^{M_0} \psi_h(x + mh), \tag{3.10}$$

is approximately unity for $|x| \lesssim M_0 h$, and decays to zero rapidly when $|x| \gtrsim M_0 h$. It also holds out that $|\chi(x)| \leqslant 1$ for all $x \in \mathbb{R}$. Therefore, using the techniques from Sections 3.1 and 3.2, we construct a

TABLE 2 *Poles for rational filter function* $S(\mathrm{i}x)$, β_j, $j = 1, \ldots, 33$, *in* (3.9)

$\beta_1 = (-5.6815244593211665, 195.60368900644573)$,
$\beta_2 = (-5.681487525698976, -195.60354575365798)$,
$\beta_3 = (-5.790937378563278, 193.51258302488918)$,
$\beta_4 = (-5.790931808899799, -193.512407062321)$,
$\beta_5 = (-5.862236792305668, 191.74408220195133)$,
$\beta_6 = (-5.862237702715605, -191.743890714106)$,
$\beta_7 = (-5.936140551272466, 190.12359157219748)$,
$\beta_8 = (-5.936124270898454, -190.12341069616582)$,
$\beta_9 = (-189.4725566345281, 0.0005010358202330836)$,
$\beta_{10} = (-176.66776934247073, 68.32388925486636)$,
$\beta_{11} = (-176.6680762718217, -68.3230079401094)$,
$\beta_{12} = (-145.68006428301598, 120.86369765397113)$,
$\beta_{13} = (-145.68050951454478, -120.863074132019)$,
$\beta_{14} = (-110.26321457057725, 153.676839340607)$,
$\beta_{15} = (-110.26363743084794, -153.6764516160099)$,
$\beta_{16} = (-79.29081523941767, 171.57427345997775)$,
$\beta_{17} = (-79.29114472221063, -171.57404315017212)$,
$\beta_{18} = (-55.356508023495955, 180.6106832990567)$,
$\beta_{19} = (-55.356733683903755, -180.61054964388333)$,
$\beta_{20} = (-37.847789557552204, 184.98161469308596)$,
$\beta_{21} = (-37.847920735512524, -184.98155004038057)$,
$\beta_{22} = (-25.222413671409782, -187.02183399118556)$,
$\beta_{23} = (-25.22237138528123, 187.0218302467066)$,
$\beta_{24} = (-6.02552562063538, 188.52945956389644)$,
$\beta_{25} = (-6.025505490003158, -188.52931522046455)$,
$\beta_{26} = (-15.837104131666214, -187.83927148804779)$,
$\beta_{27} = (-15.83719317531402, 187.8391829927176)$,
$\beta_{28} = (-6.054693412832868, 186.81930649684463)$,
$\beta_{29} = (-6.054700825299065, -186.81918507624317)$,
$\beta_{30} = (-6.0164875510372156, 184.96674677593367)$,
$\beta_{31} = (-6.016529115859488, -184.96664278981225)$,
$\beta_{32} = (-5.9124443095601, 182.8128396205155)$,
$\beta_{33} = (-5.912519458298405, -182.81276452097833)$

rational approximation $Q(ix)$ to the function $\chi(x)$ in (3.10),

$$\left| Q(\mathrm{i}x) - \sum_{m=-M_0}^{M_0} \psi_h(x + mh) \right| \leqslant \delta, \quad x \in \mathbb{R}, \tag{3.11}$$

The number of poles required to represent the sub-optimal approximation for $Q(x)$ can be drastically reduced with the reduction algorithm (Haut & Beylkin, 2012), which produces another proper rational function $S(x)$ such that

$$|Q(\mathrm{i}x) - S(\mathrm{i}x)| \leqslant \delta_0, \quad x \in \mathbb{R},$$

TABLE 3 *Residues for rational filter function* $S(\mathrm{i}x), d_j, j = 1, \ldots, 33$, *in* (3.9)

$d_1 = (-0.005515883340470063, 0.0018078912091650061)$,
$d_2 = (-0.005517983400057653, -0.0018065984116659806)$,
$d_3 = (0.29517517913626257, 0.030850668188794006)$,
$d_4 = (0.2952473284303156, -0.0309522473436694)$,
$d_5 = (-2.8087936414479624, -0.1811594749911728)$,
$d_6 = (-2.809264947520908, 0.18217487582852943)$,
$d_7 = (9.005601902434998, -2.188432070313594)$,
$d_8 = (9.006566533922124, 2.1854340243226966)$,
$d_9 = (-11.388204088432527, 2.7476480737856203e - 5)$,
$d_{10} = (-9.90121707360217, 3.846338806959291)$,
$d_{11} = (-9.901252091929033, -3.8463011680536536)$,
$d_{12} = (-6.7340017287953335, 5.617362425311499)$,
$d_{13} = (-6.7340440687043035, -5.617355388135177)$,
$d_{14} = (-3.8615372397992824, 5.421083499590135)$,
$d_{15} = (-3.8615686391768222, -5.421094962675174)$,
$d_{16} = (-2.0029034444370817, 4.379034646612036)$,
$d_{17} = (-2.0029227879567775, -4.379051310390535)$,
$d_{18} = (-0.9837969850082168, 3.2617150351336717)$,
$d_{19} = (-0.9838093306384953, -3.261730953927478)$,
$d_{20} = (-0.467449411102599, 2.3543689422597613)$,
$d_{21} = (-0.4674597496668436, -2.354383201175641)$,
$d_{22} = (-0.2107304966321986, -1.7074746753873138)$,
$d_{23} = (-0.2107185642076128, 1.7074599077825507)$,
$d_{24} = (-10.983655336199284, 6.785929970040307)$,
$d_{25} = (-10.984308465853344, -6.782645956476531)$,
$d_{26} = (-0.050388404664131234, -1.3292002320672072)$,
$d_{27} = (-0.05037565142175525, 1.3291638757790596)$,
$d_{28} = (4.942459887889321, -2.709981483171564)$,
$d_{29} = (4.942545539190744, 2.7086540705572233)$,
$d_{30} = (-0.6505063364528741, 0.12905013280033062)$,
$d_{31} = (-0.6504802925485075, -0.12888345506775425)$,
$d_{32} = (0.011414313921891395, 0.006515262160355019)$,
$d_{33} = (0.011411943056417042, -0.006518333287066878)$

