Journal of Computational Physics 426 (2021) 109794

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Foundations for high-order, conservative cut-cell methods: Stable discretizations on degenerate meshes

P.T. Brady *, D. Livescu

CCS-2, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:
Received 16 March 2020
Received in revised form 14 August 2020
Accepted 17 August 2020
Available online 14 October 2020

Keywords:
Cut-cell
High-order
Conservative
Stable
Truncation error matching
Optimization

ABSTRACT

Cut-cell methods for unsteady flow problems can greatly simplify the grid generation process and allow for high-fidelity simulations on complex geometries. However, cut-cell methods have been limited to low orders of accuracy. This is driven, largely, by the variety of procedures typically introduced to evaluate derivatives in a stable manner near the highly irregular embedded geometry. In the present work, a completely new approach, termed TEMO (truncation error matching and optimization), is taken to solve this problem. The approach is based on two simple and intuitive design principles. These principles directly allow for the construction of stable 8th order approximations to elliptic and parabolic problems. In addition, when combined with the non-linear optimization process of Ref. [6], these principles allow for stable and conservative 4th order approximations to hyperbolic problems without the addition of numerical dissipation. To the best of the authors' knowledge, these are the highest orders ever achieved for a cut-cell discretization by a significant margin. This is done for both explicit and compact finite differences and is accomplished without any geometric transformations or artificial stabilization procedures.
© 2020 Elsevier Inc. All rights reserved.

1. Introduction

The cut-cell method [17] allows for the solution of partial differential equations (PDEs) defined on complicated domains to be computed numerically on simple Cartesian meshes. This method has seen extensive use in the fluids community, so we define the domain of interest, Ω_f, as the fluid domain which is bounded by $\Gamma_f \cup \Gamma_s$, where the Cartesian and solid object boundaries are given by Γ_f and Γ_s, respectively. A schematic of this is shown in Fig. 1. Thus, the non-Cartesian physical boundaries are embedded into the simpler Cartesian mesh, leading to computational cells which have been cut by the embedded object. Rather than modifying the physical equations to implicitly account for this object, the cut-cell approach modifies the discrete operators and imposes boundary conditions directly on Γ_s.

The allure of cut-cell type methods has attracted the attention and effort of a number of researchers for many years (see [26] for a review). In theory, cut-cell methods obviate the need for unstructured meshes and allow for the use of robust, accurate and conservative finite difference/volume schemes with only slight modifications near the boundary. However, the current solutions to the severe numerical challenges of cut-cell schemes typically lead to significant modifications of the discrete algorithm. The discrete algorithms are modified by requiring extra procedures to evaluate derivatives near the boundary since a straightforward evaluation leads to instabilities. The complexity of the correction procedures and the errors incurred by them have made it difficult to achieve high order accurate cut-cell methods.

* Corresponding author.
E-mail address: ptb@lanl.gov (P.T. Brady).

https://doi.org/10.1016/j.jcp.2020.109794
0021-9991/© 2020 Elsevier Inc. All rights reserved.

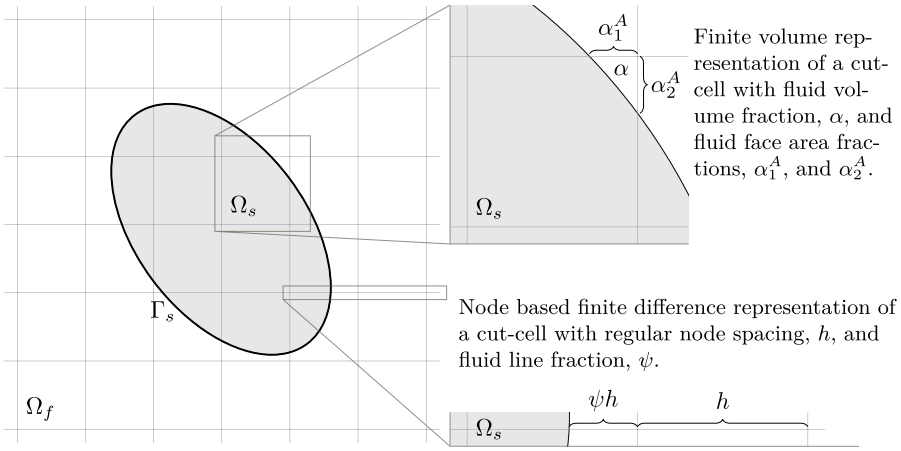

Fig. 1. Schematic of solid object, Ω_s, bounded by Γ_s, embedded in a fluid domain, Ω_f. The right side shows the representation of the generated cut-cells from both a finite volume and finite difference perspective. Note that the finite volume approach requires more geometric information than the finite difference based approach.

