

High Order Accurate Hermite Schemes on Curvilinear Grids with Compatibility Boundary Conditions

Allen Alvarez Loya^{a,1,*}, Daniel Appelö^{b,2}, William D. Henshaw^{c,3}

^a*Los Alamos National Laboratory, Los Alamos, NM 87544, USA*

^b*Department of Mathematics, Virginia Tech, Blacksburg, VA 24061 USA*

^c*Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, USA*

Abstract

High order accurate Hermite methods for the wave equation on curvilinear domains are presented. Boundaries are treated using centered compatibility conditions rather than more standard one-sided approximations. Both first-order-in-time (FOT) and second-order-in-time (SOT) Hermite schemes are developed. Hermite methods use the solution and multiple derivatives as unknowns and achieve space-time orders of accuracy $2m - 1$ (FOT) and $2m$ (SOT) for methods using $(m + 1)^d$ degree of freedom per node in d dimensions. The compatibility boundary conditions (CBCs) are based on taking time derivatives of the boundary conditions and using the governing equations to replace the time derivatives with spatial derivatives. These resulting constraint equations augment the Hermite scheme on the boundary. The solvability of the equations resulting from the compatibility conditions are analyzed. Numerical examples demonstrate the accuracy and stability of the new schemes in two dimensions.

Keywords: Wave equation; Hermite methods; compatibility boundary conditions; high-order accuracy

*Corresponding author

Email addresses: aalvarezloya@lanl.gov (Allen Alvarez Loya), appel@vt.edu (Daniel Appelö), henshw@rpi.edu (William D. Henshaw)

¹Research supported by National Science Foundation under grants DGE-1650115 and DMS-2213261

²Research supported by National Science Foundation under grant DMS-2345225, and Virginia Tech.

³Research supported by the National Science Foundation under grants DMS-1519934 and DMS-1818926.

Contents

1	Introduction	3
2	Preliminaries: governing equations, grids, and discrete approximations	4
2.1	Governing equations	4
2.2	Mappings and parameter space equations	5
2.3	Grids, Taylor polynomial representations and Hermite interpolants	5
3	Hermite algorithms	6
4	Compatibility boundary conditions	7
4.1	CBCs on a Cartesian grid	8
4.2	CBCs for corners on a Cartesian grid	10
4.2.1	Dirichlet-Dirichlet CBC corner	10
4.2.2	Neumann-Neumann CBC corner	12
4.2.3	Dirichlet-Neumann CBC corner	12
4.3	CBCs on a curvilinear grid	13
4.3.1	Dirichlet CBCs on a curvilinear grid	13
4.3.2	Neumann CBCs on a curvilinear grid	14
4.3.3	Corner CBCs on a curvilinear grid	15
5	Analysis of the CBCs	15
5.1	Solvability of the CBC matrix systems on Cartesian grids	15
5.2	Solvability of CBC matrix systems on curvilinear grids	16
5.3	Symmetry properties of the CBC conditions	18
6	Numerical results	19
6.1	Mappings and Grids	20
6.2	Manufactured and exact solutions	21
6.3	Results for the first-order in time (FOT) scheme	22
6.4	Results for the second-order in time (SOT) scheme	23
6.5	Long time simulations	24
7	Conclusions	26
Appendix A	Solvability and conditioning of the CBC matrices on Cartesian grids	27
Appendix B	Hermite evolution operators for the FOT and SOT schemes	28
Appendix B.1	FOT evolution	29
Appendix B.2	SOT evolution	29
Appendix B.2.1	First time-step	31
Appendix B.3	Choosing the time-step	31
Appendix C	Practicalities	32
Appendix C.1	Hermite interpolants	32
Appendix C.2	Taylor polynomial coefficients from function evaluations	33
Appendix C.3	Taylor polynomials for the Laplacian in curvilinear coordinates	33

Appendix C.4 Applying the wave operator in curvilinear coordinates 35

1. Introduction

We develop high-order accurate Hermite methods for the wave equation on curvilinear grids. Both first-order in time (FOT) and **second-order in time (SOT) schemes are developed**. Compatibility boundary conditions (CBCs) are used to give high-order accurate centered approximations to boundary conditions rather than more common one-sided approximations; these centered boundary conditions are generally more stable and accurate than using one sided approximations [1]. A key result of the current article is to show how CBCs can be incorporated into Hermite schemes. **This is the first Hermite method that can handle complex geometry *and* boundary conditions purely within the Hermite method framework.** The Hermite method approximates the solution to a partial differential equation using degrees of freedom at each node representing the solution and derivatives up to degree m resulting in $(m+1)^d$ degrees of freedom per node in d dimensions. The nodal values are interpolated to the cell centers and advanced in time using a Taylor series in time over half a time step. The cell centered values at the half time-step are then interpolated to the nodes and advanced the second-half time step. The evolution of the degrees of freedom is local, minimizing the communication and storage costs. The resulting schemes have order of accuracy $2m - 1$ for the FOT scheme, and $2m$ for the **SOT** scheme. We note that when the FOT scheme is used on Cartesian grids and with $d(2m+1)$ terms in the Taylor series in time (see [2, 3]) the schemes have a CFL number that is one, i.e. they are stable for $c\Delta t/h \leq 1$ where c is the wave speed and h in the grid spacing. This results holds at any order of accuracy and is significantly better than many other high-order schemes. To achieve space-time accuracies of $2m - 1$ for the FOT scheme it is sufficient to use a Taylor series in time with $2m + 1$ terms, which is what we do here. This reduces the time-step slightly, here we use $c\Delta t/h = 0.5$, but does not significantly change the efficiency of the methods as the cost per time-step is reduced by a factor d . The **SOT** scheme is even more efficient, requiring only $m + 1$ terms in time for any dimension on Cartesian meshes. On curvilinear meshes we find that we have to reduce the timestep slightly, here we use $c\Delta t/h = 0.4$.

High-order accurate Hermite methods were first introduced for hyperbolic systems of equations by Goodrich et al. in [2]. Since then there have been many enhancements to the original methods described in [2]. These improvements include but are not limited to order-adaptive implementations [4], flux-conservative formulation for conservation laws [5], coupling with a discontinuity sensor to resolve kinks [6] and coupling with discontinuous Galerkin methods to handle complex boundaries [7, 8]. In [3] the authors developed *dissipative* (FOT) and *conservative* (**SOT**) Hermite interpolation based schemes for the wave equation. The schemes were developed for rectangular geometries where boundary conditions can be imposed by a simple mirroring strategy. Here these **schemes** are extended to curvilinear geometries. For complex geometries Hermite methods have been used as efficient building blocks in hybrid methods. In [8] the authors developed a hybrid Hermite-discontinuous Galerkin (DG) scheme for solving hyperbolic systems and in [7] a Hermite-DG scheme was developed for the wave equation. The DG approach was used on curvilinear grids near boundaries, but suffered from a smaller time-step restriction than the Hermite method. One goal of the current work is to develop Hermite schemes for curvilinear domains with boundaries so they can eventually **be** used on overset grids which consist of overlapping curvilinear grids near boundaries and one or more background Cartesian grids. In this way efficient high-order accurate Hermite methods with large CFL time-steps can be used for complex geometry. **As with all high order accurate methods, to achieve design rates of convergence, Hermite methods require that the solutions are smooth enough.** Here we only consider examples that have smooth solutions. Problems with non-smooth solutions discretized with high order accurate Hermite methods were studied in [9] where it was found that (just as for finite difference methods [10]) they have sublinear rates of convergence but with constants that decrease rapidly with order.

Compatibility boundary conditions have been used for finite-difference methods for many years (at least since the early 1980s). For example, in [11–13] the authors use compatibility conditions for second-order and fourth-order accurate approximations of the incompressible Navier-Stokes equations. For wave problems, compatibility conditions have been used in [14] for compressible Navier-Stokes and linear elasticity [15], as well as high-order schemes for Maxwell’s equations [14, 16]. Shu and collaborators have used CBCs in their inverse-Lax-Wendroff approach for hyperbolic equations and conservation laws [17–20] as well as for parabolic and advection-diffusion equations [21, 22]. CBCs are used in the book by Gustafsson on high-order difference methods [23]. CBCs have been used by LeVeque and Li with their immersed interface method to develop accurate approximations at embedded interfaces [24–26]. CBCs have also been used to derive stable and accurate embedded boundary approximations [27–29]. CBCs have been incorporated into summation-by-parts schemes by Sjögreen and Petersson for the equations of elasticity [30]. [Another example where compatibility conditions are used for wave propagation is in the difference potential method by Petropavlovsky et al. \[31\].](#)

In recent work [1], the authors develop *local compatibility boundary conditions* (LCBCs) for high-order accurate finite difference methods on Cartesian and curvilinear grids. The LCBC approach was actually first inspired by the CBC approach described in the present article for use with Hermite methods (even though publication of the LCBC method appears first).

A large number of methods for the wave equation have been proposed in the literature. An incomplete list is, finite difference methods based on the summation-by-parts framework [32] and up-winding [33, 34], finite element methods which use mass lumping to achieve efficiency [35], discontinuous Galerkin methods [36–39], as well as more exotic methods such as Fourier-Continuation [40, 41] and Galerkin differences [42]. We contend that Hermite methods have unique properties (in particular the order independent CFL number in combination with the $\sim 2m^{\text{th}}$ order using $\sim (m + 1)^d$ degrees of freedoms per element) that makes their development for use on more general geometries worthwhile.

The remainder of the paper is organized as follows. In Section 2 the governing equations are presented together with a discussion of curvilinear grids and the representation of discrete solutions for the Hermite schemes. A high-level summary of the Hermite schemes is given in Section 3. Section 4 describes the compatibility boundary condition approach for both Cartesian and curvilinear grids. An analysis of the solvability and conditioning of the systems of linear equations resulting from the CBC approximations is given in Section 5. Numerical results are presented in Section 6. The Appendices contain some proofs, a description of the evolution operators for the Hermite schemes, as well as some practical implementation details.

2. Preliminaries: governing equations, grids, and discrete approximations

2.1. Governing equations

We consider the initial boundary-value problem for the solution $u = u(\mathbf{x}, t)$ to the scalar wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \Delta u + f(\mathbf{x}, t), \quad \mathbf{x} \in \Omega, \quad t > 0, \quad (1a)$$

$$u(\mathbf{x}, 0) = U_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (1b)$$

$$\frac{\partial u}{\partial t}(\mathbf{x}) = U_1(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (1c)$$

$$\mathcal{B}u(\mathbf{x}, t) = g(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega. \quad (1d)$$

Here $c > 0$ is the wave speed, $\mathbf{x} = [x_1, x_2] = [x, y]$, and $f(\mathbf{x}, t)$ and $g(\mathbf{x}, t)$ are given forcing functions. The operator \mathcal{B} in (1d) denotes the boundary condition operator being of Dirichlet or Neumann type. The initial conditions for u and $\partial_t u$ are given in terms of known functions $U_0(\mathbf{x})$ and $U_1(\mathbf{x})$. In this article we focus on problems in $d = 2$ space dimensions but note that the method can be extended to three dimensions.

2.2. Mappings and parameter space equations

To discretize (1) on a structured curvilinear grid, we assume there exists a smooth and invertible mapping \mathbf{G} ,

$$\mathbf{x} = \mathbf{G}(\mathbf{r}), \quad (2)$$

from the unit square coordinates $\mathbf{r} = [r_1, r_2] = [r, s] \in [0, 1]^2$ to the physical domain coordinates $\mathbf{x} \in \Omega$. Using the chain rule the wave equation (1a) can be transformed to parameter space

$$\partial_t^2 U = c^2 \Delta_{\mathbf{r}} U + F(\mathbf{r}, t), \quad (3a)$$

$$c^2 \Delta_{\mathbf{r}} U \stackrel{\text{def}}{=} a^{20}(\mathbf{r}) \frac{\partial^2 U}{\partial r_1^2} + a^{11}(\mathbf{r}) \frac{\partial^2 U}{\partial r_1 \partial r_2} + a^{02}(\mathbf{r}) \frac{\partial^2 U}{\partial r_2^2} + a^{10}(\mathbf{r}) \frac{\partial U}{\partial r_1} + a^{01}(\mathbf{r}) \frac{\partial U}{\partial r_2}, \quad (3b)$$

where $U(\mathbf{r}, t) = u(\mathbf{G}(\mathbf{r}), t)$, $F(\mathbf{r}, t) = f(\mathbf{G}(\mathbf{r}), t)$, and the coefficients are

$$a^{20} = c^2 \|\nabla_{\mathbf{x}} r_1\|^2 = c^2 \left[(\partial_{x_1} r_1)^2 + (\partial_{x_2} r_1)^2 \right], \quad (3c)$$

$$a^{02} = c^2 \|\nabla_{\mathbf{x}} r_2\|^2 = c^2 \left[(\partial_{x_1} r_2)^2 + (\partial_{x_2} r_2)^2 \right], \quad (3d)$$

$$a^{11} = c^2 2(\nabla_{\mathbf{x}} r_1 \cdot \nabla_{\mathbf{x}} r_2) = 2(\partial_{x_1} r_1 \partial_{x_1} r_2 + \partial_{x_2} r_1 \partial_{x_2} r_2), \quad (3e)$$

$$a^{10} = c^2 (\partial_{x_1}^2 + \partial_{x_2}^2) r_1, \quad (3f)$$

$$a^{01} = c^2 (\partial_{x_1}^2 + \partial_{x_2}^2) r_2. \quad (3g)$$

In the remainder of the article we will write $u(\mathbf{r}, t)$ instead of $U(\mathbf{r}, t)$. We note that here we only consider a constant wave speed, but that a spatially variable wave speed can easily be incorporated since we are already discretizing a variable coefficient problem in the reference domain.

2.3. Grids, Taylor polynomial representations and Hermite interpolants

The unit square parameter space will be discretized with a primal (node centered) grid and a dual (cell centered) grid. Let $\mathbf{r}_{\mathbf{i}}$ denote points on either the primal or dual grid, where $\mathbf{i} = [i_1, i_2] = [i, j]$ is a multi-index. The primal grid points are

$$\mathbf{r}_{\mathbf{i}} = [i_1 \Delta r_1, i_2 \Delta r_2], \quad i_k = 0, 1, 2, \dots, N_k, \quad (4a)$$

$$\Delta r_k = \frac{1}{N_k}, \quad (4b)$$

where Δr_k is the grid spacing in coordinate direction k and N_k is the number of grid cells. We also use $\Delta r = \Delta r_1$ and $\Delta s = \Delta r_2$. The dual grid points are

$$\mathbf{r}_{\mathbf{i}} = [i_1 \Delta r_1, i_2 \Delta r_2], \quad i_k = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \dots, N_k - \frac{1}{2}, \quad (5)$$

The spatial approximation to u near the grid point \mathbf{r}_i is represented as a Taylor polynomial with $(m + 1)^2$ degrees of freedom (DOF),

$$u_{\mathbf{i}}(\mathbf{r}) = \sum_{l_1=0}^m \sum_{l_2=0}^m u_{\mathbf{i},l_1,l_2} R_i^{l_1} S_j^{l_2}, \quad (6a)$$

$$R_i \stackrel{\text{def}}{=} \frac{r - r_i}{\Delta r}, \quad S_j \stackrel{\text{def}}{=} \frac{s - s_j}{\Delta s}, \quad (6b)$$

where the integer m is the degree of the approximation and where $u_{\mathbf{i},l_1,l_2}$ is an approximation to the scaled derivative of u ,

$$u_{\mathbf{i},l_1,l_2} \approx \frac{\Delta r^{l_1}}{l_1!} \frac{\Delta s^{l_2}}{l_2!} \partial_r^{l_1} \partial_s^{l_2} u(\mathbf{r}_i). \quad (7)$$

We also denote as $u_{\mathbf{i}}$ the set of DOFs (or grid function) associated with the Taylor polynomial,

$$u_{\mathbf{i}} = \{u_{\mathbf{i},l_1,l_2}\}_{l_1,l_2=0,1,\dots,m}. \quad (8)$$

The Hermite interpolant, centered at \mathbf{r}_i , is the polynomial that interpolates the solution and its derivatives at the four neighbouring points $[i_1 \pm \frac{1}{2}, i_2 \pm \frac{1}{2}]$ and has the representation, with $(2m + 2)^2$ degrees of freedom, of the form

$$\bar{u}_{\mathbf{i}}(\mathbf{r}) = \sum_{l_1=0}^{2m+1} \sum_{l_2=0}^{2m+1} \bar{u}_{\mathbf{i},l_1,l_2} R_i^{l_1} S_j^{l_2}. \quad (9)$$

The over-bar on $\bar{u}_{\mathbf{i}}(\mathbf{r})$ will indicate that this representation has $(2m + 2)^2$ DOFs. See [Appendix C.1](#) for details on forming the interpolant.

3. Hermite algorithms

The basic structure of Hermite scheme (FOT or SOT) is given in Algorithm 1 and illustrated in Figure 1. In the algorithm, $u_{\mathbf{i}}^n$ denotes an approximation to the DOFs (solution and derivatives) at time $t^n = n\Delta t$, where Δt is the time-step. The algorithm requires a function \mathcal{I}_H to compute the Hermite interpolant, a function \mathcal{T}_H to evolve the solution over a half time-step, and a function \mathcal{B}_H to assign the boundary conditions. P denotes the index set of primal points, D the index set of dual points, and ∂P the index set of primal boundary points. Recall that the over-bar on a variable denotes a grid function with $(2m + 2)^2$ DOFs, while no over-bar is a grid function with $(m + 1)^2$ DOFs. Note that for the FOT scheme, degrees of freedom will be stored for both the solution u and its time derivative $v = \partial_t u$, but these and other details are left out to simplify the presentation.

The FOT and SOT schemes used here are the extensions, to curvilinear grids, of the *dissipative* and *conservative* schemes from [3]. More details of these curvilinear grid schemes are given in [Appendix B](#) and [Appendix C](#).

The SOT scheme has less natural dissipation than the FOT scheme. When solving problems on curvilinear grids it may be necessary to add some additional dissipation to the SOT scheme so as to retain stability without needing to reduce the time-step. Algorithm 2 shows one way to add dissipation by using the dissipative nature of the Hermite interpolant. In this algorithm the solution on the primal grid is interpolated to the dual grid, boundary conditions are applied, and then the solution is interpolated back to the primal. This process can be repeated N_s times. This smoothing stage can be applied after the second evolution stage in Algorithm 1 (i.e. after line 10). Smoothing the solution will not change the order of accuracy.