and with a near optimally small number of poles for the prescribed L^∞ error δ_0. Since the poles of $S(\mathrm{i}x)$ and $R(\mathrm{i}x)$ are distinct, the function $S(\mathrm{i}x)R(\mathrm{i}x)$ can be expressed as a proper rational function. The final function $S(\mathrm{i}x)$ is what is shown in Fig. 5.

4. Examples

4.1 *The 2D (rotating) shallow water equations*

We apply the technique proposed to the linear shallow water equations:

$$\mathbf{v}_t = -fJ\mathbf{v} + \nabla\eta,$$
$$\eta_t = \nabla \cdot \mathbf{v},$$

where all quantities are as in Section 2.1, cf. equation (2.1).

We apply the algorithm in the spatial domain $[0,1] \times [0,1]$, using periodic boundary conditions and a constant Coriolis force $f = 1$. In this case, an exact solution can be computed analytically since the matrix exponential is diagonalized in the Fourier domain, and can be rapidly applied via the fast Fourier transform. In particular,

$$\mathcal{L}(\mathbf{r}_{\mathbf{k}}^{l} e^{\mathrm{i}\mathbf{k}\cdot\mathbf{x}}) = \mathrm{i}\omega_{\mathbf{k}}^{l} \mathbf{r}_{\mathbf{k}}^{l} e^{\mathrm{i}\mathbf{k}\cdot\mathbf{x}},$$

where $\mathbf{r}_{\mathbf{k}}^{l}$ are eigenvectors of the matrix

$$\begin{pmatrix} 0 & -f & \mathrm{i}k_1 \\ -f & 0 & \mathrm{i}k_2 \\ \mathrm{i}k_1 & \mathrm{i}k_2 & 0 \end{pmatrix}.$$

Explicit expressions for the eigenvectors $\mathbf{r}_{\mathbf{k}}^{l}$ can be found in Majda (2003).

We first compare the accuracy and efficiency of applying $e^{n\tau\mathcal{L}}\mathbf{u}_0$, for $\tau = 3$ and $n = 1, \ldots, 10$, against RK4 and against using Chebyshev polynomials. In particular, the Chebyshev method uses the approximation

$$e^{\Delta t\mathcal{L}}\mathbf{u}_0 \approx J_0(\mathrm{i})\mathbf{u}_0 + 2\sum_{k=0}^{K} (\mathrm{i})^k J_k(-\mathrm{i}) T_k(\Delta t\mathcal{L})\mathbf{u}_0, \tag{4.1}$$

coupled with the standard recursion for applying $T_k(\Delta t\mathcal{L})$.

For stability, Δt must be chosen so that the spectrum of the spatially discretized version of the operator $\Delta t\mathcal{L}$ is $\mathcal{O}(1)$ in magnitude (otherwise the terms in the sum (4.1) get very large and this results in catastrophic cancellation).

We choose a polynomial degrees of 12 and 9 for the high-accuracy and low-accuracy simulations, respectively, which we find experimentally is a good compromise between the time step size Δt needed for a given accuracy, and the number of applications of $\mathcal{L}$. In all the time-stepping schemes, we use the same spectral element discretization and parameter values as described above. All the algorithms are implemented in Octave, including the direct solver described in Section 2. The rational filter parameter k_0 is chosen according to the steps discussed in Section 3.3. For the all the following numerical examples, when applying the operator exponential using the rational approximation (1.5), we take $M = 160$ in (3.3), which results in 376 overall terms in (1.5) ($(2 \times (160 + 11) + 1 + 33)$ terms, with 33 coming from the rational filter function); the parameter h in (3.3) is taken to be $\frac{1}{3}$ for the lower accuracy simulations and $\frac{1}{5}$ for the higher accuracy simulations.

4.1.1 *First test case for the shallow water equations* We first consider the initial conditions:

$$\begin{aligned} \eta(\mathbf{x}) &= \sin(6\pi x)\cos(4\pi y) - \tfrac{1}{5}\cos(4\pi x)\sin(2\pi y), \\ v_1(\mathbf{x}) &= \cos(6\pi x)\cos(4\pi y) - 4\sin(6\pi x)\sin(4\pi y), \\ v_2(\mathbf{x}) &= \cos(6\pi x)\cos(6\pi y). \end{aligned} \tag{4.2}$$

For these initial conditions, we use $6 \times 6 = 36$ elements of equal area, and $16 \times 16 = 256$ Chebyshev quadrature nodes for each element. To assess the accuracy of the method, the exponential $e^{n\tau\mathcal{L}}\mathbf{u}_0$ is applied in the Fourier domain.