The numerical difficulties facing cut-cell methods, warranting these kinds of modifications, are three-fold:

1. Evaluation of spatial derivatives on meshes where at least one point on Γ_s is arbitrarily close to a fluid point in Ω_f (termed a "degenerate mesh" herein) without compromising stability.
2. Robust and accurate interpolation of field data to evaluate spatial gradients at the boundary.
3. Robust and accurate handling of computational cells whose fluid/solid geometry varies with time.

A solution to item 2 is required for a multi-dimensional Γ_s with general boundary conditions. A solution to item 3 is required for applying the cut-cell approach to moving objects. In the present work, we focus exclusively on item 1 (known as the "small-cell problem") as this is the foundation on which a complete cut-cell method can successfully address items 2 and 3.

A comparison of a finite volume (FV) representation of a cut-cell with a finite difference (FD) representation is illustrated in Fig. 1, where an otherwise uniform mesh of constant spacing, h, is intersected with an embedded objected bounded by Γ_s. An FV scheme will typically require integrating over faces and volumes and thus requires the computation of fluid face area fractions (indicated as α_1^A and α_2^A in Fig. 1) and the fluid volume fractions (α in Fig. 1) associated with cut-cells. An FD scheme stemming from a typical method of lines discretization only requires a single 1D geometric quantity, the fluid line fraction (ψ in Fig. 1), to characterize the boundary induced by the embedded object. Although FV type cut-cell methods have dominated the literature, the FD approach is pursued in this work due to its geometric simplicity.

Regardless of the approach chosen, the small-cell problem manifests as a prohibitive increase in the stiffness of the discrete system as $\psi \to 0$ (or $\alpha \to 0$). As $\psi \to 0$, the embedded wall approaches the first fluid point in the domain leading to a degenerate mesh. Small cells leading to stiff systems have also been a problem for finite element approaches (termed cut-element methods), requiring novel stabilization techniques [11]. With the exception of preliminary work reported in [7, 36], to the best of the authors' knowledge, every cut-cell scheme that has been devised for unsteady flows, attempts to alleviate this problem through either geometric manipulations to remove the small cells or some kind of stabilizing dynamic procedure which depends on the value of the solution at any given time [1,3,4,9,10,13–18,20,22,24,25,27–30,32–35,39–45]. That is, it is the conventional wisdom that *something* must be done about small cells. In this work, we pursue a different strategy and devise schemes which are stable over the range $\psi \in [0, 1]$ (with $\psi \to 0$ representing the strongest challenge to stability and accuracy) without any geometric modifications or dynamic procedures or stabilizing source terms. In other words, small cells are not a problem for the cut-cell discretizations presented in this paper.

The order of accuracy of cut-cell schemes for unsteady problems has been largely limited to 2nd order [1,3,4,13–18, 20,22,24,25,27,29,30,32–35,39–45]. To the best of our knowledge, the scheme of [9] for incompressible flows (which was extended to compressible flows in [10]) is the only one which has demonstrated 3rd order accuracy for unsteady problems. However, Ref. [9] mentions that the dynamic correction procedure yields schemes which are unstable in the inviscid limit. To remedy this, dissipative schemes are used in [10] to extend the method to compressible flow.

In this paper, it is shown that pursuing a fundamentally different strategy for constructing stable schemes, based on truncation error matching and optimization (hence TEMO), allows for the development of 8th order explicit and compact finite differences for parabolic and elliptic equations and conservative 4th order discretizations which are stable for hyperbolic problems.