Algorithm 1 Hermite time-stepping algorithm.

```

1: function HERMITE
2:   Compute  $\Delta t$  and number of time-steps  $N_t$ .
3:   Assign initial conditions.
4:   for  $n = 1, 2, \dots, N_t$  do
5:      $t^n = (n - 1)\Delta t$ 
6:      $\bar{u}_j^n = \mathcal{I}_H(u_i^n)$ ,  $\mathbf{i} \in P, \mathbf{j} \in D$ 
7:      $u_j^{n+\frac{1}{2}} = \mathcal{T}_H(\bar{u}_j^n)$ ,  $\mathbf{j} \in D$ 
8:      $\bar{u}_k^{n+\frac{1}{2}} = \mathcal{B}_H(u_j^{n+\frac{1}{2}})$ ,  $\mathbf{k} \in \partial P$ .
9:      $\bar{u}_i^{n+\frac{1}{2}} = \mathcal{I}_H(u_j^{n+\frac{1}{2}})$ ,  $\mathbf{i} \in P, \mathbf{j} \in D$ 
10:     $u_i^{n+1} = \mathcal{T}_H(\bar{u}_i^{n+\frac{1}{2}})$ ,  $\mathbf{i} \in P$ 
11:  end for
12: end function

```

▷ Begin time-stepping loop
 ▷ Current time.
 ▷ Interpolate to dual grid.
 ▷ Evolve on dual grid to $t^n + \Delta t/2$.
 ▷ Apply BCs to primal at $t + \Delta t/2$.
 ▷ Interpolate to primal.
 ▷ Evolve on primal to $t^n + \Delta t$.
 ▷ End time-stepping loop

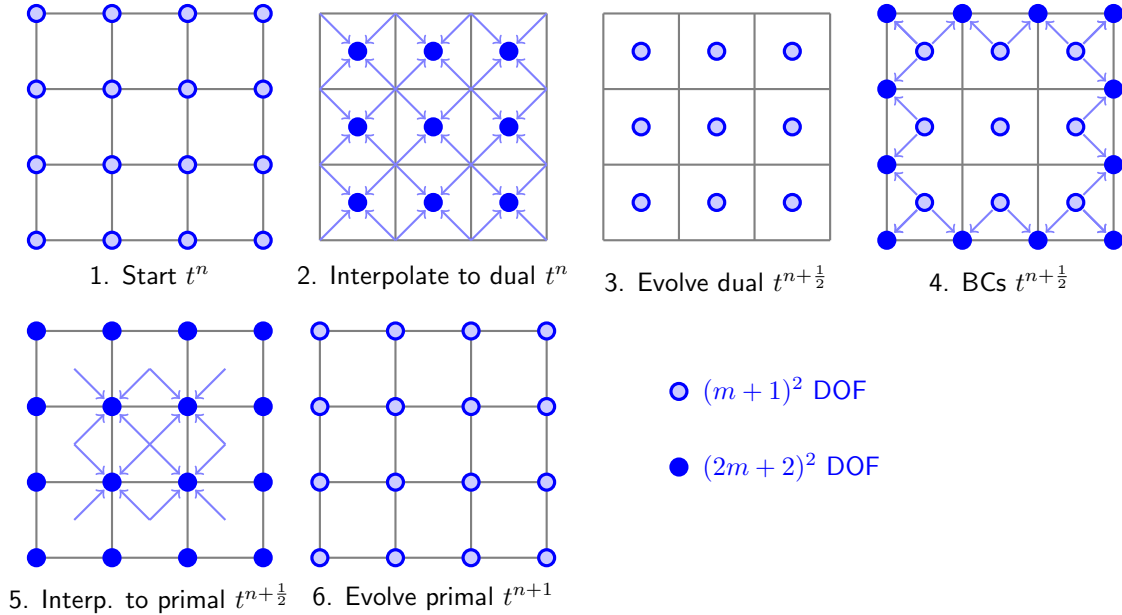


Figure 1: Stages of the Hermite FOT and SOT schemes. Open circles have $(m + 1)^2$ degrees of freedom while solid circles have $(2m + 2)^2$ degrees of freedom.

4. Compatibility boundary conditions

Compatibility boundary conditions (CBCs) for the wave equation are derived by first taking even time-derivatives of the boundary conditions. The governing equation (1a) is then used to replace even time-derivatives of u with spatial derivatives of u . The resulting conditions can be used as numerical boundary conditions. These centered conditions are generally more stable and accurate than using one sided approximations [1]. To illustrate the process for deriving CBCs, consider a Dirichlet boundary condition (Neumann BCs are similar)

$$u(\mathbf{x}, t) = g(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega. \quad (10)$$

Taking two time-derivatives of (10)

$$\partial_t^2 u(\mathbf{x}, t) = \partial_t^2 g(\mathbf{x}, t), \quad (11)$$

Algorithm 2 Hermite smoothing for the SOT scheme.

```

1: function HERMITESMOOTH( $u_i$ )
2:   for  $k = 1, 2, \dots, N_s$  do
3:      $\bar{u}_j = \mathcal{I}_H(u_i), \quad \mathbf{i} \in P, \mathbf{j} \in D$ 
4:      $\bar{u}_k = \mathcal{B}_H(u_j), \quad \mathbf{k} \in \partial P$ .
5:      $\bar{u}_i = \mathcal{I}_H(u_j), \quad \mathbf{i} \in P, \mathbf{j} \in D$ 
6:   end for
7: end function

```

▷ Apply N_s smoothing steps
▷ Interpolate to dual grid.
▷ Apply BCs to primal.
▷ Interpolate to primal.
▷ End smoothing steps

and using (1a) gives the first CBC for Dirichlet boundary conditions,

$$Lu(\mathbf{x}, t) + f(\mathbf{x}, t) = \partial_t^2 g(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega, \quad (12)$$

where L is the wave operator,

$$L \stackrel{\text{def}}{=} c^2 \Delta. \quad (13)$$

The next CBC can be derived by taking two time-derivatives of (12). The process can be repeated to derive any number of CBCs. The CBCs (such as (12)), together with tangential derivatives of the CBCs, are used to constrain the Taylor polynomial representation of the solution on the boundary.

4.1. CBCs on a Cartesian grid

Figure 2 shows a sample grid configuration in two dimensions. Given values $u_{l_1, l_2, \mathbf{j}}$ for points $\mathbf{j} \in D$ on the dual grid (open circles in Figure 2), we require values $\bar{u}_{l_1, l_2, \mathbf{k}}$ for points $\mathbf{k} \in \partial P$ on the boundary of the primal grid (solid circles in Figure 2).

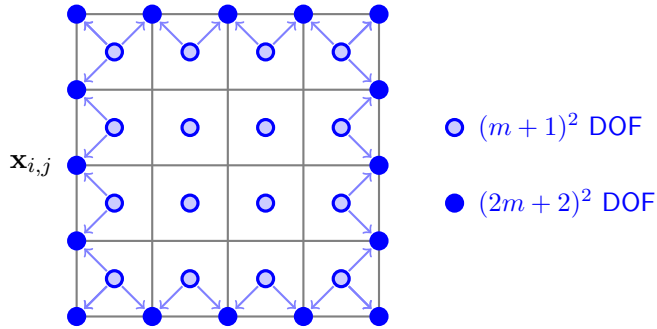


Figure 2: Compatibility boundary conditions, together with interior data, are used to define the Hermite representation on the boundary.

Consider the case of a Cartesian grid for the unit square with grid points $\mathbf{x}_i = (x_i, y_j)$. Let $\mathbf{x}_{i,j} = (x_i, y_j)$ be a point on the boundary at $x = 0$ (not a corner). The goal is to define the $(2(m+1))^2$ DOFs in the Taylor polynomial representation of the solution on the boundary,

$$\bar{u}_{i,j}(\mathbf{x}) = \sum_{l_1=0}^{2m+1} \sum_{l_2=0}^{2m+1} \bar{u}_{l_1, l_2, i, j} X_i^{l_1} Y_j^{l_2}, \quad X_i \stackrel{\text{def}}{=} \frac{x - x_i}{\Delta x}, \quad Y_j \stackrel{\text{def}}{=} \frac{y - y_j}{\Delta y}. \quad (14)$$

As indicated in Figure 2, the polynomial (14) is required to match the DOFs from the two nearby interior points. The scaled derivatives of the boundary polynomial (14) are

$$\frac{\Delta x^\alpha}{\alpha!} \frac{\Delta y^\beta}{\beta!} \partial_x^\alpha \partial_y^\beta \bar{u}_{i,j}(\mathbf{x}) = \sum_{l_1=\alpha}^{2m+1} \sum_{l_2=\beta}^{2m+1} \bar{u}_{l_1,l_2,i,j} \frac{(l_1)(\dots)(l_1-\alpha+1)}{\alpha!} \frac{(l_2)(\dots)(l_2-\beta+1)}{\beta!} X^{l_1-\alpha} Y^{l_2-\beta}, \quad (15)$$

and these, when evaluated at $(x_i + \Delta x/2, y_j \pm \Delta y/2)$, are matched to the given interior dual (scaled) derivatives,

$$u_{\alpha,\beta,i_1+\frac{1}{2},i_2\pm\frac{1}{2}}, \quad \alpha, \beta = 0, 1, \dots, m. \quad (16)$$

to give the $2(m+1)^2$ interpolation conditions,

$$\sum_{l_1=\alpha}^{2m+1} \sum_{l_2=\beta}^{2m+1} \binom{l_1}{\alpha} \binom{l_2}{\beta} \left[\frac{1}{2}\right]^{l_1-\alpha} \left[\pm\frac{1}{2}\right]^{l_2-\beta} \bar{u}_{l_1,l_2,\mathbf{i}} = u_{\alpha,\beta,i_1+\frac{1}{2},i_2\pm\frac{1}{2}}, \quad \alpha, \beta = 0, 1, \dots, m. \quad (17)$$

A further $2(m+1)^2$ conditions are needed to uniquely determine $\bar{u}_{l_1,l_2,\mathbf{i}}$ and these are obtained using CBCs.

Dirichlet boundary conditions. Consider a Dirichlet boundary condition at $x = 0$

$$u(0, y, t) = g(y, t). \quad (18)$$

The CBCs are

$$\partial_y^\alpha L^q \bar{u}_{\mathbf{i}}(0, y_j) = \partial_t^{2q} \partial_y^\alpha g(y_j, t), \quad q = 0, 1, \dots, m, \quad \alpha = 0, 1, \dots, 2m+1. \quad (19)$$

Using the binomial expansion gives

$$\partial_y^\alpha L^q = c^{2q} \partial_y^\alpha (\partial_x^2 + \partial_y^2)^q = c^{2q} \sum_{k=0}^q \binom{q}{k} \partial_x^{2(q-k)} \partial_y^{2k+\alpha}. \quad (20)$$

This leads to the $2(m+1)^2$ CBC conditions

$$c^{2q} \sum_{k=0}^q \binom{q}{k} \frac{(2(q-k))!}{\Delta x^{2(q-k)}} \frac{(2k+\alpha)!}{\Delta y^{2k+\alpha}} \bar{u}_{2(q-k),2k+\alpha,\mathbf{i}} = \partial_t^{2q} \partial_y^\alpha g(y_j, t), \quad (21)$$

for $q = 0, 1, \dots, m$, and $\alpha = 0, 1, \dots, 2m+1$. Note that in (21), only terms with $2k+\alpha \leq 2m+1$ should be kept. Equations (17) together with (21) define a linear system of equations for the unknowns $\bar{u}_{l_1,l_2,\mathbf{i}}$ on a Dirichlet boundary.

Neumann boundary conditions. Now consider a Neumann boundary condition at $x = 0$,

$$\partial_x u(0, y, t) = g(y, t), \quad (22)$$

The CBCs are

$$\partial_y^\alpha \partial_x L^q p(0, y_j) = \partial_t^{2q} \partial_y^\alpha g(y_j, t), \quad q = 0, 1, \dots, m, \quad \alpha = 0, 1, \dots, 2m + 1. \quad (23)$$

This leads to the CBC conditions

$$c^{2q} \sum_{k=0}^q \binom{q}{k} \frac{(2(q-k)+1)!}{\Delta x^{2(q-k)+1}} \frac{(2k+\alpha)!}{\Delta y^{2k+\alpha}} \bar{u}_{2(q-k)+1, 2k+\alpha, \mathbf{i}} = \partial_t^{2q} \partial_y^\alpha g(y_j, t), \quad (24)$$

for $q = 0, 1, \dots, m$, and $\alpha = 0, 1, \dots, 2m + 1$. Equations (17) together with (24) define a linear system of equations for the unknowns $\bar{u}_{l_1, l_2, \mathbf{i}}$ on a Neumann boundary.

4.2. CBCs for corners on a Cartesian grid

Now consider assigning the solution at the corner point, such as the solution at the lower left point, $\mathbf{x}_\mathbf{i} = [0, 0]$, in Figure 2. At this corner there is one interior neighbour on the dual grid at index $\mathbf{i} = [\frac{1}{2}, \frac{1}{2}]$. Following the discussion for a point on the interior of a side, the conditions to match the coefficients in the Hermite polynomial to coefficients of the polynomial at the interior point are the $(m+1)^2$ conditions,

$$\sum_{l_1=\alpha}^{2m+1} \sum_{l_2=\beta}^{2m+1} \binom{l_1}{\alpha} \binom{l_2}{\beta} \left[\frac{1}{2}\right]^{l_1-\alpha} \left[\frac{1}{2}\right]^{l_2-\beta} \bar{u}_{l_1, l_2, \mathbf{i}} = u_{\alpha, \beta, i_1+\frac{1}{2}, i_2+\frac{1}{2}}, \quad \alpha, \beta = 0, 1, \dots, m. \quad (25)$$

A further $3(m+1)^2$ conditions are needed to uniquely determine $\bar{u}_{l_1, l_2, \mathbf{i}}$ and these are obtained using CBCs. In subsequent sub-sections we discuss Dirichlet-Dirichlet (D-D), Neumann-Neumann (N-N), and Dirichlet-Neumann corners (D-N). This discussion closely follows that given in [1] but is adjusted to the setting of Hermite methods.

4.2.1. Dirichlet-Dirichlet CBC corner

Consider a Dirichlet-Dirichlet corner at $\mathbf{x} = \mathbf{0}$,

$$u(x, 0, t) = g_1(x, t), \quad (26a)$$

$$u(0, y, t) = g_2(y, t). \quad (26b)$$

The CBCs are

$$\partial_x^\alpha L^q \bar{u}_\mathbf{i}(\mathbf{x}_\mathbf{i}) = \partial_x^\alpha \partial_t^{2q} g_1(0, t), \quad q = 0, 1, \dots, m, \quad (27a)$$

$$\partial_y^\alpha L^q \bar{u}_\mathbf{i}(\mathbf{x}_\mathbf{i}) = \partial_y^\alpha \partial_t^{2q} g_2(0, t), \quad q = 0, 1, \dots, m, \quad (27b)$$

for $\alpha \in \mathcal{M}_q$, where \mathcal{M}_q is the set of integers,

$$\mathcal{M}_q \stackrel{\text{def}}{=} \{0, 1, \dots, 2m+1\} - \{0, 2, 4, \dots, 2(q-1)\}. \quad (28)$$

The set \mathcal{M}_q consists of the integers from 0 to $2m+1$ minus the even integers from 0 to $2(q-1)$. In addition, the conditions in (27) should be averaged when $\alpha = 2q$. The set \mathcal{M}_q was derived in [1]. At a corner, the union of the compatibility conditions from adjacent faces are not all independent

and the set \mathcal{M}_q was determined by choosing a independent set of conditions by examining the CBCs in the case of a Cartesian grid. As an example, when $q = 1$, \mathcal{M}_1 is missing 0,

$$\mathcal{M}_1 = \{1, 2, 3, 4, 5, \dots, 2m + 1\}, \quad (29)$$

and we average the conditions when $\alpha = 2$,

$$\partial_x L \bar{u}_i(\mathbf{x}_i) = \partial_t^{2q} \partial_x \partial_t^2 g_1(0, t), \quad (\alpha = 1), \quad (30a)$$

$$\partial_y L \bar{u}_i(\mathbf{x}_i) = \partial_t^{2q} \partial_y \partial_t^2 g_2(0, t), \quad (\alpha = 1), \quad (30b)$$

$$\frac{1}{2}(\partial_x^2 L \bar{u}_i(\mathbf{x}_i) + \partial_y^2 L \bar{u}_i(\mathbf{x}_i)) = \frac{1}{2}(\partial_x^2 \partial_t^2 g_1(0, t) + \partial_y^2 \partial_t^2 g_2(0, t)), \quad (\alpha = 2), \quad (30c)$$

$$\partial_x^\alpha L \bar{u}_i(\mathbf{x}_i) = \partial_t^{2q} \partial_x^\alpha \partial_t^2 g_1(0, t), \quad \alpha = 3, 4, 5, \dots, 2m + 1 \quad (30d)$$

$$\partial_y^\alpha L \bar{u}_i(\mathbf{x}_i) = \partial_t^{2q} \partial_y^\alpha \partial_t^2 g_2(0, t), \quad \alpha = 3, 4, 5, \dots, 2m + 1 \quad (30e)$$

When $q = 2$, \mathcal{M}_2 is missing $\{0, 2\}$,

$$\mathcal{M}_2 = \{1, 3, 4, 5, \dots, 2m + 1\}, \quad (31)$$

and we average the conditions when $\alpha = 4$,

$$\partial_x^\alpha L^2 \bar{u}_i(\mathbf{x}_i) = \partial_t^{2q} \partial_x^\alpha \partial_t^4 g_1(0, t), \quad \alpha = 1, 3, \quad (32a)$$

$$\partial_y^\alpha L^2 \bar{u}_i(\mathbf{x}_i) = \partial_t^{2q} \partial_y^\alpha \partial_t^4 g_2(0, t), \quad \alpha = 1, 3, \quad (32b)$$

$$\frac{1}{2}(\partial_x^4 L^2 \bar{u}_i(\mathbf{x}_i) + \partial_y^4 L^2 \bar{u}_i(\mathbf{x}_i)) = \frac{1}{2}(\partial_x^4 \partial_t^4 g_1(0, t) + \partial_y^4 \partial_t^4 g_2(0, t)), \quad (\alpha = 4), \quad (32c)$$

$$\partial_x^\alpha L^2 \bar{u}_i(\mathbf{x}_i) = \partial_t^{2q} \partial_x^\alpha \partial_t^4 g_1(0, t), \quad \alpha = 5, \dots, 2m + 1 \quad (32d)$$

$$\partial_y^\alpha L^2 \bar{u}_i(\mathbf{x}_i) = \partial_t^{2q} \partial_y^\alpha \partial_t^4 g_2(0, t), \quad \alpha = 5, \dots, 2m + 1 \quad (32e)$$

In summary, the CBC D-D corner conditions are (or an average of these conditions when $\alpha = 2q$)

$$c^{2q} \sum_{j=0}^q \binom{q}{j} \frac{(2(q-j)+\alpha)!}{\Delta x^{2(q-j)+\alpha}} \frac{(2j)!}{\Delta y^{2j}} \bar{u}_{2(q-j)+\alpha, 2j, \mathbf{i}} = \partial_t^{2q} \partial_x^\alpha g_1(0, t), \quad (33a)$$

$$c^{2q} \sum_{j=0}^q \binom{q}{j} \frac{(2(q-j))!}{\Delta x^{2(q-j)}} \frac{(2j+\alpha)!}{\Delta y^{2j+\alpha}} \bar{u}_{2(q-j), 2j+\alpha, \mathbf{i}} = \partial_t^{2q} \partial_y^\alpha g_2(0, t), \quad (33b)$$

for $q = 0, 1, 2, \dots, m$ and $\alpha \in \mathcal{M}_q$. Note that only entries with valid indices l_1 and l_2 for $\bar{u}_{l_1, l_2, \mathbf{i}}$ in (33) should be kept. Equations (25) and (33) define a linear system of equations to determine the Hermite coefficients in the D-D orner at $\mathbf{x} = \mathbf{0}$.