When applying the operator exponential using the rational approximation (1.5), we take $M = 160$ and $h = \frac{1}{5}$ in (3.3), which results in 376 overall terms in (1.5) ($(2 \times (160 + 11) + 1 + 33)$ terms, with 33

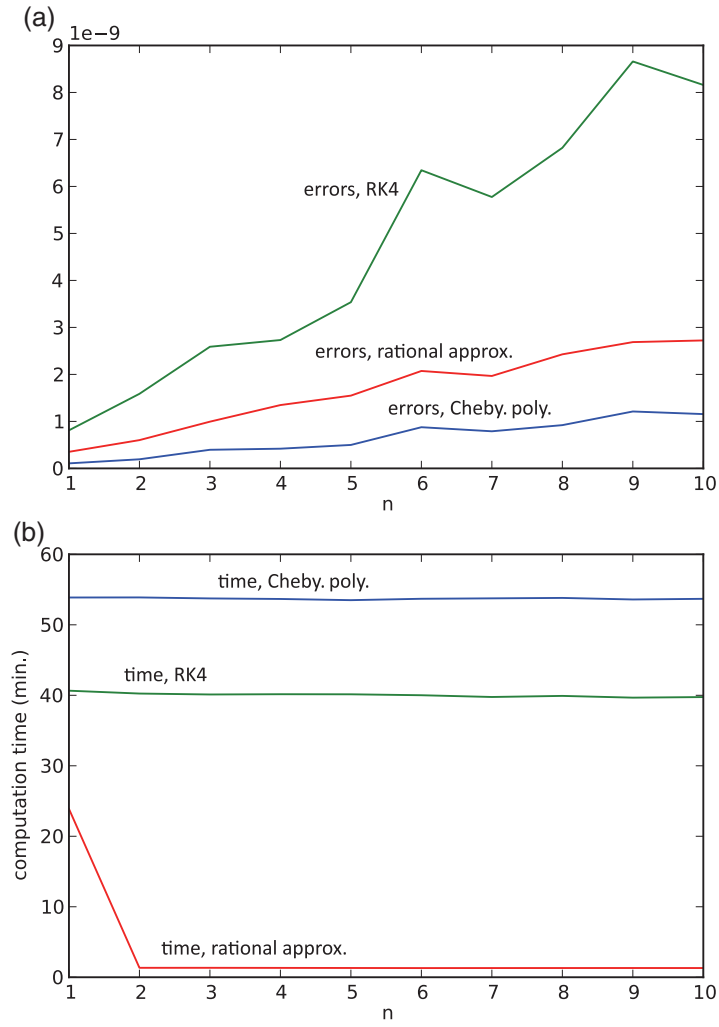

FIG. 6. (a) Plots of the L^∞ error, $\|\mathbf{u}_n - e^{n\tau\mathcal{L}}\mathbf{u}_0\|_\infty$, versus the big time step $n\tau$, where $\tau = 3$ and $1 \leqslant n \leqslant 10$. Here the approximation $\mathbf{u}_n$ is computed via RK4, the Chebyshev polynomial method, and the rational approximation method. (b) Plots of the computation time (min) versus the big time step $n\tau$, for the RK4, the Chebyshev polynomial method and the rational approximation method.

coming from the rational filter function).We also choose a big time step $\tau = 3$; with this overall choice of parameters, this results in an L^∞ error of 3.4×10^{-10} for a single (large) time step.

For this choice of parameters in the spectral element discretization, the cost of applying the solution operator of (2.3)—i.e., forming the right hand side of (2.6), solving (2.6), and evaluating (2.5)—is about 4.5 times more expensive than the cost of applying the forward operator (2.2) directly.

For the three time-stepping methods, the L^∞ errors in the approximation of $e^{n\tau\mathcal{L}}\mathbf{u}_0$, $n = 1, \ldots, 10$, are plotted in Fig. 6(a). Similarly, the total computation times (in minutes) of approximating $e^{n\tau\mathcal{L}}\mathbf{u}_0$, $n = 1, \ldots, 10$, are plotted in Fig. 6(b) (this includes the pre-computation time for representing the inverses). From Fig. 6(a), we see that the L^∞ errors from all three methods remain $< 10^{-8}$ for

TABLE 4 *Comparison of the accuracy and efficiency of applying,* $e^{\tau\mathcal{L}}\boldsymbol{u}_0$ *and* $\tau = 1.5$, *for system* (2.1) *and* $\boldsymbol{u}_0$ *in* (4.2). *The comparison uses RK*4, *Chebyshev polynomials and the rational approximation* (1.5); *in the spatial discretization of all three comparisons,* $12 \times 12 = 144$ *elements and* $16 \times 16 = 254$ *Chebyshev quadrature nodes per element are used*

$e^{\tau\mathcal{L}}$, $\tau = 1.5$	L^∞ error	Time (min)	Pre-comp. (min)
Rational approx., $M = 376$ terms	2.1×10^{-10}	4.39	103.1
RK4	7.0×10^{-10}	131.9	NA
Cheby. poly., degree 12	1.1×10^{-10}	150.5	NA

$n = 1, \ldots, 10$. From Fig. 6(b), we see that the first time step for the rational approximation method is about half the cost of both RK4 and the Chebyshev polynomial method. However, subsequent time steps for the new method is about 40 times cheaper than both RK4 and the Chebyshev polynomial method (for about the same accuracy).