The paper is organized as follows. The formulation of a finite-differences based cut-cell scheme is described in Section 2. The construction of these novel cut-cell stencils is described in detail in Section 3. Their construction is driven by

two intuitive and simple design principles which outline the desired behavior of the cut-cell schemes in the limiting cases of $\psi = 0, 1$ through truncation error matching. These are described in Section 3.1. The application of these principles to the construction of schemes that are valid over $\psi \in [0, 1]$ and have the desired behavior at $\psi = 0, 1$ is described in Section 3.2. Conservative, stable approximations to hyperbolic equations require that certain constraints be satisfied. Discrete conservation constraints and their impact on the construction of cut-cell stencils are described in Section 3.3. The work of Ref. [6] was focused exclusively on uniform mesh boundary stencils. Specifically, the optimization procedure of Ref. [6] was not designed with cut-cell discretizations in mind. Here, it is extended to cut-cell stencils and used to determine the free parameters, allowed by the design principles, in the cut-cell discretizations. This procedure is described in Section 3.4. The result of this procedure is a set of coefficients which are polynomial functions of ψ. These closed form polynomials used for the discretizations in this manuscript are given in the supplementary material described in Appendix B. Two examples of discretizations using these coefficients are given in Appendix A. The schemes are subjected to a variety of tests in Section 4 to demonstrate their stability and accuracy, even for very challenging non-linear hyperbolic problems. In all tests considered, the schemes demonstrate the advertised order of accuracy and excellent stability properties over the whole range of $\psi \in [0, 1]$ without any small-cell corrections or numerical dissipation.

2. Formulation of finite difference based cut-cell approach

The majority of cut-cell literature has been based on finite-volume formulations, with the work of [9] and [10] as notable exceptions. A consequence of this choice is the need for advanced computational geometry functionality to handle the volume/volume intersections of the embedded objects and the Cartesian mesh cells. For example, as shown in Fig. 1, a finite volume approach will require intersecting an embedded geometry with a Cartesian mesh cell and also with all the faces of the cell to compute relevant volume and area fractions. However, a finite difference scheme only requires the intersection of the embedded object with a mesh line. It is our conjecture that a cut-cell method based on finite differences will be simpler to implement and not require expensive computational geometry routines. As such, finite differences are used as the basis for the present approach.

Consider a continuous Cartesian domain, $\mathcal{D}^c$, defined by 3 orthogonal vectors: $\boldsymbol{L}_x$, $\boldsymbol{L}_y$, and $\boldsymbol{L}_z$, with respective lengths: L_x, L_y, and L_z, such that

$$\mathcal{D}^c = \mathcal{X}^c \times \mathcal{Y}^c \times \mathcal{Z}^c \,, \tag{1}$$

where $\times$ denotes the Cartesian product and

$$\mathcal{X}^c = \{x \mid x \in [0, L_x]\} \,,$$
$$\mathcal{Y}^c = \{y \mid y \in [0, L_y]\} \,,$$
$$\mathcal{Z}^c = \{z \mid z \in [0, L_z]\} \,.$$

For a node-based scheme, the domain is discretized by first defining

$$\mathcal{I} = \{0, 1, \ldots, N_x - 1\} \,,$$
$$\mathcal{J} = \{0, 1, \ldots, N_y - 1\} \,,$$
$$\mathcal{K} = \{0, 1, \ldots, N_z - 1\} \,,$$

where N_x, N_y, and N_z are specified based on resolution requirements. These are used to write the ordered, discrete sets

$$\mathcal{X} = \{x_i \mid i \in \mathcal{I}\} \subset \mathcal{X}^c \,,$$
$$\mathcal{Y} = \{y_j \mid j \in \mathcal{J}\} \subset \mathcal{Y}^c \,,$$
$$\mathcal{Z} = \{z_k \mid k \in \mathcal{K}\} \subset \mathcal{Z}^c \,,$$

where in the present case only uniform meshes are considered (i.e. $x_i = iL_x/(N_x - 1)$). Thus, the discrete Cartesian domain, $\mathcal{D}$, is given by

$$\mathcal{D} = \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \,. \tag{2}$$

Note that this is the standard definition for a discrete Cartesian domain that would be arrived at following a method of lines spatial discretization. However, the above discussion is necessary since the description of the cut-cell approach will make use of the various constituents of $\mathcal{D}$.