Here we count the number of equations to ensure we have a square system. The set \mathcal{M}_q has $2m + 2 - 2q$ elements. From (27) there are $2(2m + 2 - 2q) - 1$ equations obtained, the minus one is a result of averaging when $\alpha = 2q$. Thus, the total number of equations is given by

$$\sum_{q=0}^m 2(2m + 2 - 2q) - 1 = 3(m + 1)^2, \quad (34)$$

which is the number of equations required. The arguments for obtaining a square system for the remaining cases follow a similar computation.

4.2.2. Neumann-Neumann CBC corner

Now consider a Neumann-Neumann corner at $\mathbf{x} = \mathbf{0}$,

$$\partial_y u(x, 0, t) = g_1(x, t), \quad (35a)$$

$$\partial_x u(0, y, t) = g_2(y, t). \quad (35b)$$

The CBCs are

$$\partial_x^\alpha \partial_y L^q \bar{u}_i(\mathbf{x}_i) = \partial_x^\alpha \partial_t^{2q} g_1(0, t), \quad q = 0, 1, \dots, m, \quad (36a)$$

$$\partial_y^\alpha \partial_x L^q \bar{u}_i(\mathbf{x}_i) = \partial_y^\alpha \partial_t^{2q} g_2(0, t), \quad q = 0, 1, \dots, m. \quad (36b)$$

Following the argument from the previous section, the CBC N-N corner conditions are thus (or an average of these conditions when $\alpha = 2q + 1$)

$$c^{2q} \sum_{j=0}^q \binom{q}{j} \frac{(2(q-j) + \alpha)!}{\Delta x^{2(q-j) + \alpha}} \frac{(2j+1)!}{\Delta y^{2j+1}} \bar{u}_{2(q-j) + \alpha, 2j+1, \mathbf{i}} = \partial_t^{2q} \partial_x^\alpha g_1(0, t), \quad (37a)$$

$$c^{2q} \sum_{j=0}^q \binom{q}{j} \frac{(2(q-j) + 1)!}{\Delta x^{2(q-j) + 1}} \frac{(2j + \alpha)!}{\Delta y^{2j + \alpha}} \bar{u}_{2(q-j) + 1, 2j + \alpha, \mathbf{i}} = \partial_t^{2q} \partial_y^\alpha g_2(0, t), \quad (37b)$$

for $q = 0, 1, 2, \dots, 2m + 1$ and $\alpha \in \mathcal{N}_q$. Here \mathcal{N}_q is the set

$$\mathcal{N}_q \stackrel{\text{def}}{=} \{0, 1, \dots, 2m + 1\} - \{1, 3, 5, 7, \dots, 2q - 1\}. \quad (38)$$

The set \mathcal{N}_q consists of the integers from 0 to m minus the odd integers from 1 to $2q - 1$. In addition the conditions in (37) should be averaged when $\alpha = 2q + 1$. *As for the case of a D-D corner, the set \mathcal{N}_q was derived by choosing an independent set of conditions by examining the CBCs in the case of a Cartesian grid.* Note that only entries with valid indices l_1 and l_2 for $\bar{u}_{l_1, l_2, \mathbf{i}}$ in (37) should be kept. Equations (25) and (37) define a linear system of equations to determine the Hermite coefficients in the N-N corner at $\mathbf{x} = \mathbf{0}$.

4.2.3. Dirichlet-Neumann CBC corner

Consider a Dirichlet-Neumann corner at $\mathbf{x} = \mathbf{0}$, with Neumann on the bottom face, and Dirichlet on the left face,

$$\partial_y u(x, 0, t) = g_1(x, t), \quad (39a)$$

$$u(0, y, t) = g_2(y, t). \quad (39b)$$

The CBCs are

$$\partial_x^\alpha \partial_y L^q \bar{u}_i(\mathbf{x}_i) = \partial_x^\alpha \partial_t^{2q} g_1(0, t), \quad q = 0, 1, \dots, m, \quad (40a)$$

$$\partial_y^\beta L^q \bar{u}_i(\mathbf{x}_i) = \partial_y^\beta \partial_t^{2q} g_2(0, t), \quad q = 0, 1, \dots, m. \quad (40b)$$

Following the argument from the previous section, the CBC D-N corner conditions are thus (or an average of these conditions when $\alpha = 2q + 1$)

$$c^{2q} \sum_{j=0}^q \binom{q}{j} \frac{(2(q-j) + \alpha)!}{\Delta x^{2(q-j) + \alpha}} \frac{(2j+1)!}{\Delta y^{2j+1}} \bar{u}_{2(q-j) + \alpha, 2j+1, \mathbf{i}} = \partial_t^{2q} \partial_x^\alpha g_1(0, t), \quad (41a)$$

$$c^{2q} \sum_{j=0}^q \binom{q}{j} \frac{(2(q-j))!}{\Delta x^{2(q-j)}} \frac{(2j + \beta)!}{\Delta y^{2j + \beta}} \bar{u}_{2(q-j) + 1, 2j + \beta, \mathbf{i}} = \partial_t^{2q} \partial_y^\beta g_2(0, t), \quad (41b)$$

for $q = 0, 1, 2, \dots, m$ and $(\alpha, \beta) \in \mathcal{M}_q \times \mathcal{N}_q$. Note that α , corresponding to the Neumann BC, is in the set \mathcal{M}_q associated with the Dirichlet BC on the face, while β , corresponding to the Dirichlet BC is associated with \mathcal{N}_q . In addition the conditions in (41) should be averaged when $(\alpha, \beta) = (2q + 1, 2q)$. Note that only entries with valid indices l_1 and l_2 for $\bar{u}_{l_1, l_2, \mathbf{i}}$ in (41) should be kept. Equations (25) and (41) define a linear system of equations to determine the Hermite coefficients in the D-N corner at $\mathbf{x} = \mathbf{0}$.

4.3. CBCs on a curvilinear grid

We now consider the imposition of CBCs on a curvilinear grid. As for the Cartesian grid case, the coefficients in the Taylor polynomial representation of the solution for a point on the boundary will be determined from known interior data together with CBCs. The CBCs become algebraically more complicated on a curvilinear grid and rather than writing a general formula such as (21), a recursion is used to form the equations implied by the CBCs. Let

$$\bar{u}_{\mathbf{i}}(\mathbf{r}) = \sum_{l_1=0}^{2m+1} \sum_{l_2=0}^{2m+1} \bar{u}_{l_1, l_2, \mathbf{i}} R_i^{l_1} S_j^{l_2}, \quad R_i \stackrel{\text{def}}{=} \frac{r - r_i}{\Delta r}, \quad S_j \stackrel{\text{def}}{=} \frac{s - s_j}{\Delta s}, \quad (42)$$

denote the Taylor polynomial representation for the solution at a point on the boundary $\mathbf{r}_{\mathbf{i}} = (r_i, s_j)$. Furthermore, let $L\bar{u}_{\mathbf{i}}$ have the Taylor polynomial representation

$$L\bar{u}_{\mathbf{i}}(\mathbf{r}) = \sum_{l_1=0}^{2m+1} \sum_{l_2=0}^{2m+1} \bar{d}_{l_1, l_2, \mathbf{i}} R_i^{l_1} S_j^{l_2}, \quad (43)$$

where the application of wave operator L to the Hermite representation is described in Appendix C.3.

4.3.1. Dirichlet CBCs on a curvilinear grid

The CBCs for a Dirichlet boundary condition at the point $\mathbf{r}_{\mathbf{i}} = (r_i, s_j)$ on the boundary at $r = 0$, are

$$\partial_s^\alpha L^q \bar{u}_{\mathbf{i}}(\mathbf{r}_{\mathbf{i}}) = \partial_s^\alpha \partial_t^{2q} g(s_j, t), \quad q = 0, 1, \dots, m, \quad \alpha = 0, 1, \dots, 2m + 1. \quad (44)$$

For $q = 0$ this gives the conditions

$$\frac{\alpha!}{\Delta s^\alpha} \bar{u}_{0, \alpha} = \partial_s^\alpha g(s_j, t), \quad \alpha = 0, 1, \dots, 2m + 1. \quad (45)$$

Let $\bar{\mathbf{u}} \in \mathbb{R}^{(2(m+1))^2}$ and $\bar{\mathbf{d}} \in \mathbb{R}^{(2(m+1))^2}$ denote the vectors with components $\bar{u}_{l_1, l_2, \mathbf{i}}$ and $\bar{d}_{l_1, l_2, \mathbf{i}}$, for $l_1, l_2 = 0, 1, \dots, 2m + 1$ and fixed \mathbf{i} ,

$$\bar{\mathbf{u}} = [\bar{u}_{0,0,\mathbf{i}} \quad \bar{u}_{1,0,\mathbf{i}} \quad \bar{u}_{2,0,\mathbf{i}} \quad \dots \quad \bar{u}_{2m+1,2m+1,\mathbf{i}}]^T, \quad (46)$$

$$\bar{\mathbf{d}} = [\bar{d}_{0,0,\mathbf{i}} \quad \bar{d}_{1,0,\mathbf{i}} \quad \bar{d}_{2,0,\mathbf{i}} \quad \dots \quad \bar{d}_{2m+1,2m+1,\mathbf{i}}]^T, \quad (47)$$

where the dependence of $\bar{\mathbf{u}}$ and $\bar{\mathbf{d}}$ on \mathbf{i} has been suppressed. We then have the following relationship between the coefficients of $\bar{u}_{\mathbf{i}}$ and $L\bar{u}_{\mathbf{i}}$,

$$\bar{\mathbf{d}} = L_H \bar{\mathbf{u}}, \quad (48)$$

where L_H is the matrix implied by (43) (for details see Appendix C.4). The CBC (45) for $q = 0$ can then be expressed as

$$\frac{\alpha!}{\Delta s^\alpha} \mathbf{e}_{[0,\alpha]}^T \bar{\mathbf{u}} = \partial_s^\alpha g(s_j, t), \quad \alpha = 0, 1, \dots, 2m + 1, \quad (49)$$

where $\mathbf{e}_{[0,\alpha]} = \mathbf{e}_{\alpha(2m+1)}$ denotes the unit vector corresponding to the entry $\bar{u}_{0,\alpha}$ in the vector $\bar{\mathbf{u}}$. In general we have

$$\frac{\alpha!}{\Delta s^\alpha} \mathbf{e}_{[0,\alpha]}^T L_H^q \bar{\mathbf{u}} = \partial_s^\alpha \partial_t^{2q} g(s_j, t), \quad q = 0, 1, \dots, m, \quad \alpha = 0, 1, \dots, 2m + 1. \quad (50)$$

The CBCs are thus

$$\frac{\alpha!}{\Delta s^\alpha} \left[\mathbf{z}_\alpha^q \right]^T \bar{\mathbf{u}} = \partial_s^\alpha g(s_j, t), \quad q = 0, 1, \dots, m, \quad \alpha = 0, 1, \dots, 2m + 1, \quad (51a)$$

where the vectors \mathbf{z}_α^q satisfy the recursion

$$\left[\mathbf{z}_\alpha^0 \right]^T = \mathbf{e}_{[0,\alpha]}^T, \quad (51b)$$

$$\left[\mathbf{z}_\alpha^q \right]^T = \left[\mathbf{z}_\alpha^{q-1} \right]^T L_H, \quad q = 1, 2, \dots, m. \quad (51c)$$

The equations in (51) together with the matching conditions (17) define a linear system of equations to determine the unknowns $\bar{u}_{l_1, l_2, \mathbf{i}}$.

Note that the form of (51) for a Cartesian grid is just the same as (51) but with L being the Cartesian grid version and ∂_n and ∂_s being replaced by one of ∂_x or ∂_y .

4.3.2. Neumann CBCs on a curvilinear grid

The CBCs for a Neumann BC at the point $\mathbf{r}_\mathbf{i}$ on the boundary at $r = 0$ are

$$\partial_s^\alpha \partial_n L^q \bar{u}_\mathbf{i}(\mathbf{r}_\mathbf{i}) = \partial_s^\alpha \partial_t^{2q} g(s_j, t), \quad q = 0, 1, \dots, m, \quad \alpha = 0, 1, \dots, 2m + 1, \quad (52a)$$

where the normal derivative is

$$\partial_n = \mathbf{n} \cdot \nabla = n_1 \partial_x + n_2 \partial_y = (n_1 r_x + n_2 r_y) \partial_r + (n_1 s_x + n_2 s_y) \partial_s = b_1(r, s) \partial_r + b_2(r, s) \partial_s. \quad (52b)$$

Let $\partial_n \bar{u}_\mathbf{i}$ have the Taylor polynomial representation

$$\partial_n \bar{u}_\mathbf{i}(\mathbf{r}) = \sum_{l_1=0}^{2m+1} \sum_{l_2=0}^{2m+1} \bar{n}_{l_1, l_2, \mathbf{i}} R_i^{l_1} S_j^{l_2}. \quad (53)$$

Following the previous section, let $\bar{\mathbf{u}}$ denote the vector of coefficients for $\bar{u}_{l_1, l_2, \mathbf{i}}$. In terms of this vector of unknowns, the Neumann CBCs can then be expressed as

$$\frac{\alpha!}{\Delta s^\alpha} \mathbf{e}_{[0,\alpha]}^T N_H L_H^q \bar{\mathbf{u}} = \partial_s^\alpha \partial_t^{2q} g(s_j, t), \quad (54a)$$

or

$$\frac{\alpha!}{\Delta s^\alpha} \mathbf{w}_\alpha^T L_H^q \bar{\mathbf{u}} = \partial_s^\alpha \partial_t^{2q} g(s_j, t), \quad (54b)$$

$$\mathbf{w}_\alpha^T \stackrel{\text{def}}{=} \mathbf{e}_{[0,\alpha]}^T N_H. \quad (54c)$$

where N_H is the matrix implied by (53). The vectors

$$\left[\mathbf{z}_\alpha^q\right]^T \stackrel{\text{def}}{=} \mathbf{w}_\alpha^T L_H^q, \quad (55a)$$

can be computed with the recursion

$$\left[\mathbf{z}_\alpha^0\right]^T = \mathbf{e}_{[0,\alpha]}^T N_H, \quad (55b)$$

$$\left[\mathbf{z}_\alpha^q\right]^T = \left[\mathbf{z}_\alpha^{q-1}\right]^T L_H, \quad q = 1, 2, \dots, m, \quad (55c)$$

and the CBCs are

$$\frac{\alpha!}{\Delta s^\alpha} \left[\mathbf{z}_\alpha^q\right]^T \bar{\mathbf{u}} = \partial_s^\alpha \partial_t^{2q} g(s_j, t). \quad (55d)$$

The equations in (55) together with the matching conditions (17) define a linear system of equations to determine the unknowns $\bar{u}_{l_1, l_2, i}$.

4.3.3. Corner CBCs on a curvilinear grid

The CBCs at a corner follow the discussion in Section 4.2 for a Cartesian grid, except that the equations for the CBCs for the Dirichlet or Neumann case on a Cartesian grid should be replaced with the corresponding equations for a curvilinear grid. **For example, at a D-D corner one should use the interpolation conditions (25), and replace the Cartesian grid compatibility conditions (27) with the corresponding curvilinear variants such as (44).**

5. Analysis of the CBCs

In this section, we analyze the compatibility boundary conditions developed in Section 4. We consider the solvability and conditioning of the resulting linear system of equations. We also analyze the symmetry properties that arise when using CBCs on a Cartesian grid.

We write the boundary conditions developed in Section 4 as a linear system

$$M\bar{\mathbf{u}} = \mathbf{b}, \quad (56)$$

where $\bar{\mathbf{u}}$ and \mathbf{b} are vectors of degree $4(m+1)^2$ and M is a matrix of size $4(m+1)^2 \times 4(m+1)^2$. This system consists of the interpolation conditions (17) together with the appropriate CBCs. In order for this system to be solvable we require the matrix M to be non-singular. We aim to derive a set of conditions that guarantees solvability.

5.1. Solvability of the CBC matrix systems on Cartesian grids

The equations appearing in M should be scaled to improve the conditioning of the matrix. A standard approach to improve the conditioning is to scale each row by the largest entry in absolute value. This will be called row scaling. Further improvements to the conditioning can be obtained through a process known as equilibration⁴. For Cartesian grids we consider the max-norm condition number of M defined in the usual way as

$$\kappa_\infty(M) \stackrel{\text{def}}{=} \|M\|_\infty \|M^{-1}\|_\infty, \quad (57)$$

⁴The Matlab function `equilibrate` permutes and rescales the matrix to have diagonal entries of magnitude one and off-diagonal entries of magnitude at most one. The HSL Mathematical Software Library [43], <http://www.hsl.rl.ac.uk>, also has scaling routines that can be used to improve the conditioning.

where the max-norm of a matrix is the maximum row-sum of the magnitudes of the matrix elements. We consider, without loss of generality, a boundary at $x = 0$. Let γ denote the tall-cell ratio,

$$\gamma \stackrel{\text{def}}{=} \frac{\Delta x}{\Delta y}. \quad (58)$$

Theorem 1 (CBC solvability for Cartesian grids). *The matrix M resulting from the CBC equations on a Cartesian grid with Dirichlet or Neumann boundary conditions or at a corner where Dirichlet or Neumann boundary conditions meet is nonsingular for $m = 1, 2, 3, 4$. Furthermore, the max-norm condition number of the row-scaled M only depends on the ratio $\gamma = \Delta x/\Delta y$ and thus remains unchanged as the mesh is refined.*

The proof of Theorem 1 for $m = 1, 2, 3, 4$ is given in [Appendix A](#). We conjecture that the result holds for any positive integer m . As an example, here is the form of the condition number for $m = 1$, for Dirichlet and Neumann boundaries,

$$\kappa_\infty(M) = \max(41, 28 + 3\gamma^2) \times \max\left(\frac{121}{16}, 1 + \gamma^2\right), \quad (\text{Dirichlet}), \quad (59a)$$

$$\kappa_\infty(M) = \max\left(\frac{125}{16}, 1 + \gamma^2, \frac{21}{4} + \frac{3\gamma^2}{4}\right) \times \max\left(\frac{121}{16}, 1 + \gamma^2\right), \quad (\text{Neumann}). \quad (59b)$$

Note that the condition numbers in (59) scale in proportion to γ^2 and thus become large as γ get large. This is a known phenomena from other discretizations that use CBCs: the grid spacing normal to the boundary should not be [too](#) large compared to the grid spacings in the tangential directions.