4.1.2 *Second test case: doubling the spatial resolution* Next, we compute $e^{\tau\mathcal{L}}\mathbf{u}_0$, $\tau = 1.5$, with the initial conditions:

$$\begin{aligned} \eta(\mathbf{x}) &= \sin(12\pi x)\cos(8\pi y) - \tfrac{1}{5}\cos(8\pi x)\sin(4\pi y), \\ v_1(\mathbf{x}) &= \cos(12\pi x)\cos(8\pi y) - 4\sin(12\pi x)\sin(8\pi y), \\ v_2(\mathbf{x}) &= \cos(12\pi x)\cos(12\pi y). \end{aligned} \tag{4.3}$$

In particular, we double the bandlimit in each direction. In each of the time-stepping schemes, we use $12 \times 12 = 144$ elements of equal area, and $16 \times 16 = 256$ Chebyshev quadrature nodes for each element. We again use the same parameters for the rational approximation (1.5) as described in Section 4.1.1.

We only examine the error and computation time for one big time step. For the rational approximation method, we present both the pre-computation time for obtaining data-sparse representations of the 376 inverses in (1.5), and the computation time for applying the approximation in (1.5) (once the data-sparse representations are known). The results are summarized in Table 4. Since we only consider a single time step, the pre-computation time and application time are included separately. The main conclusion to draw from these results is that doubling the spatial resolution does not appreciably change the relative efficiency of the three time-stepping methods (once representations for the inverse operators in (1.5) are pre-computed).

4.1.3 *Third test case: applying the operator exponential over a long time interval* We now access the accuracy of the new method when repeatedly applying $e^{\tau\mathcal{L}}$, $\tau = 1$, in order to evolve the solution over longer time intervals. In this example, we use the initial conditions:

$$\begin{aligned} \eta(\mathbf{x}) &= \exp(-100((x-1/2)^2 + (y-1/2)^2)), \\ v_1(\mathbf{x}) &= \cos(6\pi x)\cos(4\pi y) - 4\sin(6\pi x)\sin(4\pi y), \\ v_2(\mathbf{x}) &= \cos(6\pi x)\cos(6\pi y). \end{aligned} \tag{4.4}$$

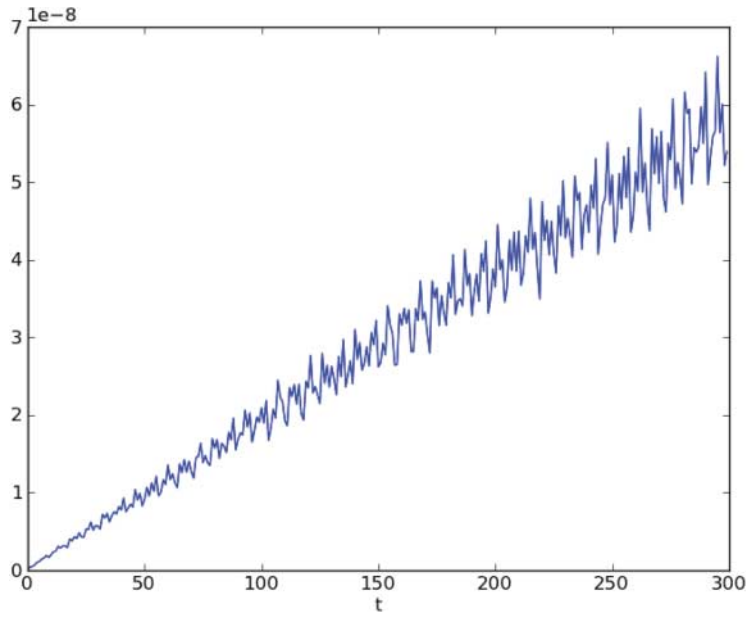

FIG. 7. Plot of the L^∞ error, $\|\mathbf{u}_n - e^{n\tau\mathcal{L}}\mathbf{u}_0\|_\infty$, versus the big time step $n\tau$, where $\tau = 1$ and $1 \leqslant n \leqslant 300$. Here $\mathbf{u}_n$ denotes the numerical approximation to $e^{n\tau\mathcal{L}}\mathbf{u}_0$, as computed by the rational approximation (1.5) and the direct solver from Section 2.

Note that these initial conditions cannot be expressed as a finite sum of eigenfunctions of $\mathcal{L}$. We use the same spatial discretization parameters as in Section 4.1.2.

In Fig. 7, we show the L^∞ error of the computed approximation $\mathbf{u}_n(\mathbf{x})$ to $\mathbf{u}(\mathbf{x}, n\tau)$, $n = 1, \ldots, 300$. As expected, the error increases linearly in the number of applications of the exponential. Note that, due to the large step size of $\tau = 1$, the error accumulates slowly in time and the solution can be propagated with high accuracy over a large number of characteristic wavelengths.