Generalizing on Fig. 1, M embedded objects are introduced into $\mathcal{D}^c$. Associated with the mth object are $\Omega_{s,m}$ and $\Gamma_{s,m}$, which are continuous sets describing the embedded volume and bounding area, respectively. The total solid volume, $\mathcal{S}^c$, is given by

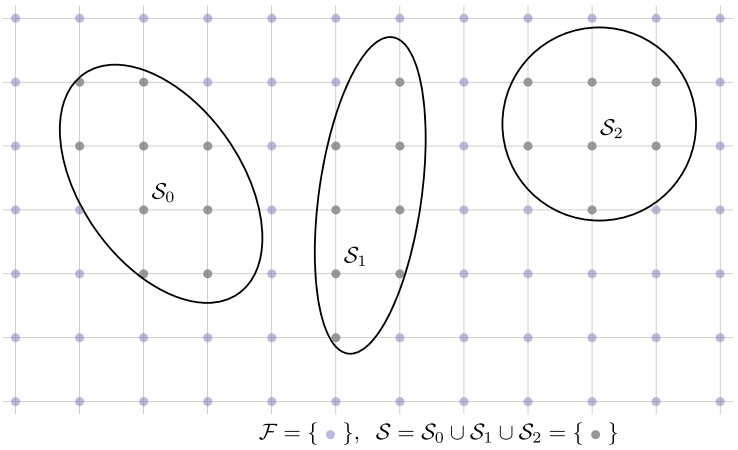

Fig. 2. The introduction of 3 embedded objects into the domain, $\mathcal{D}$. The discrete sets, $\mathcal{S}_0$, $\mathcal{S}_1$, and $\mathcal{S}_2$, which contain the points in the domain that are inside each solid, are shown. The points making up each set is highlighted in gray. The remaining nodes in the domain belong to the set of fluid nodes, $\mathcal{F}$, and are highlighted in blue. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\mathcal{S}^c = \bigcup_{m \in \mathcal{M}} \Omega_{s,m} \,,$$

where $\mathcal{M} = \{0, 1, \ldots, M-1\}$ is an indexing set for the embedded objects. The discrete version of this set is then,

$$\mathcal{S} = \mathcal{S}^c \cap \mathcal{D} \,.$$

It is also possible to group the solid points according to their embedded object index:

$$\mathcal{S}_m = \Omega_{s,m} \cap \mathcal{D} \implies \mathcal{S} = \bigcup_{m \in \mathcal{M}} \mathcal{S}_m \,.$$

The corresponding set of fluid points is then $\mathcal{F} = \mathcal{D} - \mathcal{S}$. An example of this is shown in Fig. 2 with $M = 3$. The individual contributions to the set of solid points are labeled $\mathcal{S}_0, \mathcal{S}_1$, and $\mathcal{S}_2$. It is important to note that the points in $\mathcal{S}$ are not used in the cut-cell method. That is, the governing equations are not solved in $\mathcal{S}$. Nor are they included in the solution procedure as a means of indirectly imposing boundary conditions. However, an implementation of the method may use the storage associated with those points (if any).

Splitting the domain, $\mathcal{D}$, into its disjoint constituents, $\mathcal{F}$ and $\mathcal{S}$ is not enough to fully describe a finite differences based cut-cell method. The intersections of mesh lines with the embedded objects also need to be accounted for. To this end, one dimensional rays coincident with the mesh lines are given by,

$$\begin{aligned}
\mathcal{R}^x_{jk} &= \{(x, y_j, z_k) \mid x \in \mathcal{X}^c\} \,, \\
\mathcal{R}^y_{ik} &= \{(x_i, y, z_k) \mid y \in \mathcal{Y}^c\} \\
\mathcal{R}^z_{ij} &= \{(x_i, y_j, z) \mid z \in \mathcal{Z}^c\} \,,
\end{aligned}$$

where x_i, y_j, and z_k are the elements of $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, respectively. The intersection of these rays with the mth embedded object is given by,

$$\begin{aligned}
\mathcal{R}^x_m &= \bigcup_{k \in \mathcal{K}} \bigcup_{j \in \mathcal{J}} \mathcal{R}^x_{jk} \cap \Gamma_{s,m} \,, \\
\mathcal{R}^y_m &= \bigcup_{k \in \mathcal{K}} \bigcup_{i \in \mathcal{I}} \mathcal{R}^y_{ik} \cap \Gamma_{s,m} \,, \\
\mathcal{R}^z_m &= \bigcup_{j \in \mathcal{J}} \bigcup_{i \in \mathcal{I}} \mathcal{R}^z_{ij} \cap \Gamma_{s,m} \,.
\end{aligned}$$