Table 1 summarizes the condition numbers for Dirichlet and Neumann boundaries and corners for $\gamma = 1$. Results for row-scaling and equilibration (Matlab) of the matrix are shown. Equilibration roughly reduces the condition number by a factor of 10 over row-scaling.

CBC Condition Number $\kappa_\infty(M)$ for Cartesian Grids										
BC	$m = 1$		$m = 2$		$m = 3$		$m = 4$		$m = 5$	
	rs	eq	rs	eq	rs	eq	rs	eq	rs	eq
D	3.1e2	4.8e1	5.8e3	7.2e2	1.3e5	1.2e4	3.8e6	1.7e5	9.7e7	2.5e6
N	5.9e1	1.4e1	7.0e2	1.8e2	1.4e4	3.6e3	3.3e5	4.4e4	7.5e6	5.9e5
D-D	5.4e2	4.0e1	8.3e3	5.4e2	1.8e5	9.0e3	4.8e6	1.4e5	1.2e8	2.1e6
N-N	5.7e1	7.0e1	2.7e2	2.6e2	2.9e3	6.7e2	3.8e4	1.1e4	6.3e5	1.7e5
D-N	1.6e2	1.8e1	1.4e3	5.0e1	2.2e4	2.9e3	4.2e5	1.4e5	8.5e6	7.8e5

Table 1: Max-norm condition numbers of the CBC matrices with row-scaling (rs) and equilibration (eq) for $\gamma = \Delta x/\Delta y = 1$. D denotes a Dirichlet BC, N a Neumann BC, D-D a Dirichlet-Dirichlet corner and so on.

5.2. Solvability of CBC matrix systems on curvilinear grids

To study the solvability of the CBC matrix on a curvilinear grid we freeze coefficients near a point on the boundary and consider the wave equation with the constant coefficient operator

$$L_0 = c^{20} \partial_r^2 + 2c^{11} \partial_r \partial_s + c^{02} \partial_s^2 + c^{10} \partial_r + c^{01} \partial_s. \quad (60)$$

The matrix M in the CBC matrix can be formed symbolically for this case and it's determinant can be evaluated. This leads to the following result.

Theorem 2 (CBC solvability for Curvilinear grids). *The CBC matrix M for the constant-coefficient operator L_0 in (60) with Dirichlet or Neumann boundary conditions, for $m = 1, 2, 3, 4$, is non-singular provided $c^{20} > 0$ and Δr is sufficiently small (left or right face) or $c^{02} > 0$ and Δs is sufficiently small (top or bottom face). If $c^{10} = 0$ (left or right face) or $c^{01} = 0$ (top or bottom face), then the matrix is non-singular for any Δr and Δs .*

Proof. We will focus on the left boundary at $r = 0$, the arguments for the other boundaries are done similarly. For Dirichlet or Neumann boundary conditions, the determinant of the matrix M , has the form

$$\det(M) = K_m G_m(\xi), \quad \xi \stackrel{\text{def}}{=} \frac{c^{10} \Delta r}{2c^{20}}, \quad m = 1, 2, 3, 4, \quad (61)$$

where K_m is a non-zero constant that depends on Δr , Δs and c^{20} , and $G_m(\xi)$ is a polynomial with $G_m(0) = 1$. For Dirichlet boundary conditions the polynomials are given by

$$G_1(\xi) = \left[1 - \frac{\xi}{4}\right]^4, \quad (62a)$$

$$G_2(\xi) = \left[1 - \frac{9}{16}\xi + \frac{3}{32}\xi^2 - \frac{1}{192}\xi^3\right]^6, \quad (62b)$$

$$G_3(\xi) = \left[1 - \frac{29}{32}\xi + \frac{5}{16}\xi^2 - \frac{7}{128}\xi^3 + \frac{1}{192}\xi^4 - \frac{1}{3840}\xi^5 + \frac{1}{184320}\xi^6\right]^8, \quad (62c)$$

$$G_4(\xi) = \left[1 - \frac{325}{256}\xi + \frac{345}{512}\xi^2 - \frac{155}{768}\xi^3 + \frac{235}{6144}\xi^4 - \frac{99}{20480}\xi^5 + \frac{245}{589824}\xi^6\right. \quad (62d)$$

$$\left. - \frac{199}{8257536}\xi^7 + \frac{5}{5505024}\xi^8 - \frac{1}{49545216}\xi^9 + \frac{1}{4954521600}\xi^{10}\right]^{10}. \quad (62e)$$

For Neumann boundary conditions the corresponding polynomials are

$$G_1(\xi) = \left[1 - \frac{\xi}{2}\right]^4, \quad (63a)$$

$$G_2(\xi) = \left[1 - \xi + \frac{1}{4}\xi^2 - \frac{1}{48}\xi^3\right]^6, \quad (63b)$$

$$G_3(\xi) = \left[1 - \frac{3}{2}\xi + \frac{3}{4}\xi^2 - \frac{35}{192}\xi^3 + \frac{3}{128}\xi^4 - \frac{1}{640}\xi^5 + \frac{1}{23040}\xi^6\right]^8, \quad (63c)$$

$$G_4(\xi) = \left[1 - 2\xi + \frac{3}{2}\xi^2 - \frac{233}{384}\xi^3 + \frac{29}{192}\xi^4 - \frac{63}{2560}\xi^5 + \frac{31}{11520}\xi^6\right. \quad (63d)$$

$$\left. - \frac{127}{645120}\xi^7 + \frac{1}{107520}\xi^8 - \frac{1}{3870720}\xi^9 + \frac{1}{309657600}\xi^{10}\right]^{10}. \quad (63e)$$

For ξ sufficiently small it can be seen that $G_m(\xi) > 0$ and thus the matrix M is nonsingular. In particular, Table 2 gives the largest $\xi = \xi_{\max}$ such that $G_m(\xi) > 0$ for $0 \leq \xi \leq \xi_{\max}$ for the functions above. \square

To study the actual conditioning of the CBC matrices in practice, Figure 3 graphs $\kappa_{\infty}(M)$ for the CBC matrices on two curvilinear grids as the mesh is refined. The polynomial mapping and the X mapping defined in Section 6.1 are considered (see Figure 4) for different boundary and corner conditions. Results are shown for the row scaled matrix and the equilibrated matrix. The condition numbers are seen to increase with the Hermite degree m . The condition numbers are fairly constant or sometimes decrease as Δr decreases.

BC	ξ_{\max}			
	G_1	G_2	G_3	G_4
Dirichlet	4.00	2.23	2.65	2.35
Neumann	2.00	1.48	2.23	1.08

Table 2: Largest $\xi = \xi_{\max}$ such that $G_m(\xi) > 0$ for $0 \leq \xi \leq \xi_{\max}$ for the Dirichlet and Neumann case.

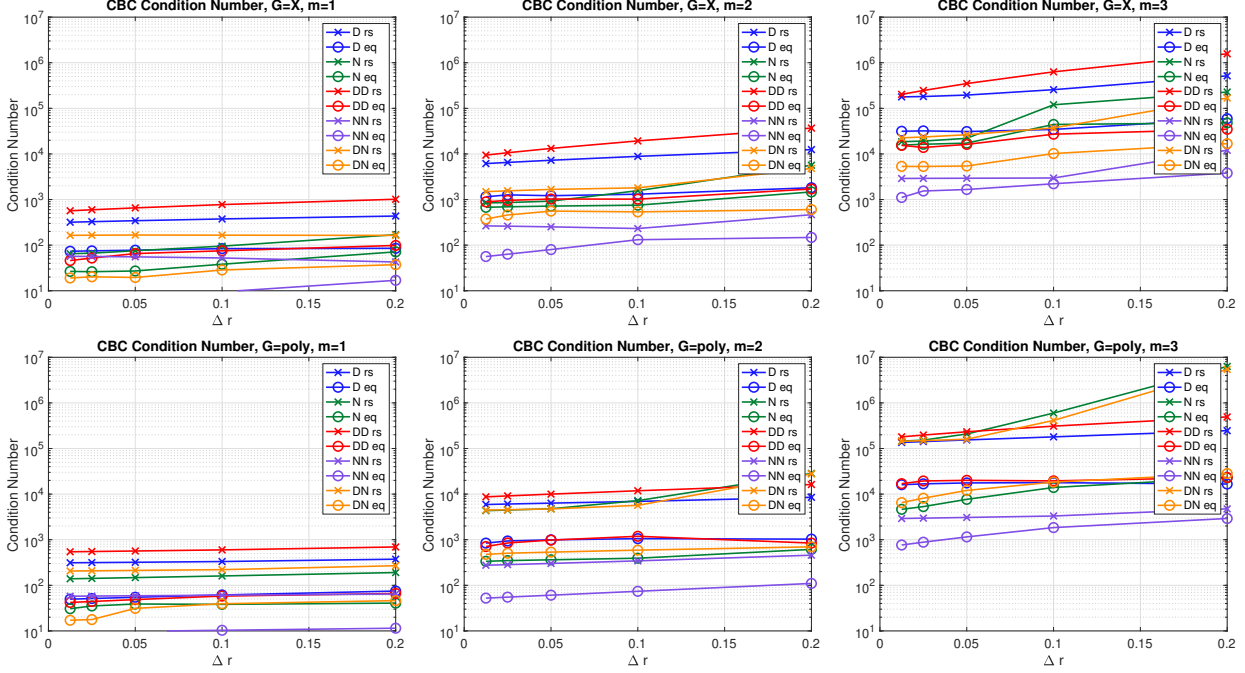


Figure 3: CBC matrix condition numbers on curvilinear grids using row-scaling (rs) and equilibration (eq) for a Dirichlet BC (D), Neumann (N) BC, D-D corner, N-N corner, and D-N corner. Top: polynomial mapping. Bottom: X mapping. Left column: $m = 1$. Middle column: $m = 2$. Right column: $m = 3$.

5.3. Symmetry properties of the CBC conditions

Consider the case of a Cartesian grid with homogeneous Dirichlet or Neumann boundary conditions. Let us focus on the boundary at $x = 0$, the result for other boundaries will be similar. The CBCs are

$$\Delta^n \partial_x u(0, y) = [\partial_x^2 + \partial_y^2]^n \partial_x^\alpha u(0, y) = 0, \quad n = 0, 1, 2, \dots, \quad (64)$$

where $\alpha = 0$ for Dirichlet boundary conditions and $\alpha = 1$ for Neumann. It follows from the binomial expansion that

$$\partial_x^{2n} \partial_x^\alpha u(0, y) = - \sum_{j=1}^n \binom{n}{j} (\partial_x^2)^{n-j} (\partial_y^2)^j \partial_x^\alpha u(0, y). \quad (65)$$

For $n = 0$ we have

$$\partial_x^\alpha u(0, y) = 0, \quad (66)$$

which implies $\partial_y^\beta \partial_x^\alpha u(0, y) = 0$ for $\beta = 0, 1, 2, \dots$. Using this in the right-hand-side of (65) for $n = 1$ gives

$$\partial_x^2 \partial_x^\alpha(0, y) = 0, \quad (67)$$

which in turn implies $\partial_y^\beta \partial_x^2 \partial_x^\alpha(0, y) = 0$ for $\beta = 0, 1, 2, \dots$. This can be used in (65) for $n = 2$ to show $\partial_x^4 \partial_x^\alpha(0, y) = 0$. Repeating this argument leads to

$$\partial_x^{2n} \partial_x^\alpha(0, y) = 0, \quad n = 0, 1, 2, \dots \quad (68)$$

On a Dirichlet boundary with $u(0, y, t) = 0$ it then follows that all even x -derivatives of u are zero on the boundary,

$$\partial_x^{2n} u(0, y, t) = 0, \quad n = 0, 1, 2, \dots \quad (69)$$

This implies that u has odd symmetry in x at the boundary. On a Neumann boundary with $\partial_x u(0, y, t) = 0$ it follows that all odd x -derivatives of u are zero

$$\partial_x^{2n+1} u(0, y, t) = 0, \quad n = 0, 1, 2, \dots, \quad (70)$$

and the solution has even symmetry in x at the boundary. The conditions (69) and (70) are often used as a simple way to set numerical boundary conditions by odd or even reflection.

The next Theorem shows that the CBC approach leads to a Taylor polynomial representation that has these same symmetry conditions.

Theorem 3 (Symmetry of the CBC conditions). *The Taylor polynomial representation of the solution, resulting from application of the CBC conditions on the boundary at $x = 0$ of a Cartesian grid, has odd symmetry for homogeneous Dirichlet boundary conditions and even symmetry for homogeneous Neumann conditions.*

Proof. The Taylor polynomial representation on the boundary $x_i = 0$ ($i = 0$) is

$$\bar{u}_{i,j}(\mathbf{x}) = \sum_{l_1=0}^{2m+1} \sum_{l_2=0}^{2m+1} \bar{u}_{l_1, l_2, i, j} X_i^{l_1} Y_j^{l_2}, \quad X_i \stackrel{\text{def}}{=} \frac{x}{\Delta x}, \quad Y_j \stackrel{\text{def}}{=} \frac{y - y_j}{\Delta y}. \quad (71)$$

For a homogeneous Dirichlet boundary condition the CBC conditions that are imposed are

$$\partial_y^\alpha \Delta^q u(0, y_j) = 0, \quad q = 0, 1, 2, \dots, m, \quad \alpha = 0, 1, 2, \dots, 2m + 1. \quad (72)$$

Following the argument leading to (69) these conditions imply

$$\partial_y^\alpha \partial_x^{2q} u(0, y_j) = 0, \quad q = 0, 1, 2, \dots, m, \quad \alpha = 0, 1, 2, \dots, 2m + 1. \quad (73)$$

Whence

$$\bar{u}_{l_1, l_2, i, j} = 0, \quad l_1 = 0, 2, 4, \dots, 2m, \quad l_2 = 0, 1, 2, \dots, 2m + 1. \quad (74)$$

Therefore only odd powers of x remain in the Taylor polynomial, which gives the desired result. For example, at $y = y_j$, the polynomial takes the form

$$\bar{u}_{i,j}(x, y_j) = \bar{u}_{1,0,i,j} x + \bar{u}_{3,0,i,j} x^3 + \dots + \bar{u}_{2m+1,0,i,j} x^{2m+1}. \quad (75)$$

The result for Neumann boundary conditions follows by a similar argument. \square

6. Numerical results

Numerical results are now presented to demonstrate the accuracy and stability of the Hermite schemes on curvilinear grids with compatibility boundary conditions. Results are shown for both orthogonal and non-orthogonal grids using Dirichlet and Neumann boundary conditions.

6.1. Mappings and Grids

Plots of the grids used to evaluate the Hermite schemes are shown in Figure 4. Two of the grids, the rhombus and X mapping, are non-orthogonal. The grids are defined in terms of mappings as defined next.

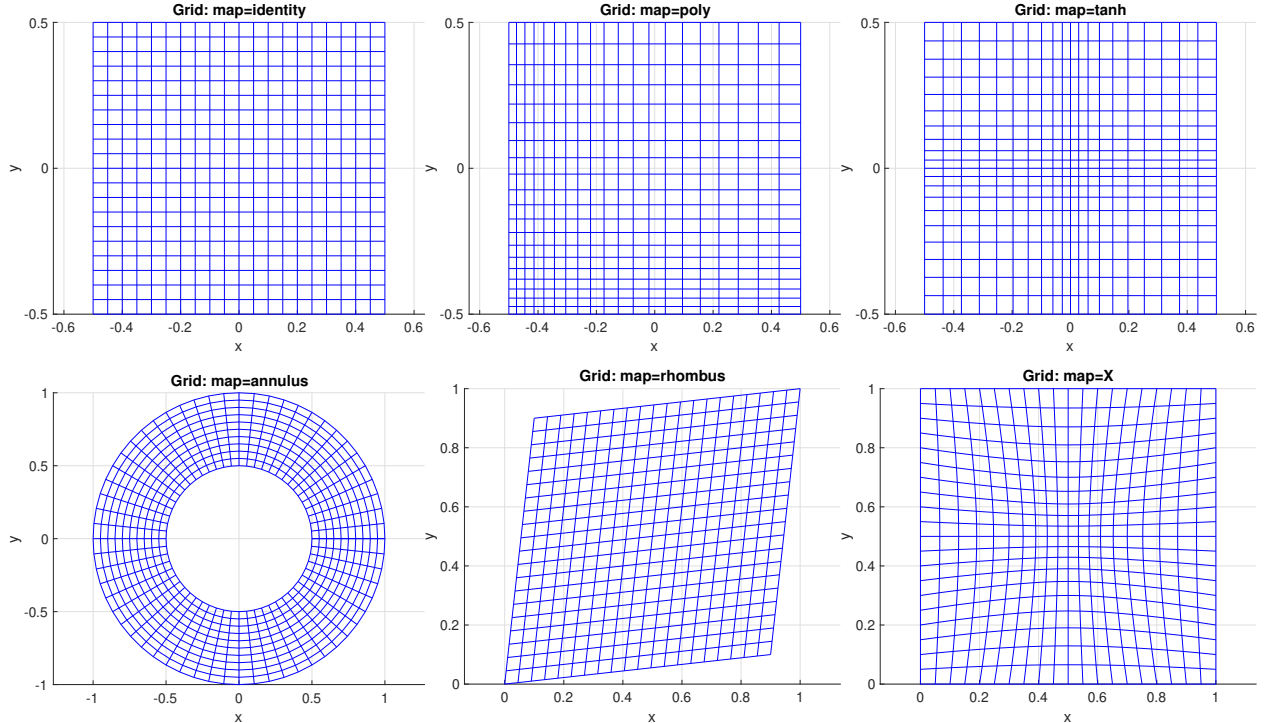


Figure 4: Plots of the grids used in testing the Hermite schemes. Top, left to right: identity, polynomial and tanh grids. Bottom, left to right: annulus, rhombus and X grids.

Polynomial mapping. The *polynomial* mapping can be used to cluster points near a boundary. In one-dimension it takes the form

$$x = G(r) = x_a + (x_b - x_a) \left(\alpha r + (1 - \alpha) r^2 \right), \quad (76)$$

and maps $r \in [0, 1]$ to $x \in [x_a, x_b]$. We choose $\alpha = 0.5$. This mapping is applied in both the r_1 and r_2 directions to give the polynomial grid in Figure 4.