4.1.4 *Fourth test case: applying the operator exponential with lower accuracy* We now repeat the first example 4.1.1, but this time using lower accuracy for the temporal discretization. In particular, we again use use $M = 376$ inverses for the rational approximation. However, we take a both smaller accuracy by choosing $h = \frac{1}{3}$ in the rational approximation (3.3), as well as a larger single (large) time step $\tau = 5$. This choice of parameters results in an L^∞ error of 4.04×10^{-6} for a single (large) time step.

The results are summarized in Table 5. From this table, we see that the pre-computation time needed to represent the $M = 376$ solution operators in (1.5) is 17.8 min, and the computation time needed to apply the exponential is 0.925 min; the final accuracy in the L^∞ norm is given by 4.0×10^{-6}. For the Chebyshev polynomial method, we used degree 9 polynomials and a time step of 0.004; the overall time for the application of the exponential is 23.3 min, and the final accuracy is given by 3.7×10^{-6}. Finally, for RK4 we used a time step of 0.002; using RK4 takes an overall time of 5.44 min, with a final accuracy of 1.37×10^{-5}.

4.1.5 *Fifth test case: applying the operator exponential to a nonsmooth initial condition* We now repeat the third example 4.1.3 for long time simulations, but this time we use an initial condition with a

TABLE 5 *Comparison of the accuracy and efficiency of applying,* $e^{\tau\mathcal{L}}\boldsymbol{u}_0$ *and* $\tau = 5$, *for system* (2.1) *and* $\boldsymbol{u}_0$ *in* (4.2). *The comparison uses RK*4, *Chebyshev polynomials and the rational approximation* (1.5); *in the spatial discretization of all three comparisons,* $6 \times 6 = 36$ *elements and* $16 \times 16 = 254$ *Chebyshev quadrature nodes per element are used*

$e^{\tau\mathcal{L}}$, $\tau = 5$	L^∞ error	Time (min)	Pre-comp. (min)
Rational approx., $M = 376$ terms	4.0×10^{-6}	0.925	17.8
RK4	1.3×10^{-5}	5.44	NA
Cheby. poly., degree 9	3.7×10^{-6}	23.3	NA

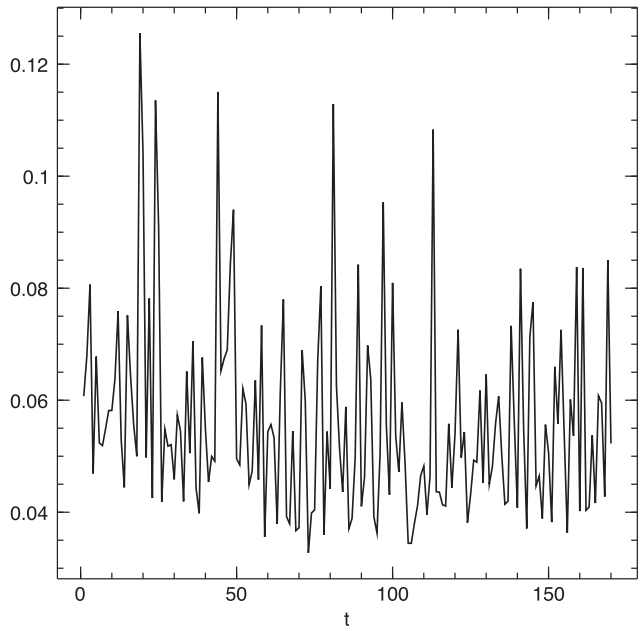

FIG. 8. Plot of the L^∞ error, $\|\mathbf{u}_n - e^{n\tau\mathcal{L}}\mathbf{u}_0\|_\infty$, versus the big time step $n\tau$, where $\tau = 3$ and $1 \leqslant n \leqslant 170$. Here the initial condition in Section 4.1.5 is used, which contains a cusp-type singularity, and $\mathbf{u}_n$ denotes the numerical approximation to $e^{n\tau\mathcal{L}}\mathbf{u}_0$ obtained from the rational approximation method.

cusp-type singularity. In particular, we use $\tau = 3$, $M = 376$ inverses for the rational approximation, the same spectral element grid from Section 4.1.1, and initial conditions:

$$\begin{aligned} \eta(\mathbf{x}) &= \exp(-100\sqrt{((x-1/2)^2 + (y-1/2)^2)}), \\ v_1(\mathbf{x}) &= 0, \\ v_2(\mathbf{x}) &= 0. \end{aligned} \tag{4.5}$$

Note that the interface variable η now has a cusp-type singularity at $(\frac{1}{2}, \frac{1}{2})$.

TABLE 6 *Comparison of three methods for a high accuracy computation of the operator exponential* $e^{t\mathcal{L}}\boldsymbol{u}_0$ *and* $t = 1.5$, *for system* (2.8) *and* $\boldsymbol{u}_0$ *in* (4.7). *The comparison uses RK*4, *Chebyshev polynomials and the rational approximation* (1.5); *in the spatial discretization of all three comparisons,* $12 \times 12 = 144$ *elements and* $16 \times 16 = 254$ *Chebyshev quadrature nodes per element are used*

$e^{t\mathcal{L}}$, $t = 1.5$	L^∞ error	Time (min)	Pre-comp. (min)
Rational approx., $M = 376$ terms	1.6×10^{-9}	3.76	113.4
RK4	3.5×10^{-10}	63.9	NA
Cheby. poly., degree 12	3.5×10^{-8}	57.5	NA

In Fig. 8, we show the L^∞ error of the computed approximation $\mathbf{u}_n(\mathbf{x})$ to $\mathbf{u}(\mathbf{x}, n\tau)$, $n = 1, \ldots, 300$. As expected, the singularity in η results in essentially no accuracy (due to a lack of spatial resolution in the spectral element grid). However, note that the time-stepping scheme itself remains stable throughout the evolution.