As each embedded object may be associated with different physical boundary conditions, it is necessary to form the sets of intersection points associated with each object. The full set of extra boundary points in each direction, induced by the embedded objects are

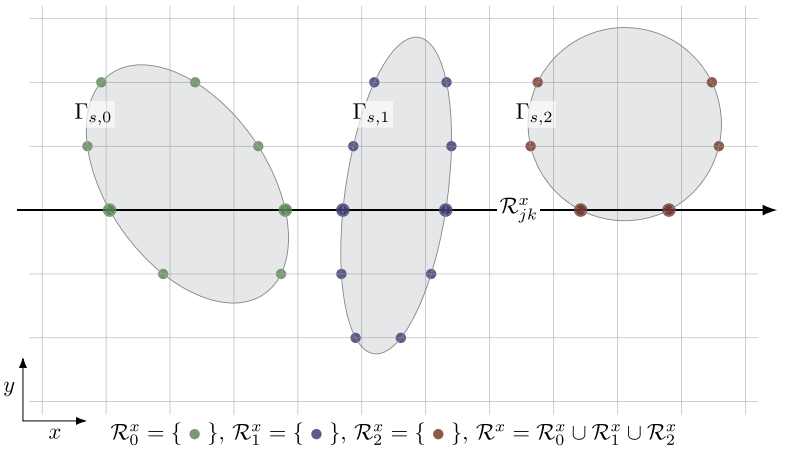

Fig. 3. The intersection of the boundary sets $\Gamma_{s,0}$, $\Gamma_{s,1}$, and $\Gamma_{s,2}$, with rays in the x direction are shown. The points in the ray/object intersection sets, $\mathcal{R}^x_0$, $\mathcal{R}^x_1$, and $\mathcal{R}^x_2$, are shown in green, blue, and red, respectively. The thicker black arrow is an example of a ray in the x direction at a given (j,k) position, $\mathcal{R}^x_{jk}$. The intersection points of this particular ray with the embedded objects are indicated with larger node sizes. The full set of intersection points can be constructed by marching such rays over all available (j,k) positions.

$$\mathcal{R}^x = \bigcup_{m\in\mathcal{M}} \mathcal{R}^x_m\,,$$
$$\mathcal{R}^y = \bigcup_{m\in\mathcal{M}} \mathcal{R}^y_m\,,$$
$$\mathcal{R}^z = \bigcup_{m\in\mathcal{M}} \mathcal{R}^z_m\,.$$

Fig. 3 shows an example of the construction of $\mathcal{R}^x_m$ for 3 embedded objects arrayed on a domain. The intersection of a ray in the x-direction, $\mathcal{R}^x_{jk}$, with the embedded objects is shown. Constructing the full set of intersection points involves marching this ray over $\mathcal{J}\times\mathcal{K}$. These intersection calculations are one dimensional calculations and are therefore relatively fast compared to the three dimensional volume intersection calculations required for finite volume cut-cell schemes. The construction of $\mathcal{R}^y$ and $\mathcal{R}^z$ follow similarly but are not drawn.

The points $\mathcal{F}$ and $\mathcal{R}^{x/y/z}$ are all that is needed when Dirichlet boundary conditions are associated with each embedded object. However, when Neumann conditions are encountered on an embedded object, or when there is no boundary condition and the value is allowed to float, then the governing equations must be solved on $\mathcal{R}^{x/y/z}$ which requires another set of points.