Hyperbolic tangent mapping. The *tanh* mapping can be used to cluster points in the interior of the domain. In one dimension it is defined by

$$x = G(r) = x_a + (x_b - x_a) \left(\alpha r + a (\tanh(\beta(r - r_0)) - \tanh(\beta(-r_0))) \right), \quad (77a)$$

$$\alpha = 1 - a \left(\tanh(\beta(1 - r_0)) - \tanh(\beta(-r_0)) \right), \quad (77b)$$

where a is an amplitude and α is chosen so $x(1) = x_b$. We take $r_0 = 0.5$, $\beta = 5$, and $a = -0.15$.

Rhombus mapping. The rhombus mapping is a simple non-orthogonal mapping defined by

$$\mathbf{x} = \mathbf{G}(\mathbf{r}) = \begin{bmatrix} (1 - \alpha)r_1 + \alpha r_2 \\ (1 - \beta)r_2 + \beta r_1 \end{bmatrix}, \quad (78)$$

where we choose $\alpha = 0.1$ and $\beta = 0.1$

X mapping. The X mapping is a non-orthogonal mapping defined by

$$\mathbf{x} = \mathbf{G}(\mathbf{r}) = \begin{bmatrix} r_1 + \beta r_2(1 - r_2) \sin(2\pi r_1) \\ r_2 + \beta r_1(1 - r_1) \sin(2\pi r_2) \end{bmatrix}, \quad (79)$$

where we take $\beta = 0.2$.

6.2. Manufactured and exact solutions

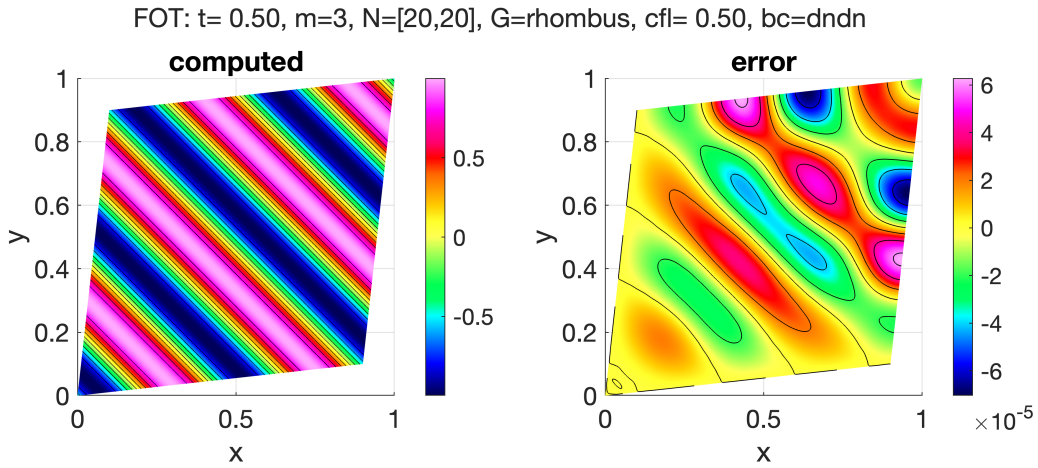


Figure 5: Rhombus. Computed solution and error using the FOT scheme with $m = 3$ (order 5) and the sine solution. Boundary conditions are Dirichlet (left, bottom) and Neumann (right and top).

Sine solution. The sine solution (shown in Figure 5) is

$$u(\mathbf{x}, t) = \sin(k_x x + k_y y - \omega t), \quad (80a)$$

$$\omega = c \sqrt{k_x^2 + k_y^2}. \quad (80b)$$

This is an exact solution to the free space problem but requires inhomogeneous boundary conditions.

Eigenfunction of a square. Eigenfunctions of the unit square with Dirichlet boundary conditions (see Figure 6) take the form

$$u(\mathbf{x}, t) = \sin(\pi k_x x) \sin(\pi k_y y) \cos(\omega t), \quad (81a)$$

for integer values of k_x and k_y , where

$$\omega = c \sqrt{(\pi k_x)^2 + (\pi k_y)^2}. \quad (81b)$$

Similar expressions can be found for Neumann boundary conditions or a mix of Dirichlet and Neumann boundary conditions.

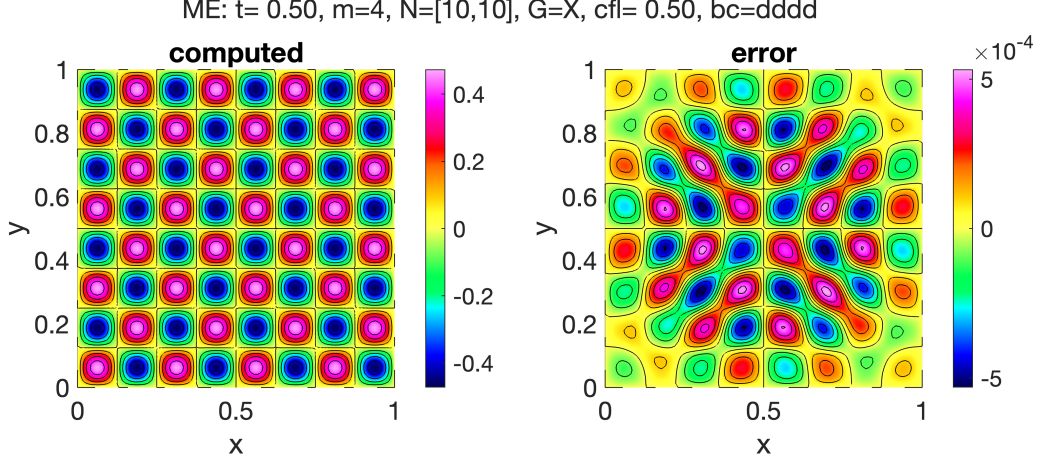


Figure 6: Square eigenfunction on the X grid. Computed solution and error using the SOT scheme with $m = 4$ (order 8).

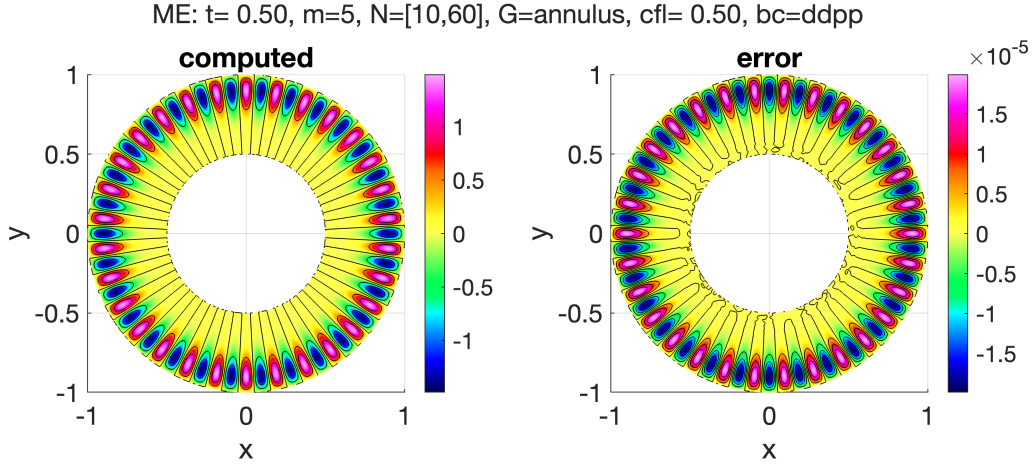


Figure 7: Annulus eigenfunction. Computed solution and error using the SOT scheme with $m = 5$ (order 10).

Eigenfunction of an annulus. Eigenfunctions of an annulus with an inner radius $r_a = 0.5$ and outer radius $r_b = 1.0$ and with Dirichlet boundary conditions, as shown in Figure 7, are of the form

$$u(r, \theta, t) = \frac{1}{\sqrt{c_J^2 + c_Y^2}} \left(c_J J_{n_\theta}(\lambda_{n_\theta, n_r} r) + c_Y Y_{n_\theta}(\lambda_{n_\theta, n_r} r) \right) \cos(n_\theta \theta) \cos(c \lambda_{n_\theta, n_r} t), \quad (82a)$$

$$c_J = Y_{n_\theta}(\lambda_{n_\theta, n_r} r_a), \quad c_Y = -J_{n_\theta}(\lambda_{n_\theta, n_r} r_a), \quad (82b)$$

where J_{n_θ} and Y_{n_θ} are the Bessel functions of the first kind. The values of the eigenvalues λ_{n_θ, n_r} are roots of $d(\lambda) = J_{n_\theta}(\lambda r_a) Y_{n_\theta}(\lambda r_b) - J_{n_\theta}(\lambda r_b) Y_{n_\theta}(\lambda r_a) = 0$.

6.3. Results for the first-order in time (FOT) scheme

Grid convergence results for the FOT scheme are shown in Figure 8 for $m = 1, 2, 3, 4$. These computations consider orthogonal and non-orthogonal grids, Dirichlet and Neumann boundary conditions (or a combination thereof to test the various treatments at corners), and both exact solutions and manufactured solutions. In particular, results are shown for the following cases,

1. Square eigenfunction, Dirichlet boundary conditions, tanh mapping,

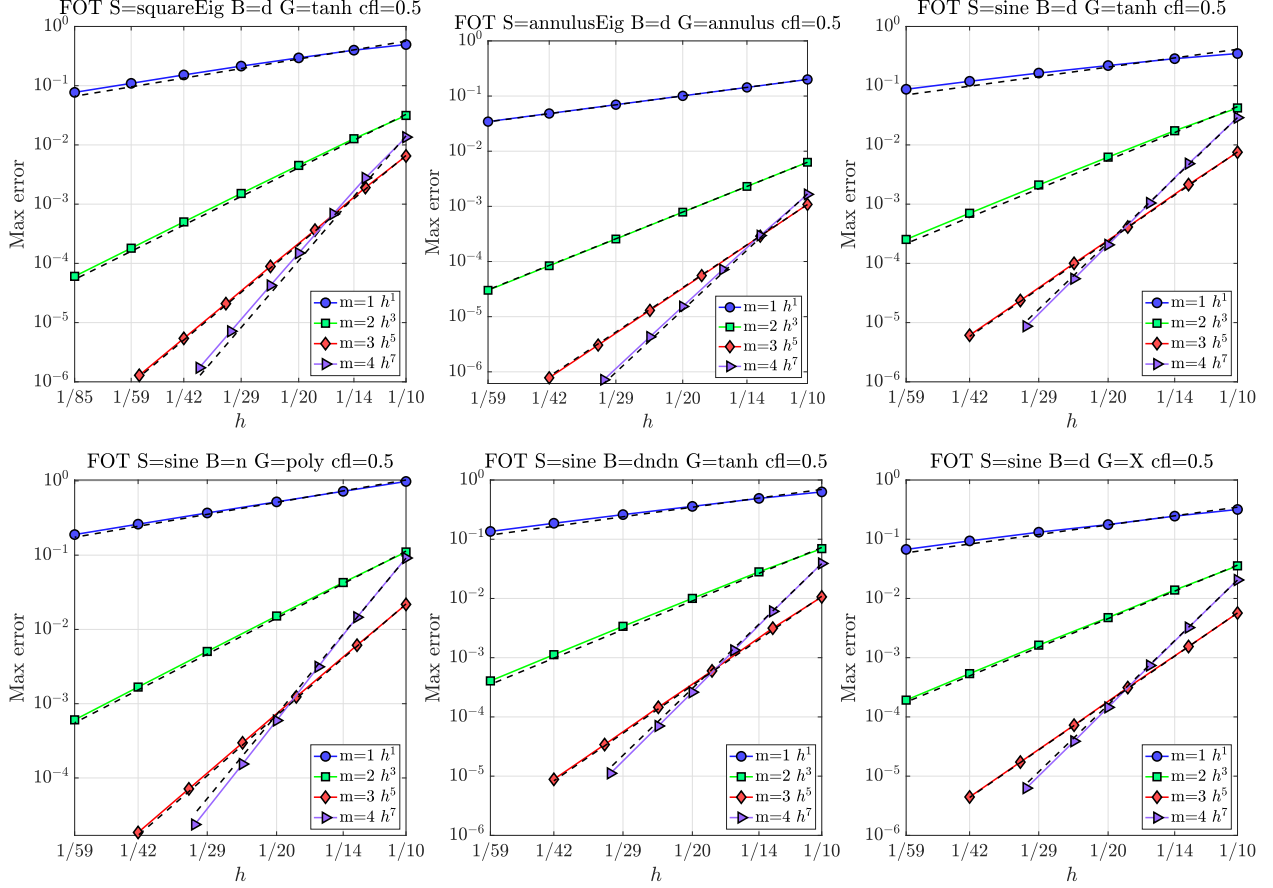


Figure 8: Grid convergence, Hermite FOT scheme in 2D. Figure titles give run details: S=solution type, B=boundary conditions, G=grid type. The expected order of accuracy for the FOT scheme is $2m - 1$ for a degree m Hermite approximation.

2. Annulus eigenfunction, Dirichlet boundary conditions, annulus mapping,
3. Sine solution, Neumann boundary conditions, polynomial mapping,
4. Sine solution, Dirichlet boundary conditions, rhombus mapping,
5. Sine solution, Dirichlet (left, bottom), Neumann (right,top), tanh mapping,
6. Sine solution, Dirichlet boundary conditions, X mapping.

The relative max-norm errors are computed at time $t = 0.5$. The wave speed c is taken as $c = 1$ in all cases. The time-step was chosen according to [Appendix B.3](#) with $C_{CFL} = 0.5$. For the square eigenfunction and sine solution we take $k_x = k_y = 2^{m-1}$ while for the Annulus eigenfunction we choose the solution with $n_\theta = 1 + 2^m - 1$, and $n_r = 2^{m-1}$. In all cases the results in [Figure 8](#) show that the expected order of accuracy of $2m - 1$ is observed.

[Figure 5](#) shows the computed solution and errors on the rhombus grid using the FOT scheme and the sine solution (80). The computed results, shown at $t = 0.5$, are computed with $m = 3$ using Dirichlet boundary conditions (left, bottom), and Neumann boundary (right,top). The error is seen to be smooth up to the boundary; this is a good indication of accuracy and quality of the CBC conditions.

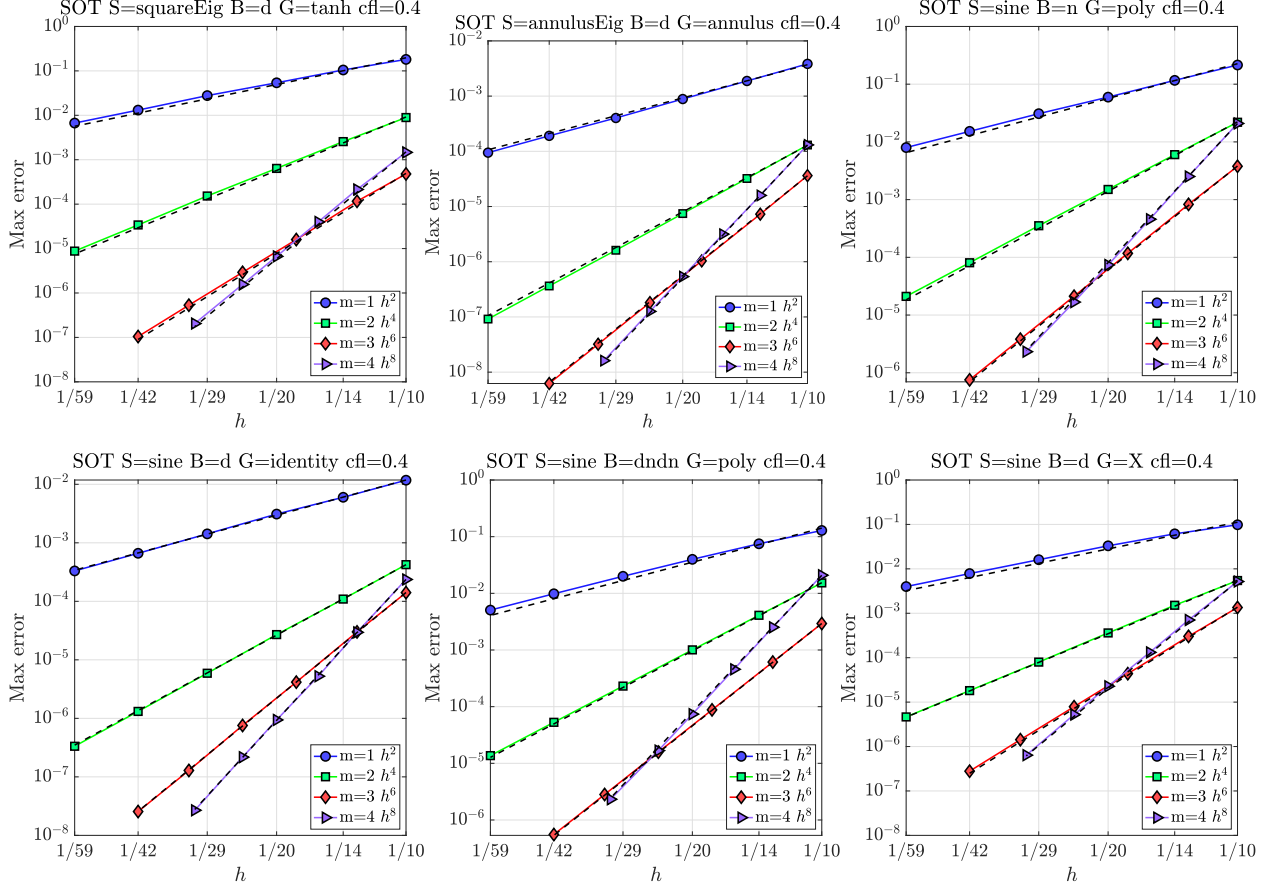


Figure 9: Grid convergence, Hermite SOT scheme in 2D. Figure titles give run details: S=solution type, B=boundary conditions, G=grid type. The expected order of accuracy for the SOT scheme is $2m$ for a degree m Hermite approximation.

6.4. Results for the second-order in time (SOT) scheme

Grid convergence results for the SOT scheme are shown in Figure 8 for $m = 1, 2, 3, 4$. Results are shown for the following cases,

1. Square eigenfunction, Dirichlet boundary conditions, tanh mapping,
2. Annulus eigenfunction, Dirichlet boundary conditions, annulus mapping, $N_s = 1$ smoothing steps (algorithm 2),
3. Sine solution, Neumann boundary conditions, polynomial mapping,
4. Sine solution, Dirichlet boundary conditions, identity mapping,
5. Sine solution, Dirichlet (left, bottom), Neumann (right, top), polynomial mapping,
6. Sine solution, Dirichlet boundary conditions, X mapping, $N_s = 1$ smoothing steps (algorithm 2).

Parameters are chosen as in section 6.3 for the FOT scheme except that here we take $C_{CFL} = 0.4$. In all cases the expected order of accuracy of $2m$ is observed.

Figure 6 shows the computation of a square eigenfunction (81) on the non-orthogonal X grid using the SOT scheme using $m = 4$ (eight-order accurate scheme). Figure 7 shows the computation on an eigenfunction of an annulus (82) using the SOT scheme with $m = 5$ (tenth order scheme). In both cases the errors are seen to be smooth up to the boundary.