4.2 *Example 2*

4.2.1 *First test case: applying the operator exponential with high accuracy*

In our second example, we consider the wave propagation problem

$$u_{tt} = \kappa \Delta u, \quad \mathbf{x} \in [0, 1] \times [0, 1], \tag{4.6}$$

where $\kappa(\mathbf{x}) > 0$ is a smooth function, the initial conditions $u(\mathbf{x}, 0)$ and $u_t(\mathbf{x}, 0)$ are prescribed, and periodic boundary conditions are used.

Since the procedure and results are similar to those in Section 4.1, we simply test the efficiency and accuracy of this method over a single time step $\tau = 1.5$. In particular, we compare the accuracy and efficiency for one application $e^{\tau\mathcal{L}}\mathbf{u}_0$, $\tau = 1.5$, against RK4 and against using Chebyshev polynomials. In our numerical experiments, we use the initial condition

$$u(x, y, 0) = \sin(2\pi x)\sin(2\pi y) + \sin(4\pi x)\sin(4\pi y), \tag{4.7}$$

and $u_t(x, y, 0) = 0$. We also use

$$\kappa(x, y) = \left(\frac{3 + \sin(4\pi x)}{4}\right)^{1/2} \left(\frac{3 + \sin(4\pi y)}{4}\right)^{1/2}.$$

Finally, in the spatial discretization, we use $12 \times 12 = 144$ elements with $16 \times 16 = 256$ points per element (for all three time-stepping methods), and $M = 376$ poles in (1.5). For these parameters, the time to apply the inverse of (2.9)—which involves forming the right hand side in (2.11), solving for v, and computing (2.10)—is about 5.2 times more expensive than directly applying the forward operator (2.8).

Unlike Section 4.1, the operator exponential is not diagonalized in the Fourier domain. To assess the accuracy, we use the Chebyshev polynomial method with a small enough step size to yield an estimated error of $< 10^{-10}$. In particular, we verify that the L^∞ residual, $\|\mathbf{u}(\mathbf{x}, t; \Delta t) - \mathbf{u}(\mathbf{x}, t; \Delta t/2)\|_\infty$, using numerical approximations to $\mathbf{u}(\mathbf{x}, t)$ computed with step sizes Δt and $\Delta t/2$ and the Chebyshev polynomial method, is $< 10^{-10}$. We then use $\mathbf{u}(\mathbf{x}, t; \Delta t/2)$ as a reference solution.

TABLE 7 *Comparison of three methods for a lower accuracy computation of the operator exponential* $e^{t\mathcal{L}}\boldsymbol{u}_0$ *and* $t = 2.5$, *for system* (2.8) *and* $\boldsymbol{u}_0$ *in* (4.7). *The comparison uses RK*4, *Chebyshev polynomials and the rational approximation* (1.5); *in the spatial discretization of all three comparisons,* $12 \times 12 = 144$ *elements and* $16 \times 16 = 254$ *Chebyshev quadrature nodes per element are used*

$e^{t\mathcal{L}}$, $t = 2.5$	L^∞ error	Time (min.)	Pre-comp. (min.)
Rational approx., $M = 376$ terms	1.1×10^{-6}	4.0	121
RK4	1.4×10^{-6}	16.2	NA
Cheby. poly., degree 9	8.5×10^{-6}	73.8	NA

The results are summarized in Table 6. From this table, we see that the pre-computation time needed to represent the $M = 376$ solution operators in (1.5) is 113.4 min, and the computation time needed to apply the exponential is 3.7 min; the final accuracy in the L^∞ norm is given by 1.6×10^{-9}. For the Chebyshev polynomial method, 575 time steps of size $\Delta t \approx .0026$ are taken, for an overall time of 57 min; the final accuracy is given by 3.5×10^{-8}. Finally, for RK4, 7,500 time steps of size $\Delta t = \frac{1}{5} \times 10^{-3}$ are taken, for an overall time of 63.9 minutes; the final accuracy is 3.5×10^{-10}.

4.2.2 *Second test case: applying the operator exponential with lower accuracy* We now apply the operator exponential with lower accuracy, using the same initial conditions and spatial parameter values as Section 4.2.1. In particular, we again use $M = 376$ inverses for the rational approximation, but instead choose a larger temporal discretization parameter $h = \frac{1}{3}$ in the rational approximation (3.3), as well as a larger single (large) time step $\tau = \frac{5}{2}$, which results in both $\approx 10^{-6}$ accuracy. We verify stability of the scheme for many time steps (though we only show efficiency and accuracy comparison for a single time step).