To describe this required final set of points, first consider the set $\mathcal{P}^x_m$ which can be used to label (or index) all the points of $\mathcal{R}^x_m$. Utilizing the same idea for the y and z directions:

$$\mathcal{P}^x_m = \{0,\ldots,|\mathcal{R}^x_m|\}\,,$$
$$\mathcal{P}^y_m = \{0,\ldots,|\mathcal{R}^y_m|\}\,,$$
$$\mathcal{P}^z_m = \{0,\ldots,|\mathcal{R}^z_m|\}\,,$$

where $|\cdot|$ is used to indicate the size of the set. An example of this labeling for $\mathcal{R}^x_0$ and $\mathcal{R}^y_0$ is shown in Fig. 4 which focuses on the $m=0$th shape from Fig. 3. To motivate the construction of the final set of points, consider the data required to evaluate $(\partial/\partial x, \partial/\partial y)$ at point 6 in $\mathcal{R}^x_0$ on the left side of Fig. 4. By construction, the point is on a mesh line in the x direction, making it straightforward to write a FD discretization of $\partial/\partial x$. However, point 6 is not on a mesh line in the y direction. Thus, formulating a FD discretization of $\partial/\partial y$ at point 6, requires another ray of data originating from 6 and proceeding in the y direction. This auxiliary ray is denoted by $\mathcal{R}^{xy}_{0,6}$ in Fig. 4 and will be explained next.

By construction, the pth element of $\mathcal{R}^x_m$ is the tuple of spatial coordinates, $(x, y_j, z_k)_p$, recording the intersection of a mesh line ray with the embedded object. To isolate different elements of the spatial coordinates in the following descriptions, the p subscript will be distributed to each component such that $(x, y_j, z_k)_p = (x_p, y_{j_p}, z_{k_p})$. The ray/object intersections are then written,

$$\mathcal{R}^x_m = \bigcup_{p\in\mathcal{P}^x_m} \{(x_p, y_{j_p}, z_{j_p})\}\,,$$
$$\mathcal{R}^y_m = \bigcup_{p\in\mathcal{P}^y_m} \{(x_{i_p}, y_p, z_{j_p})\}\,,$$

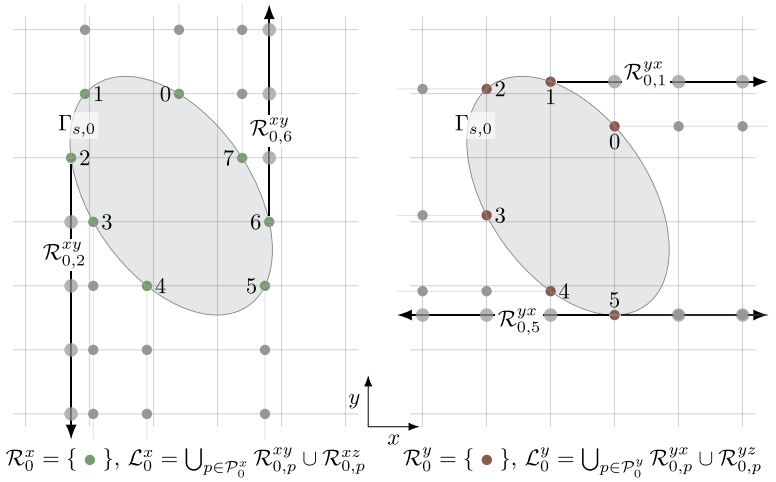

Fig. 4. An examination of the construction of the auxiliary line sets, $\mathcal{L}^x_m$, and $\mathcal{L}^y_m$, for the $m = 0$ embedded object in the domain. The labeled points in $\mathcal{R}^x_0$ and $\mathcal{R}^y_0$ are shown on the left and right, respectively. The collection of points in $\mathcal{L}^x_{m,p}$, are the intersection points of the Cartesian mesh and a ray originating from the pth point in $\mathcal{R}^x_m$ and extending in the $\pm y$ and $\pm z$ directions as appropriate.

$$\mathcal{R}^z_m = \bigcup_{p\in\mathcal{P}^z_m} \{(x_{i_p}, y_{j_p}, z_p)\}\,.$$