6.5. Long time simulations

To numerically demonstrate the stability of the FOT and SOT schemes some long time simulations are performed. Results are shown for the following cases,

1. Sine solution, Dirichlet (left, bottom), Neumann (right,top), identity mapping, $N_1 = N_2 = 40$ grid points.
2. Sine solution, Dirichlet (inner radius), Neumann (outer radius), periodic in θ , annulus mapping, $N_s = 1$ smoothing steps (algorithm 2), $N_1 = 7$, $N_2 = 42$ grid points.
3. Sine solution, Dirichlet (left, bottom), Neumann (right,top), X mapping, $N_s = 1$ smoothing steps (algorithm 2), $N_1 = N_2 = 20$ grid points.

These cases test orthogonal and non-orthogonal mappings, Dirichlet and Neumann boundary conditions and all variations of corner conditions.

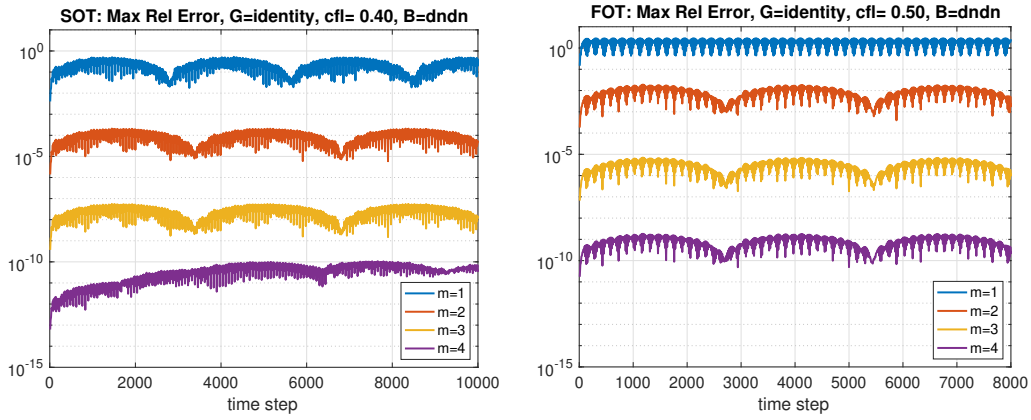


Figure 10: Max-norm errors versus time step for the identity mapping. Left: SOT scheme. Right: FOT scheme.

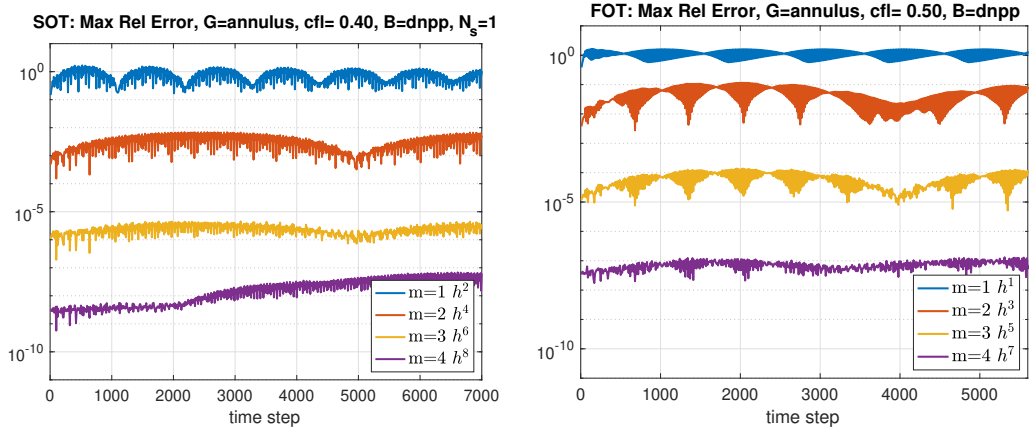


Figure 11: Max-norm errors versus time step for the Annulus mapping. Left: SOT scheme. Right: FOT scheme.

Figures 10, 11, and 12 show the relative max-norm errors in the solution as a function of time-step for the three cases given above. The relative errors are computed relative to the max-norm of the true solution which is one for the sine solution. The errors are plotted every 10 time steps. In each case the errors are seen to remain bounded with no exponential blowup that would be a sign of an instability.

For the X mapping results are not shown for $m = 4$ since for this case large errors were detected on the Neumann boundaries where the grid was most skew. This is consistent with the analysis in

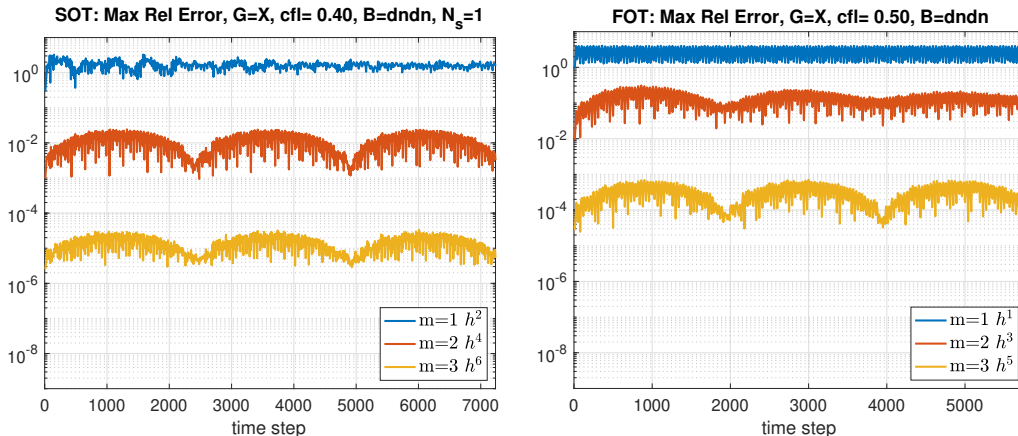


Figure 12: Max-norm errors versus time step for the X mapping. Left: SOT scheme. Right: FOT scheme.

Section 5.2, and the determinant condition (63e) for $m = 4$ which indicates that this CBC condition can be poorly conditioned on a skewed grid when the mesh spacing is not small enough. Indeed, the errors on the Neumann boundaries for $m = 4$ do decrease as the mesh is refined.

7. Conclusions

High-order accurate Hermite schemes for the wave equation on curvilinear grids have been presented. The first-order in time (FOT) schemes have accuracy $2m - 1$ for degree m Hermite schemes, while the second-order in time (SOT) schemes have accuracy $2m$. Compatibility boundary conditions (CBCs) are used to build centered polynomial approximations on the boundary. The automatic construction of the schemes for Dirichlet and Neumann boundary conditions at arbitrary order of accuracy were given. Similarly the construction of the CBC schemes at corners was provided. The solvability and conditioning of the matrices resulting from the CBC approximations were studied. With the current formulation, the conditioning of the matrices associated with the CBCs is manageable for m up to about 5 corresponding to an order of accuracy of 10 (SOT) or 9 (FOT). Iterative refinement could be used to go to larger values of m . It may also be possible to reformulate the equations to reduce the condition number. For Cartesian grids with homogeneous Dirichlet or Neumann boundary conditions it was shown that the CBCs give polynomial approximations with odd or even symmetry, respectively, and thus are equivalent to applying odd or even reflection boundary conditions. Numerical examples in two dimensions demonstrated the accuracy and stability of the schemes at different orders of accuracy and for a variety of grids, both orthogonal and non-orthogonal. Some practical considerations in implementing the Hermite schemes on curvilinear grids were provided in the Appendices.

There are several avenues of research for future work. The schemes can be extended to three space dimensions. Choosing the CBCs at vertices, where three faces meet, will need to be worked out and the conditioning of the CBC matrices in three-dimensions is an open question. The compatibility approach can be extended to treat interfaces between different material domains, where, for example the wave speeds jump. We will investigate approaches to recover the CFL-one time-step restriction on curvilinear grids, such as through the use of artificial dissipation or filters. Improving the conditioning of the CBC matrices for very high-order accurate schemes could be useful. The extension of general order Hermite schemes to unstructured grids will be considered as well as the use of the CBC approach to finite element methods. The method as presented can only handle domains that can be mapped to the unit square. Handling more complex domains would

require using an overset grid or block-structured approach. This would be an an interesting future endeavor.

Appendix A. Solvability and conditioning of the CBC matrices on Cartesian grids

In this section a proof of Theorem 1 is given. The CBC matrices M can be formed using a symbolic software package such as Maple. In the case of Cartesian grids, and m not too large, explicit forms of the max-norm condition numbers can be found. Numerical values of the condition numbers on Cartesian grids for the tall-cell ratio $\gamma = \Delta x/\Delta y = 1$ can be found in Table 1.

Dirichlet boundary. The Dirichlet CBCs (21) can be row-scaled by a factor

$$C(\alpha, q) = K \Delta y^\alpha \Delta x^{2q}, \quad (\text{A.1})$$

where K is chosen to make the maximum entry (in absolute value) in the row to be one. The resulting row-scaled equations will only depend on $\gamma = \Delta x/\Delta y$. For $m = 1$ we have an explicit formula for $\kappa_\infty(M)$,

$$\kappa_\infty(M) = \max(41, 28 + 3\gamma^2) \times \max\left(\frac{121}{16}, 1 + \gamma^2\right). \quad (\text{A.2})$$

Note that $\kappa_\infty(M)$ only depends on the tall-cell ratio γ . As is known from other computations, the CBCs become less well conditioned for large γ . For $m = 1$ and $\gamma \leq 1$, the condition number is very modest in size,

$$\kappa_\infty(M) = \frac{4961}{16} \approx 310. \quad (\text{A.3})$$

For $m = 2$

$$\begin{aligned} \kappa_\infty(M) = & \max\left(\frac{1819}{4}, 371 + \frac{25\gamma^2}{2}, \frac{597}{2} + \frac{65\gamma^2}{4}, 5 + 5\gamma^4 + 10\gamma^2, \frac{1917}{16} + \frac{21\gamma^2}{2} + \frac{15\gamma^4}{4}\right) \\ & \times \max\left(\frac{3249}{256}, 1 + \gamma^2, 1 + \frac{\gamma^2}{3} + \gamma^4\right). \end{aligned} \quad (\text{A.4})$$

For $m = 2$ and $\gamma = 1$ the condition number (with row-scaling) is

$$\kappa_\infty(M) = \frac{5909931}{1024} \approx 5770. \quad (\text{A.5})$$

For $m > 2$ the expressions become quite lengthy and instead we just report the condition numbers for $\gamma = 1$ in Table 1. As m increases the conditions numbers increase quite rapidly. Even with row-scaling, the condition number for large m is becoming quite large.

Neumann boundary. The CBCs for Neumann boundaries can also be scaled so that the resulting equations only depend on γ . For $m = 1$,

$$\kappa_\infty(M) = \max\left(\frac{125}{16}, 1 + \gamma^2, \frac{21}{4} + \frac{3\gamma^2}{4}\right) \times \max\left(\frac{121}{16}, 1 + \gamma^2\right) \quad (\text{A.6})$$

and for $\gamma \leq 1$,

$$\kappa_\infty(M) = \frac{15125}{256} \approx 310. \quad (\text{A.7})$$

For $m = 2$,

$$\begin{aligned} \kappa_\infty(M) &= \max\left(\frac{883}{16}, \frac{373}{8} + \frac{35\gamma^2}{8}, 1 + \gamma^4 + 2\gamma^2, \frac{471}{32} + \frac{21\gamma^2}{8} + \frac{15\gamma^4}{16}\right) \\ &\quad \times \max\left(\frac{3249}{256}, 1 + \gamma^2, 1 + \frac{3\gamma^2}{5} + \gamma^4\right) \end{aligned} \quad (\text{A.8})$$

and for $\gamma \leq 1$,

$$\kappa_\infty(M) \approx 700. \quad (\text{A.9})$$

For $m > 2$ and $\gamma \leq 1$, see the values in Table 1. It is seen that the condition numbers of the matrices for Neumann boundary conditions is similar to the condition numbers for Dirichlet boundary conditions.

Dirichlet-Dirichlet corner. For $m = 1$,

$$\kappa_\infty(M) = \max\left(71, \frac{95}{2} + \frac{3}{\gamma^2}, \frac{95}{2} + 3\gamma^2, \frac{485}{16} + \frac{3\gamma^2}{4} + \frac{3}{4\gamma^2}\right) \times \max\left(\frac{121}{16}, \frac{1}{3} + \frac{1}{\gamma^2}, \frac{1}{3} + \gamma^2\right), \quad (\text{A.10})$$

and for $\gamma = 1$,

$$\kappa_\infty(M) \approx 537. \quad (\text{A.11})$$

The condition number at a corner grows with γ^2 and with γ^{-2} . Thus it is advisable to have $\gamma \approx 1$ near a corner.

Neumann-Neumann corner. For $m = 1$,

$$\begin{aligned} \kappa_\infty(M) &= \max\left(\frac{121}{16}, 1 + \frac{1}{3\gamma^2}, 1 + \frac{\gamma^2}{3}, \frac{247}{64} + \frac{1}{4\gamma^2}, \frac{247}{64} + \frac{\gamma^2}{4}, \frac{585}{256} + \frac{\gamma^2}{48} + \frac{1}{48\gamma^2}\right) \\ &\quad \times \max\left(\frac{121}{16}, 1 + \frac{1}{3\gamma^2}, 1 + \frac{\gamma^2}{3}\right), \end{aligned} \quad (\text{A.12})$$

and for $\gamma = 1$,

$$\kappa_\infty(M) \approx 572. \quad (\text{A.13})$$

Dirichlet-Neumann corner. For $m = 1$,

$$\begin{aligned} \kappa_\infty(M) &= \max\left(\frac{43}{2}, 1 + \frac{1}{\gamma^2}, \frac{115}{8} + \gamma^2, \frac{227}{16} + \frac{3}{4\gamma^2}, \frac{573}{64} + \frac{\gamma^2}{4} + \frac{1}{16\gamma^2}\right) \\ &\quad \times \max\left(\frac{121}{16}, 1 + \frac{1}{\gamma^2}, 1 + \gamma^2\right), \end{aligned} \quad (\text{A.14})$$

and for $\gamma = 1$,

$$\kappa_\infty(M) \approx 163. \quad (\text{A.15})$$

Appendix B. Hermite evolution operators for the FOT and SOT schemes

The basic steps in the Hermite scheme are given in Algorithm 1. The scheme involves an Hermite interpolant \mathcal{I}_H , an evolution operator \mathcal{T}_H and a boundary condition operator \mathcal{B}_H . In this section the evolution operators \mathcal{T}_H for the FOT and SOT schemes are described.

Appendix B.1. FOT evolution

The FOT scheme stores both the solution u and the velocity v on the primal and dual grids. The degree of Taylor polynomial for v is taken as one less than that for u . Thus, the degree of $u_i(\mathbf{r})$ is m and $2m + 1$ for $\bar{u}_i(\mathbf{r})$, while the degree for $v_i(\mathbf{r})$ is $m - 1$ and $2m - 1$ for $\bar{v}_i(\mathbf{r})$. Consider the process of evolving the solution on the dual grid (line 7 in Algorithm 1) or on the primal grid (line 10 in Algorithm 1). Given \bar{u}_i^n and \bar{v}_i^n , the goal is to determine $u_i^{n+\frac{1}{2}}$ and $v_i^{n+\frac{1}{2}}$. To this end, the solution is expanded in a Taylor polynomial in space and time

$$\bar{u}_i^n(\mathbf{r}, t) = \sum_{l_1=0}^{2m+1} \sum_{l_2=0}^{2m+1} \sum_{\beta=0}^{2m+1} \bar{u}_{i,l_1,l_2,\beta} R_i^{l_1} S_j^{l_2} T_n^\beta, \quad (\text{B.1a})$$

$$\bar{v}_i^n(\mathbf{r}, t) = \sum_{l_1=0}^{2m-1} \sum_{l_2=0}^{2m-1} \sum_{\beta=0}^{2m+1} \bar{v}_{i,l_1,l_2,\beta} R_i^{l_1} S_j^{l_2} T_n^\beta, \quad (\text{B.1b})$$

$$R_i \stackrel{\text{def}}{=} \frac{r - r_i}{\Delta r}, \quad S_j \stackrel{\text{def}}{=} \frac{s - s_j}{\Delta s}, \quad T_n \stackrel{\text{def}}{=} \frac{t - t^n}{\Delta t}, \quad (\text{B.1c})$$

for some coefficients $\bar{u}_{i,l_1,l_2,\beta}$ and $\bar{v}_{i,l_1,l_2,\beta}$. The evolution equations for the FOT scheme enforce the following constraints

$$\partial_x^{\alpha_1} \partial_y^{\alpha_2} \partial_t^\beta \partial_t u = \partial_x^{\alpha_1} \partial_y^{\alpha_2} \partial_t^\beta v, \quad \alpha_k = 0, 1, \dots, 2m + 1, \quad \beta = 0, 1, \dots, 2m + 1, \quad (\text{B.2a})$$

$$\partial_x^{\alpha_1} \partial_y^{\alpha_2} \partial_t^\beta \partial_t v = \partial_x^{\alpha_1} \partial_y^{\alpha_2} \partial_t^\beta (Lu), \quad \alpha_k = 0, 1, \dots, 2m - 1, \quad \beta = 0, 1, \dots, 2m + 1, \quad (\text{B.2b})$$

at $r = r_i$, $s = s_j$, and $t = t^n$. The values of the coefficients $\bar{u}_{i,l_1,l_2,\beta}$ and $\bar{v}_{i,l_1,l_2,\beta}$ in (B.1) for $s = 0$ are determined from the Hermite interpolants for \bar{u}_i^n and \bar{v}_i^n at the current time t^n . The values of the coefficients for $s = 1, 2, \dots$ are found by a recursion derived by enforcing the conditions in (B.2). Given the coefficients $\bar{u}_{i,l_1,l_2,\beta}$ and $\bar{v}_{i,l_1,l_2,\beta}$, the coefficients in the solution at time $t^{n+\frac{1}{2}}$ are found by evaluating (B.1) at time $t^n + \Delta t/2$. This leads to the evolution function given in Algorithm 3. Recall that L_H in Algorithm 3 is the matrix representation of the operator L . In the case of a Cartesian grid, enforcing (B.2) leads to the recursions

$$\frac{\beta + 1}{\Delta t} \bar{u}_{i,l_1,l_2,\beta+1} = \bar{v}_{i,l_1,l_2,\beta}, \quad (\text{B.3a})$$

$$\frac{\beta + 1}{\Delta t} \bar{v}_{i,l_1,l_2,\beta+1} = c^2 \frac{(l_1 + 2)(l_1 + 1)}{\Delta x^2} \bar{u}_{i,l_1+2,l_2,\beta} + c^2 \frac{(l_2 + 2)(l_2 + 1)}{\Delta y^2} \bar{u}_{i,l_1,l_2+2,\beta}, \quad (\text{B.3b})$$

for $\beta = 0, 1, 2, \dots, 2m + 1$.