We assess the accuracy of all three methods by against the solution obtained via use of Chebyshev polynomial method with degree 12 polynomials and $\Delta t = .0025/2$. The results are summarized in Table 7. For the rational approximation method, the total precomputation time for computing a representation of the operator exponential is 121 min, and the time for a single application of the operator exponential is 4.0 min; the final accuracy is given by 1.1×10^{-6}. Similarly, for the Chebyshev polynomial method, the total time for applying the operator exponential using a time step of 0.0025 and degree 9 polynomials is 73.8 min, with a final accuracy of 8.5×10^{-6}. Finally, the time for applying the operator exponential using RK4 and a time step of 0.00125 is 16.2 min, with a final accuracy of 1.41×10^{-6}.

5. Generalizations

The manuscript presents an efficient technique for explicitly computing a highly accurate approximation to the operator $\varphi(\tau\mathcal{L})$ for the case where $\mathcal{L}$ is a skew-Hermitian operator and where $\varphi(t) = e^t$, so that $\varphi(\tau\mathcal{L})$ is the time-evolution operator of the hyperbolic PDE $\partial u/\partial t = \mathcal{L}u$. The technique can be extended to more general functions φ. In particular, in using exponential integrators (cf. Cox & Matthews, 2002), it is desirable to apply functions $\varphi_j(\tau\mathcal{L})$, where $\varphi_j(\cdot)$ are the so-called phi-functions. In Section 3, we presented (near) optimal rational approximations of the first few phi-functions. An important property of these representations is that the same poles can be used to simultaneously apply

all the phi-functions, and with a uniformly small error. In particular, linear combinations of the same $2M + 1$ solutions $(\tau\mathcal{L} - \alpha_m)^{-1}\mathbf{u}_0$, $m = -M, \ldots, M$, can be used to apply $\varphi_j(\tau\mathcal{L})$ for $j = 1, 2, \ldots$. In a similar way, linear combinations of the same $2M + 1$ solutions can be used to apply $e^{s\mathcal{L}}$ for $0 \leqslant s \leqslant \tau$.

In addition, where there is a priori knowledge of large spectral gaps—for example, when there is scale separation between fast and slow waves—the techniques in this paper, coupled with those in Haut & Beylkin (2012), can be used to construct efficient rational approximations of e^{ix} which are (approximately) nonzero only where the spectrum of $\mathcal{L}$ is nonzero. Since suitably constructed rational approximations can capture functions with sharp transitions using a small number of poles (see Haut & Beylkin, 2012), this approach requires a potentially much smaller number of inverse applications.

References

Bergamaschi, L. & Vianello, M. (2000) Efficient computation of the exponential operator for large, sparse, symmetric matrices. *Numer. Linear Algebra Appl.*, **7**, 27–45.

Beylkin, G. & Sandberg, K. (2005) Wave propagation using bases for bandlimited functions. *Wave Motion*, **41**, 263–291.

Cody, W. J., Meinardus, G. & Varga, R. S. (1969) Chebyshev rational approximations to e^{-x} in $[0, +\infty)$ and applications to heat-conduction problems. *J. Approx. Theory*, **2**, 50–65.

Cox, S. M. & Matthews, P. C. (2002) Exponential time differencing for stiff systems. *J. Comput. Phys.*, **176**, 430–455.

Damle, A., Beylkin, G., Haut, T. S. & Monzon, L. (2013) Near optimal rational approximations of large data sets. *Appl. Comput. Harmon. Anal.*, **35**, 251–263.

Demanet, L. & Ying, L. (2009) Wave atoms and time upscaling of wave equations. *Numer. Math.*, **113**, 1–71.

Duff, I. S., Erisman, A. M. & Reid, J. K. (1986) *Direct Methods for Sparse Matrices*. Oxford: Clarendon Press.

Gavrilyuk, I. P., Hackbusch, W. & Khoromskij, B. N. (2005) Hierarchical tensor-product approximation to the inverse and related operators for high-dimensional elliptic problems. *Computing*, **74**, 131–157.

George, A. (1973) Nested dissection of a regular finite element mesh. *SIAM J. Numer. Anal.*, **10**, 345–363.

Gillman, A. & Martinsson, P. G. (2014) A direct solver with $O(n)$ complexity for variable coefficient elliptic PDEs discretized via a high-order composite spectral collocation method. *SIAM J. Sci. Comput.*, **36**, A2023–A2046.

Güttel, S. (2013) Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection. *GAMM Mitt.*, **36**, 8–31.

Haut, T. S. & Beylkin, G. (2012) Fast and accurate con-eigenvalue algorithm for optimal rational approximations. *SIAM J. Matrix Anal. Appl.*, **33**, 1101–1125.

Haut, T. S. & Wingate, B. (2014) An asymptotic parallel-in-time method for highly oscillatory PDEs. *SIAM J. Sci. Comput.*, **36**, A693–A713.

Higham, N. (2005) The scaling and squaring method for the matrix exponential revisited. *SIAM J. Matrix Anal. Appl.*, **26**, 1179–1193.

Hochbruck, M. & Lubich, C. (1997) On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, **34**, 1911–1925.

Hochbruck, M. & Ostermann, A. (2010) Exponential integrators. *Acta Numer.*, **19**, 209–286.

Majda, A. J. (2003) *Introduction to PDEs and Waves for the Atmosphere and Ocean*. Courant lecture notes in mathematics. Providence (RI), New York: Courant Institute of Mathematical Sciences.