The set $\mathcal{R}^{xy}_{m,p}$ is introduced to describe the ray in the y direction, originating from the pth point of $\mathcal{R}^x_m$. Likewise, $\mathcal{R}^{xz}_{m,p}$ describes the ray in the z direction, originating from the pth point of $\mathcal{R}^x_m$. It is clear that the auxiliary ray sets, $\mathcal{R}^{yx}_{m,p}$ and $\mathcal{R}^{yz}_{m,p}$ associated with $\mathcal{R}^y_m$ as well as $\mathcal{R}^{zx}_{m,p}$ and $\mathcal{R}^{zy}_{m,p}$ associated with $\mathcal{R}^z_m$ are needed. They are formulated:

$$\begin{aligned}
\mathcal{R}^{xy}_{m,p} &= \{(x_p, y_j, z_{k_p}) \mid j \in [j_p \pm 1, j_p \pm q]\}\,,\\
\mathcal{R}^{xz}_{m,p} &= \{(x_p, y_{j_p}, z_k) \mid k \in [k_p \pm 1, k_p \pm q]\}\,,\\
\mathcal{R}^{yx}_{m,p} &= \{(x_i, y_p, z_{k_p}) \mid i \in [i_p \pm 1, i_p \pm q]\}\,,\\
\mathcal{R}^{yz}_{m,p} &= \{(x_{i_p}, y_p, z_k) \mid k \in [k_p \pm 1, k_p \pm q]\}\,,\\
\mathcal{R}^{zx}_{m,p} &= \{(x_i, y_{j_p}, z_p) \mid i \in [i_p \pm 1, i_p \pm q]\}\,,\\
\mathcal{R}^{zy}_{m,p} &= \{(x_{i_p}, y_j, z_p) \mid j \in [j_p \pm 1, j_p \pm q]\}\,,
\end{aligned}$$

where the q is chosen based on accuracy and stability requirements, and the choice of $\pm$ in the index range is dictated by the embedded object normal vector. For suitably aligned objects, both index ranges may be equally applicable. Fig. 4 shows the construction of the auxiliary ray sets $\mathcal{R}^{xy}_{0,p}$ and $\mathcal{R}^{yx}_{0,p}$, and gives an example of 4 of the rays to highlight the geometric interpretation of the notation. It can be convenient to group the auxiliary ray sets for an object into auxiliary line sets:

$$\begin{aligned}
\mathcal{L}^x_m &= \bigcup_{p\in\mathcal{P}^x_m} \mathcal{R}^{xy}_{m,p} \cup \mathcal{R}^{xz}_{m,p}\,,\\
\mathcal{L}^y_m &= \bigcup_{p\in\mathcal{P}^y_m} \mathcal{R}^{yx}_{m,p} \cup \mathcal{R}^{yz}_{m,p}\,,\\
\mathcal{L}^z_m &= \bigcup_{p\in\mathcal{P}^z_m} \mathcal{R}^{zx}_{m,p} \cup \mathcal{R}^{zy}_{m,p}\,.
\end{aligned}$$

Solving systems of equations with general boundary conditions and embedded object shapes, using the present finite differences based cut-cell method, requires knowledge of $\mathcal{F}$, $\mathcal{R}^x_m$, $\mathcal{R}^y_m$, $\mathcal{R}^z_m$, $\mathcal{L}^x_m$, $\mathcal{L}^y_m$, and $\mathcal{L}^z_m$. In the spirit of finite differences, all of these sets are constructed via simple one dimensional geometry operations and do not require any volume/volume intersections that are typical of finite volume based schemes. Neumann boundary conditions do require information about the surface normal and moving objects will require geometric information such as center-of-mass and orientation angles. Efficient generation of this data is outside the scope of this manuscript.

It should be noted that the auxiliary line sets, $\mathcal{L}^{x/y/z}_m$ are not on an equal footing with the points in $\mathcal{F}$ or the ray/object intersections, $\mathcal{R}^{x/y/z}$. That is, the governing equations are solved on $\mathcal{F}$ and $\mathcal{R}^{x/y/z}$ but never on $\mathcal{L}^{x/y/z}$. Rather, $\mathcal{L}^{x/y/z}$ represent convenient placeholders for one dimensional interpolation operations. In some sense, $\mathcal{L}^{x/y/z}$ could be referred to

as "ghost points". However, this terminology has been avoided since the typical use of ghost points is to avoid solving the equations directly on $\mathcal{R}^{x/y/z}$ rather than to provide supplemental information to facilitate the discretization of the system on $\mathcal{R}^{x/y/z}$.