Appendix B.2. SOT evolution

The evolution of the SOT scheme is based on the Taylor series expansion of the second divided difference in time,

$$\frac{u(\mathbf{r}, t + \delta) - 2u(\mathbf{r}, t) + u(\mathbf{r}, t - \delta)}{\delta^2} = 2 \sum_{\mu=1}^{\infty} \frac{1}{(2\mu)!} \delta^{2\mu} \partial_t^{2\mu} u(\mathbf{r}, t). \quad (\text{B.4})$$

Using $\partial_t^{2\mu} u = L^\mu u$ and setting $\delta = \Delta t/2$ leads to

$$u(\mathbf{r}, t + \frac{\Delta t}{2}) = 2u(\mathbf{r}, t) - u(\mathbf{r}, t - \frac{\Delta t}{2}) + 2 \sum_{\mu=1}^{\infty} \frac{1}{(2\mu)!} \left[\frac{\Delta t}{2} \right]^{2\mu} L^\mu u(\mathbf{r}, t). \quad (\text{B.5})$$

Algorithm 3 FOT Evolution: advance the solution for half a time-step.

```

1: function  $[u_{\mathbf{i}}^{n+\frac{1}{2}}, v_{\mathbf{i}}^{n+1/2}] = \text{EVOLVEFOT}(\bar{u}_{\mathbf{i}}^n, \bar{v}_{\mathbf{i}}^n)$ 
2:    $\bar{u}_{\mathbf{i},l_1,l_2,0} = \bar{u}_{\mathbf{i},l_1,l_2}^n, \quad l_1, l_2 = 0, 1, 2, \dots, 2m+1$ 
3:    $\bar{v}_{\mathbf{i},l_1,l_2,0} = \bar{v}_{\mathbf{i},l_1,l_2}^n, \quad l_1, l_2 = 0, 1, 2, \dots, 2m-1$ 
4:   for  $\beta = 0, 1, \dots, 2m+1$  do
5:      $\bar{u}_{\mathbf{i},l_1,l_2,\beta+1} = \frac{\Delta t}{s+1} \bar{v}_{\mathbf{i},l_1,l_2,\beta}, \quad l_1, l_2 = 0, 1, 2, \dots, 2m+1$ 
6:      $\bar{w}_{\mathbf{i},l_1,l_2} = \bar{u}_{\mathbf{i},l_1,l_2,\beta}, \quad l_1, l_2 = 0, 1, 2, \dots, 2m+1$ 
7:      $\bar{v}_{\mathbf{i},l_1,l_2,\beta+1} = \frac{\Delta t}{s+1} (L_H \bar{w}_{\mathbf{i}})_{\mathbf{i},l_1,l_2,\beta} \quad \triangleright$  See Algorithm 8 for  $L_H \bar{w}_{\mathbf{i}}$ 
8:   end for
9:    $u_{\mathbf{i}}^{n+\frac{1}{2},l_1,l_2} = \sum_{\beta=0}^{2m+2} \bar{u}_{\mathbf{i},l_1,l_2,\beta} \left(\frac{1}{2}\right)^\beta, \quad l_1, l_2 = 0, 1, 2, \dots, m+1 \quad \triangleright$  Evaluate Taylor series in time
10:   $v_{\mathbf{i},l_1,l_2}^{n+\frac{1}{2}} = \sum_{\beta=0}^{2m+2} \bar{v}_{\mathbf{i},l_1,l_2,\beta} \left(\frac{1}{2}\right)^\beta, \quad l_1, l_2 = 0, 1, 2, \dots, m-1 \quad \triangleright$  Evaluate Taylor series in time
11: end function

```

The SOT evolution equations are derived from taking spatial derivatives of (B.5), leading to the approximations

$$\partial_{r_1}^{\alpha_1} \partial_{r_2}^{\alpha_2} u(\mathbf{r}, t^{n+\frac{1}{2}}) \approx 2 \partial_{r_1}^{\alpha_1} \partial_{r_2}^{\alpha_2} u(\mathbf{r}, t^n) - \partial_{r_1}^{\alpha_1} \partial_{r_2}^{\alpha_2} u(\mathbf{r}, t^{n-\frac{1}{2}}) + 2 \sum_{\mu=1}^m \frac{1}{(2\mu)!} \left[\frac{\Delta t}{2}\right]^{2\mu} \partial_{r_1}^{\alpha_1} \partial_{r_2}^{\alpha_2} L^\mu u(\mathbf{r}, t), \quad (\text{B.6})$$

at $\mathbf{r} = \mathbf{r}_{\mathbf{i}}$ and for $\alpha_1, \alpha_2 = 0, 1, 2, \dots, m$. Algorithm 4 gives the SOT evolution function used to compute the Hermite DOFs $u_{\mathbf{i},l_1,l_2}^{n+\frac{1}{2}}$ based on (B.6).

Algorithm 4 SOT Evolution: advance the solution a half time-step.

```

1: function  $u_{\mathbf{i}}^{n+\frac{1}{2}} = \text{EVOLVESOT}(\bar{u}_{\mathbf{i}}^n, u_{\mathbf{i}}^{n-\frac{1}{2}})$ 
2:    $u_{\mathbf{i},l_1,l_2}^{n+\frac{1}{2}} = 2 \bar{u}_{\mathbf{i},l_1,l_2}^n - u_{\mathbf{i},l_1,l_2}^{n-\frac{1}{2}}, \quad l_1, l_2 = 0, 1, 2, \dots, m$ 
3:    $\bar{w}_{\mathbf{i}} = \bar{u}_{\mathbf{i}}^n \quad \triangleright$  Holds  $L^\mu \bar{u}_{\mathbf{i}}^n$ 
4:   for  $\mu = 1, 2, \dots, m$  do
5:      $\bar{w}_{\mathbf{i}} = L_H \bar{w}_{\mathbf{i}} \quad \triangleright$  See Algorithm 8 for  $L_H \bar{w}_{\mathbf{i}}$ 
6:      $u_{\mathbf{i},l_1,l_2}^{n+\frac{1}{2}} = u_{\mathbf{i},l_1,l_2}^{n+\frac{1}{2}} + 2 \frac{1}{(2\mu)!} \left[\frac{\Delta t}{2}\right]^{2\mu} \bar{w}_{\mathbf{i},l_1,l_2}, \quad l_1, l_2 = 0, 1, 2, \dots, m$ 
7:   end for
8: end function

```

The explicit form of the update on a Cartesian grid is

$$u_{\mathbf{i},l_1,l_2}^{n+\frac{1}{2}} = 2 \bar{u}_{\mathbf{i},l_1,l_2}^n - u_{\mathbf{i},l_1,l_2}^{n-\frac{1}{2}} + \frac{2}{l_1! l_2!} \sum_{\mu=1}^m \frac{1}{(2\mu)!} \left[\frac{c\Delta t}{2}\right]^{2\mu} \sum_{j=0}^{\mu} \binom{\mu}{j} \frac{(2(\mu-j)+l_1)! (2j+l_2)!}{h_x^{2(\mu-j)} h_y^{2j}} \bar{u}_{\mathbf{i},2(\mu-j)+l_1,2j+l_2}^n, \quad (\text{B.7})$$

for $l_1, l_2 = 0, 1, \dots, m$.

Appendix B.2.1. First time-step

The SOT scheme requires two starting values, the solution on the primal grid at $t = 0$ and the solution on the dual grid at $t = -\Delta t/2$. The solution and its spatial derivatives at $t = 0$ are found from the initial condition (1b). The solution at $t = -\Delta t/2 \stackrel{\text{def}}{=} \delta$ can be found from a Taylor series in time,

$$u(\mathbf{x}, t + \delta) = u(\mathbf{x}, 0) + \delta \partial_t u(\mathbf{x}, 0) + \frac{\delta^2}{2!} \partial_t^2 u(\mathbf{x}, 0) + \frac{\delta^3}{3!} \partial_t^3 u(\mathbf{x}, 0) + \frac{\delta^4}{4!} \partial_t^4 u(\mathbf{x}, 0) + \dots \quad (\text{B.8})$$

Given initial conditions,

$$u(\mathbf{x}, 0) = U_0(\mathbf{x}), \quad (\text{B.9a})$$

$$\partial_t u(\mathbf{x}, 0) = U_1(\mathbf{x}), \quad (\text{B.9b})$$

the even time-derivatives of $u(\mathbf{x}, 0)$ are (assuming here that the body forcing $f(\mathbf{x}, t)$ is zero)

$$\partial_t^{2q} u(\mathbf{x}, 0) = L^q U_0(\mathbf{x}), \quad q = 0, 1, 2, \dots, \quad (\text{B.10})$$

where $L = c^2 \Delta$. The odd time-derivatives are

$$\partial_t^{2q+1} u(\mathbf{x}, 0) = L^q U_1(\mathbf{x}), \quad q = 0, 1, 2, \dots \quad (\text{B.11})$$

The Taylor series (B.8) can also be used to update the spatial derivatives. Algorithm 5 outlines the first (backward) step. For degree m one should keep $2m + 1$ terms in the Taylor series (the last term is the δ^{2m} term). The Algorithm 5 keeps one additional term.

Algorithm 5 First (backward) time-step for the SOT scheme

```

1: function [ $u_{\mathbf{i}}^0, u_{\mathbf{j}}^{-\frac{1}{2}}$ ] = TAKEFIRSTSTEP
2:   Set  $u_{\mathbf{i}}^0$  from  $\partial_{r_1}^{l_1} \partial_{r_2}^{l_2} U_0$ ,       $l_1, l_2 = 0, 1, \dots, m, \quad \mathbf{i} \in P$                                  $\triangleright$  Initial solution.
3:   Set  $v_{\mathbf{i}}^0$  from  $\partial_{r_1}^{l_1} \partial_{r_2}^{l_2} U_1$ ,       $l_1, l_2 = 0, 1, \dots, m, \quad \mathbf{i} \in P$                                  $\triangleright$  Initial time-derivative.
4:    $\bar{u}_{\mathbf{j}} = \mathcal{I}_H(u_{\mathbf{i}}^0)$ ,                     $\mathbf{i} \in P, \mathbf{j} \in D$                                  $\triangleright$  Interpolate to dual grid.
5:    $\bar{v}_{\mathbf{j}} = \mathcal{I}_H(v_{\mathbf{i}}^0)$ ,                     $\mathbf{i} \in P, \mathbf{j} \in D$                                  $\triangleright$  Interpolate to dual grid.
6:    $\delta = -\Delta t/2$ 
7:    $u_{\mathbf{j}, l_1, l_2}^{-\frac{1}{2}} = \bar{u}_{\mathbf{j}, l_1, l_2} + \delta \bar{v}_{\mathbf{j}, l_1, l_2}$ ,   $l_1, l_2 = 0, 1, \dots, m, \quad \mathbf{j} \in D$ 
8:   for  $k = 1, 2, \dots, m$  do
9:      $u_{\mathbf{j}, l_1, l_2}^{-\frac{1}{2}} = u_{\mathbf{j}, l_1, l_2}^{-\frac{1}{2}} + \frac{\delta^{2k}}{2k!} (L_H^k \bar{u}_{\mathbf{j}})_{\mathbf{j}, l_1, l_2} + \frac{\delta^{2k+1}}{(2k+1)!} (L_H^k \bar{v}_{\mathbf{j}})_{\mathbf{j}, l_1, l_2}$ ,   $l_1, l_2 = 0, 1, \dots, m, \quad \mathbf{j} \in D$ 
10:  end for
11: end function

```

Appendix B.3. Choosing the time-step

On Cartesian grids, the time step is chosen from

$$\frac{c \Delta t}{\min(\Delta x, \Delta y)} = C_{CFL}, \quad (\text{B.12})$$

where C_{CFL} is the CFL parameter. The SOT schemes with CBCs on Cartesian grids appear to be stable to $C_{CFL} = 1.0$ (found experimentally). The FOT scheme on Cartesian grids requires special

fixes to reach $C_{CFL} = 1.0$ as described in [6]. On curvilinear grids we estimate the smallest grid spacing in the r and s coordinate directions from the grid points,

$$\Delta x_{min} = \min_{\mathbf{i}} \left(|\mathbf{x}_{i_1+1, i_2} - \mathbf{x}_{\mathbf{i}}|, |\mathbf{x}_{i_1, i_2+1} - \mathbf{x}_{\mathbf{i}}| \right), \quad (\text{B.13})$$

and choose Δt from

$$\frac{c \Delta t}{\Delta x_{min}} = C_{CFL}. \quad (\text{B.14})$$

On curvilinear grids, the schemes generally have a lower CFL limit than for Cartesian grids (but as the mesh is refined this limit appears to approach the Cartesian grid stable CFL). For the computations in this article a choice of $C_{CFL} = 0.5$ was taken for the FOT scheme and $C_{CFL} = 0.4$ for the SOT scheme, unless otherwise specified. Numerical experiments suggest that the addition of some dissipation to either scheme will increase the stable C_{CFL} . An investigation into this behavior will be left to future work.

Appendix C. Practicalities

This section provides some helpful information for those readers interested in implementing Hermite schemes.

Appendix C.1. Hermite interpolants

In one space dimension the degree m Taylor polynomial representation of the solution is

$$u_i(r) = \sum_{l_1=0}^m u_{i, l_1} R_i^{l_1}, \quad R_i \stackrel{\text{def}}{=} \frac{r - r_i}{\Delta r}. \quad (\text{C.1})$$

The degree $2m + 1$ Hermite interpolant given by

$$\bar{u}_{i+\frac{1}{2}}(r) = \sum_{l_1=0}^{2m+1} \bar{u}_{i+\frac{1}{2}, l_1} R_i^{l_1}, \quad (\text{C.2})$$

is chosen to match the solution and its derivatives at points r_i and r_{i+1} ,

$$\partial_r^\alpha \bar{u}_{i+\frac{1}{2}}(r_i) = \partial_r^\alpha u_i(r_i), \quad \alpha = 0, 1, 2, \dots, m, \quad (\text{C.3a})$$

$$\partial_r^\alpha \bar{u}_{i+\frac{1}{2}}(r_{i+1}) = \partial_r^\alpha u_{i+1}(r_{i+1}), \quad \alpha = 0, 1, 2, \dots, m. \quad (\text{C.3b})$$

Now

$$\frac{\Delta r^\alpha}{\alpha!} \partial_r^\alpha u_i(r_i) = u_{i, \alpha}, \quad \frac{\Delta r^\alpha}{\alpha!} \partial_r^\alpha u_{i+1}(r_{i+1}) = u_{i+1, \alpha}, \quad (\text{C.4})$$

while

$$\frac{\Delta r^\alpha}{\alpha!} \partial_r^\alpha \bar{u}_{i+\frac{1}{2}}(r) = \sum_{l_1=\alpha}^{2m+1} \binom{l_1}{\alpha} \bar{u}_{i+\frac{1}{2}, l_1} R_i^{l_1-\alpha}. \quad (\text{C.5})$$

This leads to the interpolation conditions

$$\sum_{l_1=\alpha}^{2m+1} \binom{l_1}{\alpha} \bar{u}_{i+\frac{1}{2},l_1} \left[-\frac{1}{2} \right]^{l_1-\alpha} = u_{i,\alpha}, \quad (\text{C.6a})$$

$$\sum_{l_1=\alpha}^{2m+1} \binom{l_1}{\alpha} \bar{u}_{i+\frac{1}{2},l_1} \left[+\frac{1}{2} \right]^{l_1-\alpha} = u_{i+1,\alpha}. \quad (\text{C.6b})$$

for $\alpha = 0, 1, 2, \dots, m$. Equations (C.6) define a linear system of equations whose solution can be written as

$$\bar{u}_{i+\frac{1}{2}} = \mathcal{I}_H^{(1)}(u_i), \quad (\text{C.7})$$

where $\mathcal{I}_H^{(1)}$ is the Hermite interpolation operator in coordinate direction r_1 . The Hermite interpolant in two dimensions is defined by repeated application of one-dimensional interpolants, first in the r_1 -direction and then in the r_2 -direction as given in Algorithm 6.

Algorithm 6 Compute the Hermite interpolant in two dimensions, $\bar{u}_i = \mathcal{I}_H(u_i)$.

```

1: function  $\bar{u}_i = \mathcal{I}_H(u_i)$ 
2:   for  $l_2 = 0, 1, \dots, m$  do ▷ Interpolate in  $r_1$  direction
3:      $\bar{u}_{i+\frac{1}{2},i_2,0:2m+1,l_2} = \mathcal{I}_H^{(1)}(u_{i_1,i_2,0:m+1,l_2}),$   $i_1 = 0, 1, \dots, N_1 - 1, \quad i_2 = 0, 1, \dots, N_2$ 
4:   end for
5:   for  $l_1 = 0, 1, \dots, 2m + 1$  do ▷ Interpolate in  $r_2$  direction
6:      $\bar{u}_{i_1+\frac{1}{2},i_2+\frac{1}{2},l_1,0:2m+1} = \mathcal{I}_H^{(2)}(\bar{u}_{i_1+\frac{1}{2},i_2,l_1,0:m+1}),$   $i_1 = 0, 1, \dots, N_1 - 1, \quad i_2 = 0, 1, \dots, N_1 - 1$ 
7:   end for
8: end function

```

Appendix C.2. Taylor polynomial coefficients from function evaluations

Algorithm 7 contains a useful procedure that determines approximations to the scaled Taylor polynomial coefficients (as used by the Hermite schemes) of a known function $f(x)$. This function can be used for setting up initial conditions, evaluating forcing functions and their derivatives for boundary conditions (such as in (21)), as well as computing the Taylor polynomial representations of the curvilinear coefficients of L in (3).

Algorithm 7 finds approximate Taylor coefficients for a function $f(x)$ on a cell with center x_i and width Δx . The algorithm evaluates the function on a set of $q + 1$ scaled Chebyshev points on $[x_i - \Delta x/2, x_i + \Delta x/2]$, forms an interpolant in Newton divided difference form, and then converts the coefficients of this interpolant into the coefficients of a power series. This procedure is described in *Algorithm for Dual System*, in *Solution of Vandermonde Systems of Equations*, by Åke Björk and Victor Pereyra [45]. It avoids computing the interpolant from a Vandermonde system, which would be very ill conditioned. For a Hermite polynomial of degree m we choose $q = 2m + 1$ interpolation intervals.