Martinsson, P. G. (2013) A direct solver for variable coefficient elliptic PDEs discretized via a composite spectral collocation method. *J. Comput. Phys.*, **242**, 460–479.

Martinsson, P. G. (2015) The Hierarchical Poincare-Steklov (HPS) solver for elliptic PDEs: a tutorial, arxiv.org report #1506.01308.

Maz'ya, V. & Schmidt, G. (1996) On approximate approximations using Gaussian kernels. *IMA J. Numer. Anal.*, **16**, 13–29.

Maz'ya, V. & Schmidt, G. (2007) *Approximate Approximations*. Mathematical Surveys and Monographs, vol. 141. Providence, RI: American Mathematical Society.
Müller, F. & Varnhorn, W. (2007) Error estimates for approximate approximations with Gaussian kernels on compact intervals. *J. Approx. Theor.*, **145**, 171–181.
Paldor, N. & Sigalov, A. (2011) An invariant theory of the linearized shallow water equations with rotation and its application to a sphere and a plane. *Dyn. Atmospheres Oceans*, **51**, 26–44.
Schmelzer, T. & Trefethen, L. N. (2007a) Evaluating matrix functions for exponential integrators via Carathéodory–Fejér approximation and contour integrals. *ETNA, Electron. Trans. Numer. Anal.*, **29**, 1–18.
Schmelzer, T. & Trefethen, L. N. (2007b) Evaluating matrix functions for exponential integrators via Carathéodory-Fejér approximation and contour integrals. *Electron. Trans. Numer. Anal.*, **29**, 1–18.

Appendix Error bounds

Define the rational approximation $R_M(\mathrm{i}x)$ to $e^{\mathrm{i}x}$,

$$R_M(\mathrm{i}x) = \sum_{m=-M-11}^{M+11} \frac{c_m}{\mathrm{i}x - h(\mu + \mathrm{i}m)}, \tag{A.1}$$

where c_m is defined as

$$c_m = \frac{h}{\psi_h(1)} \sum_{k=-\max(-11,m-M)}^{\max(-11,m-M)} a_k e^{-(\mathrm{i}/h)(m-k)},$$

and the values a_k and μ are given in Table 1. Then we have the following theorem.

Theorem A.1 The rational approximation in (A.1) satisfies

$$|e^{\mathrm{i}x} - R_M(\mathrm{i}x)| \leqslant \delta_1 + 2(M+11)\delta_2,$$

where

$$\delta_1 = \frac{1}{\psi_h(1)} \left(\sum_{k \neq 0} \hat{\psi}_h\left(\frac{k}{h}\right) + \sum_{|m|>M} \psi_h(x+mh) \right),$$

$\psi_h(x) = (4\pi)^{-1/2} e^{-x^2/(4h^2)}$ and $\delta_2 \approx 5 \times 10^{-13}$.

Proof. From equation (3.3), we see that it suffices to bound δ_1. To do so, we use an application of the Poisson summation formula:

$$\sum_{m=-\infty}^{\infty} \Psi_h(x+mh) = \frac{1}{h} \sum_{k=-\infty}^{\infty} e^{2\pi\mathrm{i}(k/h)x} \hat{\Psi}_h\left(\frac{k}{h}\right).$$

Indeed, applying this to $\Psi_h(x) = e^{-2\pi \mathrm{i}x}\psi_h(x)$, we have that

$$\begin{aligned}\sum_{m=-\infty}^{\infty} \Psi_h(x+mh) &= e^{-2\pi \mathrm{i}x} \sum_{m=-\infty}^{\infty} e^{-2\pi \mathrm{i}mh}\psi_h(x+mh) \\ &= \frac{1}{h}\sum_{k=-\infty}^{\infty} e^{2\pi \mathrm{i}(k/h)x}\hat{\Psi}_h\left(\frac{k}{h}\right) \\ &= \frac{1}{h}\sum_{k=-\infty}^{\infty} e^{2\pi \mathrm{i}(k/h)x}\hat{\psi}_h\left(\frac{k}{h}+1\right),\end{aligned}$$

where the last equality uses the fact that

$$\hat{\Psi}_h\left(\frac{k}{h}\right) = \hat{\psi}_h\left(\frac{k}{h}+1\right).$$

Therefore,

$$\left|\sum_{m=-\infty}^{\infty} e^{-2\pi \mathrm{i}mh}\psi_h(x+mh) - \frac{\hat{\psi}_h(1)}{h}e^{2\pi \mathrm{i}x}\right| \leqslant \frac{1}{h}\sum_{k\neq 0}\hat{\psi}_h\left(\frac{k}{h}\right).$$

Finally, truncating the sum we obtain the bound (3.6). □

Theorem A.1 shows that the first term in the right-hand side (involving $\hat{\psi}_h(k/h)$) is negligible for $h \lesssim 1$, owing to the tight frequency localization of ψ_h. Indeed, the error decays at a super exponential rate once $1/h$ is larger than the bandlimit of $e^{\mathrm{i}x}$. In addition, the second term in the right-hand side above (involving $\psi_h(x+mh)$) is negligible for $|x| \leqslant (M - m_0)h$, where $m_0 = \mathcal{O}(1)$. Therefore, the interval over which the approximation is valid is $|x| \lesssim Mh$.