Appendix C.3. Taylor polynomials for the Laplacian in curvilinear coordinates

The coefficients, $a^{\alpha\beta}(\mathbf{r})$, in the Laplacian in curvilinear coordinates in (3), are approximated as Taylor polynomials of degree $2m + 1$ in each direction,

$$a_i^{\alpha\beta}(\mathbf{r}) = \sum_{l_1=0}^{2m+1} \sum_{l_2=0}^{2m+1} a_{i,l_1,l_2}^{\alpha\beta} R_i^{l_1} S_j^{l_2}. \quad (\text{C.8})$$

Algorithm 7 Find scaled Taylor coefficients f_j , $j = 0, 1, \dots, q$ from function evaluations.

```

1: function f = GETTAYLORCOEFFICIENTS(  $q, x_i, \Delta x, f$  )
2:   // Evaluate the function  $f(x)$  at  $q + 1$  Chebyshev points on the cell centered at  $x_i$  with width  $\Delta x$ 
3:   for  $j = 0, 1, 2, \dots, q$  do
4:      $z_j = -\frac{1}{2} \cos(\pi j/q)$  ▷ Chebyshev points on  $[-\frac{1}{2}, \frac{1}{2}]$ 
5:      $f_j = f(x_i + \Delta x z_j)$  ▷ Evaluate  $f$  on scaled Chebyshev points on  $[x_i - \Delta x/2, x_i + \Delta x/2]$ 
6:   end for
7:   for  $k = 1, 2, \dots, q$  do
8:     for  $j = q, q-1, \dots, k$  do
9:        $f_j = \frac{f_j - f_{j-1}}{z_j - z_{j-k}}$  ▷ Newton divided differences
10:    end for
11:  end for
12:  for  $k = q-1, q-2, \dots, 0$  do
13:    for  $j = k, k+1, \dots, q-1$  do
14:       $f_j = f_j - z_k f_{j+1}$  ▷ Recursion to convert Newton form to Taylor form
15:    end for
16:  end for
17: end function

```

Values for $a_{\mathbf{i}, l_1, l_2}^{\alpha\beta}$ can be conveniently computed making use of the approach outlined in [Appendix C.2](#) which requires a function to evaluate $a_{\mathbf{i}}^{\alpha\beta}(\mathbf{r})$ for different values of \mathbf{r} . Consider, for example, computing the Taylor polynomial approximation to $a^{20}(\mathbf{r})$,

$$a^{20}(\mathbf{r}) = (\partial_{x_1} r_1)^2 + (\partial_{x_2} r_1)^2. \quad (\text{C.9})$$

Let us suppose that we have a function to compute entries in the Jacobian matrix⁵

$$\frac{\partial \mathbf{x}}{\partial \mathbf{r}} = \frac{\partial \mathbf{G}(\mathbf{r})}{\partial \mathbf{r}} = \left[\frac{\partial x_\mu}{\partial r_\nu} \right]. \quad (\text{C.10})$$

The inverse metrics can be found from the inverse of the Jacobian matrix

$$\frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \left[\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right]^{-1} = \left[\frac{\partial r_\mu}{\partial x_\nu} \right]. \quad (\text{C.11})$$

Now fit Taylor polynomials to the inverse metrics

$$\left[\frac{\partial r_\mu}{\partial x_\nu} \right]_{\mathbf{i}}(\mathbf{r}) = \sum_{l_1=0}^{2m+1} \sum_{l_2=0}^{2m+1} \left[\frac{\partial r_\mu}{\partial x_\nu} \right]_{\mathbf{i}, l_1, l_2} R_i^{l_1} S_j^{l_2}. \quad (\text{C.12})$$

The coefficients in the Taylor polynomial for $a^{20}(\mathbf{r})$ in (C.9) can then be computed from the polynomials for the inverse metrics through multiplication and addition of polynomials, truncated to degree $2m + 1$ in each direction. The coefficients a^{10} and a^{01} in (3) depend on derivatives of the inverse metrics and these can be computed using the chain rule and taking derivatives of the Taylor polynomials.

⁵If the metric derivatives are not known then one can work directly with $\mathbf{G}(\mathbf{r})$ and take derivatives of the Taylor polynomials to get the entries in the Jacobian matrix.

Appendix C.4. Applying the wave operator in curvilinear coordinates

Given the Taylor polynomial representations for the curvilinear coefficients $a_i^{\mu,\nu}(\mathbf{r})$ applying the wave operator L to a Hermite representation $\bar{u}_i(\mathbf{r})$ is straightforward. For example, consider the computation of $\bar{w} = a^{20} \partial_r^2 \bar{u}$. We have

$$\partial_r^2 \bar{u}_i(\mathbf{r}) = \sum_{l_1=2}^{2m+1} \sum_{l_2=0}^{2m+1} \bar{u}_{i,l_1,l_2} \frac{l_1(l_1-1)}{\Delta r^2} R_i^{l_1-2} S_j^{l_2}. \quad (\text{C.13})$$

Then

$$a_i^{20}(\mathbf{r}) \partial_r^2 \bar{u}_i(\mathbf{r}) = \sum_{l'_1=0}^{2m+1} \sum_{l'_2=0}^{2m+1} a_{i,l'_1,l'_2}^{20} R_i^{l'_1} S_j^{l'_2} \times \sum_{l''_1=2}^{2m+1} \sum_{l''_2=0}^{2m+1} \bar{u}_{i,l''_1,l''_2} \frac{l''_1(l''_1-1)}{\Delta r^2} R_i^{l''_1-2} S_j^{l''_2}, \quad (\text{C.14a})$$

$$= \sum_{l'_1=0}^{2m+1} \sum_{l'_2=0}^{2m+1} \sum_{l''_1=2}^{2m+1} \sum_{l''_2=0}^{2m+1} a_{i,l'_1,l'_2}^{20} \bar{u}_{i,l''_1,l''_2} \frac{l''_1(l''_1-1)}{\Delta r^2} R_i^{l'_1+l''_1-2} S_j^{l'_2+l''_2} \quad (\text{C.14b})$$

Setting $l'_1 + l''_1 - 2 = l_1$ and $l'_2 + l''_2 = l_2$ and $l'_1 = k_1$ and $l'_2 = k_2$ implies

$$l''_1 = l_1 - k_1 + 2, \quad l''_2 = l_2 - k_2 \quad (\text{C.15})$$

and thus

$$a_i^{20}(\mathbf{r}) \partial_r^2 \bar{u}_i(\mathbf{r}) = \sum_{l_1=0}^{2m+1} \sum_{l_2=0}^{2m+1} \bar{w}_{i,l_1,l_2} R_i^{l_1} S_j^{l_2}, \quad (\text{C.16a})$$

$$\bar{w}_{i,l_1,l_2} = \sum_{k_1=0}^{l_1} \sum_{k_2=0}^{l_2} a_{i,k_1,k_2}^{\alpha\beta} \bar{u}_{i,l_1-k_1+2,l_2-k_2} \frac{(l_1 - k_1 + 2)(l_1 - k_1 + 1)}{\Delta r^2} \quad (\text{C.16b})$$

Note that the polynomials in (C.16) have been truncated to degree $2m+1$ and that any terms with an array index out of bounds should be ignored. Algorithm 8 gives the full algorithm to apply L (with the same caveat that terms with invalid subscripts should be ignored.)

References

- [1] N. G. A. Hassanieh, J. W. Banks, W. D. Henshaw, D. W. Schwendeman, Local Compatibility Boundary Conditions for High-Order Accurate Finite-Difference Approximations of PDEs, *SIAM Journal on Scientific Computing* 44 (6) (2022) A3645–A3672.
- [2] J. Goodrich, T. Hagstrom, J. Lorenz, Hermite methods for hyperbolic initial-boundary value problems, *Mathematics of computation* 75 (254) (2006) 595–630.
- [3] D. Appelo, T. Hagstrom, A. Vargas, Hermite methods for the scalar wave equation, *SIAM Journal on Scientific Computing* 40 (6) (2018) A3902–A3927.
- [4] R. Chen, T. Hagstrom, P-adaptive Hermite methods for initial value problems, *ESAIM: Mathematical Modelling and Numerical Analysis* 46 (3) (2012) 545–557.
- [5] A. Kornelus, D. Appelo, Flux-conservative Hermite methods for simulation of nonlinear conservation laws, *Journal of Scientific Computing* 76 (1) (2018) 24–47.

Algorithm 8 Evaluate $\bar{w}_i = L_H \bar{u}_i$, for L in curvilinear coordinates.

```

1: function  $\bar{w}_i = \text{APPLYOPERATOR}(\bar{u}_i)$ 
2:   //  $L = c^2 \Delta = a^{20}(\mathbf{r}) \frac{\partial^2}{\partial r_1^2} + a^{11}(\mathbf{r}) \frac{\partial^2}{\partial r_1 \partial r_2} + a^{02}(\mathbf{r}) \frac{\partial^2}{\partial r_2^2} + a^{10}(\mathbf{r}) \frac{\partial}{\partial r_1} + a^{01}(\mathbf{r}) \frac{\partial}{\partial r_2}$ 
3:   for  $l_2 = 0, 1, \dots, 2m + 1$  do
4:     for  $l_1 = 0, 1, \dots, 2m + 1$  do
5:        $\bar{w}_{i,l_1,l_2} = 0$ 
6:       for  $k_2 = 0, 1, \dots, l_1$  do
7:         for  $k_1 = 0, 1, \dots, l_2$  do
8:            $\bar{w}_{i,l_1,l_2} = \bar{w}_{i,l_1,l_2}$ 
9:              $+ a_{i,k_1,k_2}^{20} \bar{u}_{i,l_1-k_1+2,l_2-k_2} \frac{(l_1 - k_1 + 2)(l_1 - k_1 + 1)}{\Delta r^2}$ 
10:             $+ a_{i,k_1,k_2}^{11} \bar{u}_{i,l_1-k_1+1,l_2-k_2+1} \frac{(l_1 - k_1 + 1)(l_2 - k_2 + 1)}{\Delta r \Delta s}$ 
11:             $+ a_{i,k_1,k_2}^{02} \bar{u}_{i,l_1-k_1,l_2-k_2+2} \frac{(l_2 - k_2 + 2)(l_2 - k_2 + 1)}{\Delta s^2}$ 
12:             $+ a_{i,k_1,k_2}^{10} \bar{u}_{i,l_1-k_1+1,l_2-k_2} \frac{(l_1 - k_1 + 1)}{\Delta r} + a_{i,k_1,k_2}^{01} \bar{u}_{i,l_1-k_1,l_2-k_2+1} \frac{(l_2 - k_2 + 1)}{\Delta s}$ 
13:          end for
14:        end for
15:      end for
16:    end for
17: end function

```

- [6] A. Alvarez Loya, D. Appelö, A Hermite method with a discontinuity sensor for Hamilton–Jacobi equations, *Journal of Scientific Computing* 90 (3) (2022) 1–31.
- [7] O. Beznosov, D. Appelö, Hermite-discontinuous Galerkin overset grid methods for the scalar wave equation, *Communications on Applied Mathematics and Computation* 3 (3) (2021) 391–418.
- [8] X. R. Chen, D. Appelö, T. Hagstrom, A hybrid Hermite–discontinuous Galerkin method for hyperbolic systems with application to Maxwell’s equations, *Journal of Computational Physics* 257 (2014) 501–520.
- [9] D. Appelö, T. Hagstrom, On advection by Hermite methods, *Pacific Journal Of Applied Mathematics* 4 (2) (2011) 125–139.
- [10] On sub-linear convergence for linearly degenerate waves in capturing schemes, *Journal of Computational Physics* 227 (14) (2008) 6985–7002.
- [11] W. D. Henshaw, H.-O. Kreiss, L. G. Reyna, A fourth-order-accurate difference approximation for the incompressible Navier-Stokes equations, *Computers & fluids* 23 (4) (1994) 575–593.
- [12] W. D. Henshaw, A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids, *Journal of computational physics* 113 (1) (1994) 13–25.
- [13] F. Meng, J. Banks, W. Henshaw, D. Schwendeman, Fourth-order accurate fractional-step IMEX schemes for the incompressible Navier–Stokes equations on moving overlapping grids, *Computer Methods in Applied Mechanics and Engineering* 366 (2020) 113040.
- [14] W. D. Henshaw, D. W. Schwendeman, Moving overlapping grids with adaptive mesh refinement for high-speed reactive and non-reactive flow, *Journal of Computational Physics* 216 (2) (2006) 744–779.

- [15] D. Appelö, J. W. Banks, W. D. Henshaw, D. W. Schwendeman, Numerical methods for solid mechanics on overlapping grids: Linear elasticity, *Journal of Computational Physics* 231 (18) (2012) 6012–6050.
- [16] J. B. Angel, J. W. Banks, W. D. Henshaw, M. J. Jenkinson, A. V. Kildishev, G. Kovačič, L. J. Prokopeva, D. W. Schwendeman, A high-order accurate scheme for Maxwell’s equations with a generalized dispersive material model, *Journal of Computational Physics* 378 (2019) 411–444.
- [17] S. Tan, C.-W. Shu, Inverse lax-wendroff procedure for numerical boundary conditions of conservation laws, *Journal of Computational Physics* 229 (21) (2010) 8144–8166.
- [18] R. B. d. R. Borges, N. D. P. daSilva, F. A. A. Gomes, C.-W. Shu, S. Tan, A Sequel of Inverse Lax–Wendroff High Order Wall Boundary Treatment for Conservation Laws, *Archives of Computational Methods in Engineering*.
- [19] J. Lu, C.-W. Shu, S. Tan, M. Zhang, An inverse Lax-Wendroff procedure for hyperbolic conservation laws with changing wind direction on the boundary, *Journal of Computational Physics* 426 (2021) 109940.
- [20] C.-W. Shu, S. Tan, Chapter 2 - inverse Lax–Wendroff procedure for numerical boundary treatment of hyperbolic equations, in: R. Abgrall, C.-W. Shu (Eds.), *Handbook of Numerical Methods for Hyperbolic Problems*, Vol. 18 of *Handbook of Numerical Analysis*, Elsevier, 2017, pp. 23–52.
- [21] T. Li, C.-W. Shu, M. Zhang, Stability Analysis of the Inverse Lax–Wendroff Boundary Treatment for High Order Central Difference Schemes for Diffusion Equations, *Journal of Scientific Computing* 70 (2) (2017) 576–607.
- [22] J. Lu, J. Fang, S. Tan, C.-W. Shu, M. Zhang, Inverse Lax–Wendroff procedure for numerical boundary conditions of convection–diffusion equations, *Journal of Computational Physics* 317 (2016) 276–300.
- [23] B. Gustafsson, *High Order Difference Methods for Time Dependent PDE*, Springer Series in Computational Mathematics, Springer, 2008.
- [24] R. J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM Journal on Numerical Analysis* 31 (4) (1994) 1019–1044.
- [25] R. J. LeVeque, Z. Li, Immersed interface methods for stokes flow with elastic boundaries or surface tension, *SIAM Journal on Scientific Computing* 18 (3) (1997) 709–735.
- [26] R. J. LeVeque, D. Calhoun, Cartesian grid methods for fluid flow in complex geometries, in: L. J. Fauci, S. Gueron (Eds.), *Computational Modeling in Biological Fluid Dynamics*, Vol. 124 of *IMA Volumes in Mathematics and its Applications*, Springer-Verlag, 2001, pp. 117–143.
- [27] H.-O. Kreiss, N. A. Petersson, A second order accurate embedded boundary method for the wave equation with Dirichlet data, *SIAM Journal on Scientific Computing* 27 (4) (2006) 1141–1167.

- [28] S. Nilsson, N. A. Petersson, B. Sjögreen, H.-O. Kreiss, Stable difference approximations for the elastic wave equation in second order formulation, *SIAM Journal on Numerical Analysis* 45 (5) (2007) 1902–1936.
- [29] D. Appelö, N. A. Petersson, A fourth-order accurate embedded boundary method for the wave equation, *SIAM Journal on Scientific Computing* 34 (6) (2012) A2982–A3008.
- [30] B. Sjögreen, N. A. Petersson, A fourth order finite difference scheme for the elastic wave equation in second order formulation, *Scient. Comput.* 52 (2012) 17–48.
- [31] S. Petropavlovsky, S. V. Tsynkov, E. Turkel, Numerical solution of 3d exterior unsteady wave propagation problems using boundary operators, *SIAM Journal on Scientific Computing* 42 (5) (2020) A3462–A3488.
- [32] K. Mattsson, J. Nordström, Summation by parts operators for finite difference approximations of second derivatives, *J. Comput. Phys.* 199 (2004) 503–540.
- [33] J. Banks, W. Henshaw, Upwind schemes for the wave equation in second-order form, *Journal of Computational Physics* 231 (17) (2012) 5854–5889.
- [34] J. Banks, W. H. , High-order upwind methods for wave equations on curvilinear and overlapping grids, in: R. M. Kirby, M. Berzins, J. S. Hesthaven (Eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014: Selected papers from the ICOSAHOM conference, June 23-27, 2014, Salt Lake City, Utah, USA, 2015*, pp. 137–145.
- [35] P. Joly, Variational methods for time-dependent wave propagation problems., Vol. 31 of *Topics in computational wave propagation*, Springer, Berlin, 2003, pp. 201–264.
- [36] D. S. Marcus J. Grote, Anna Schneebeli, Discontinuous Galerkin finite element method for the wave equation, *SIAM Journal on Numerical Analysis* 44 (6) (2006) 2408–2431.
- [37] B. Riviere, M. Wheeler, Discontinuous finite element methods for acoustic and elastic wave problems. Part i: semidiscrete error estimates, *Contemporary Mathematics* 329 (2003) 271–282.
- [38] D. Appelö, T. Hagstrom, A new discontinuous Galerkin formulation for wave equations in second order form, *SIAM Journal On Numerical Analysis* 53 (6) (2015) 2705–2726.
- [39] C.-S. Chou, C.-W. Shu, Y. Xing, Optimal energy conserving local discontinuous Galerkin methods for second-order wave equation in heterogeneous media, *Journal of Computational Physics* 272 (2014) 88 – 107.
- [40] O. Bruno, M. Lyon, High-order unconditionally-stable FC-AD solvers for general smooth domains I. basic elements., Submitted.
- [41] M. Lyon, O. Bruno, High-order unconditionally-stable FC-AD solvers for general smooth domains II. elliptic, parabolic and hyperbolic PDEs: Theoretical considerations, Submitted.
- [42] J. Banks, T. Hagstrom, On Galerkin difference methods, *J. Comput. Phys.* 313 (2016) 310–327.
- [43] J. Scott, et al., Hsl@60: a brief history of the HSL mathematical software library, Tech. rep., STFC (2023).

- [44] D. Appelö, M. Inkman, T. Hagstrom, T. Colonius, Hermite methods for aeroacoustics: Recent progress, in: 17th AIAA/CEAS Aeroacoustics Conference (32nd AIAA Aeroacoustics Conference), Portland, Oregon, June 5-8, 2011, no. AIAA-2011-2757, 2011.
- [45] Å. Björck, V. Pereyra, Solution of Vandermonde systems of equations, *Mathematics of Computation* 24 (1970) 893–903.