This manuscript will not address the fully general case where $\mathcal{L}^{x/y/z} \cap \mathcal{R}^{x/y/z} \neq \varnothing$. Instead, a proof of concept for stable finite differences based cut-cell discretizations will be explored for general shapes with Dirichlet boundary conditions (i.e. $\mathcal{L}^{x/y/z}$ is not needed) or for planar shapes with Neumann and outflow conditions (i.e. $\mathcal{L}^{x/y/z} \subset \mathcal{R}^{x/y/z}$). With these simplifications for the Neumann and outflow conditions, solving the governing equations on the boundary does not require any interpolation operations as the lines of data required by the derivative approximations are aligned with the mesh. This approach allows for the cut-cell scheme to be parameterized by a single one dimensional value, ψ, the fluid line fraction (as shown in Fig. 1). Thus, the FD stencils on and near $\mathcal{R}^{x/y/z}$ can be written as functions of ψ. The stability of these schemes can be systematically assessed over the range of $\psi \in [0, 1]$. This systematic assessment can be used to optimize the cut-cell schemes for stability over the full range of $\psi \in [0, 1]$ such that no small-cell corrections or any in-situ dynamic procedures are necessary.

3. Construction of embedded stencils

The procedure for generating FD cut-cell stencils can be best understood when contrasted with the procedure for generating FD derivative approximations on Cartesian meshes which do not contain any embedded objects, that is, $\mathcal{D} = \mathcal{F}$. For example, if a two dimensional domain, $\mathcal{D}$, is discretized with N_x, N_y points in each direction, the approximation to $\partial/\partial x$ over the whole domain is given by the Kronecker product,

$$\frac{\partial}{\partial x} \approx \mathbf{O}^x = \mathbf{O}_1^x \otimes \mathbf{I}^{N_y}, \tag{3}$$

where $\mathbf{I}^{N_y}$ denotes the identity matrix with N_y elements and $\mathbf{O}_1^x$ is an $(N_x \times N_x)$ matrix describing the FD discretization along a mesh line in the x direction. The choice of $\mathbf{O}_1^x \otimes \mathbf{I}^{N_y}$ or $\mathbf{I}^{N_y} \otimes \mathbf{O}_1^x$ is arbitrary from a mathematical standpoint and should be chosen based on the layout of the data to which the operator, $\mathbf{O}^x$, will be applied. For the central finite differences considered in this paper, the sparse structure of $\mathbf{O}_1^x$ is:

$$\mathbf{O}_1^x = \begin{bmatrix} \mathbf{B}_l(\boldsymbol{\alpha}) & & \\ & \mathbf{C} & \\ & & \mathbf{B}_r(\boldsymbol{\alpha}) \end{bmatrix},$$

where $\mathbf{C}$ is a narrow band circulant matrix describing the interior discretization, and $\mathbf{B}_{l/r}(\boldsymbol{\alpha})$ are the small block matrices describing the lopsided boundary stencils needed to evaluate the derivative to high accuracy near the left, (l), and right, (r), walls. They are written as functions of free coefficients, $\boldsymbol{\alpha}$. Standard central/compact finite differences are used in the interior in of the domain. The details of the these schemes can be found many places, including Refs. [6,23]. The exact form of the boundary schemes are given later in equation (5a).

In the cut-cell case where $\mathcal{D} \neq \mathcal{F}$, the full operator $\mathbf{O}^x$ can no longer be constructed by repeating the one dimensional operator, $\mathbf{O}_1^x$, N_y times along the diagonal. Instead, a new one dimensional operator is defined:

$$\mathbf{O}_1^{x,n} = \begin{bmatrix} \mathbf{B}_l(\psi_{l,n}, \boldsymbol{\alpha}) & & \\ & \mathbf{C} & \\ & & \mathbf{B}_r(\psi_{r,n}, \boldsymbol{\alpha}) \end{bmatrix},$$

where $\mathbf{B}_{l/r}$ are now functions of a one dimensional fluid line fraction $\psi_{l/r}$. The n superscript and subscript are for labeling. In the cut-cell case, the discrete operator, $\mathbf{O}^x$, is no longer given by equation (3) but rather by,

$$\mathbf{O}^x = \begin{bmatrix} \mathbf{O}_1^{x,0} & & & & \\ & \mathbf{O}_1^{x,1} & & & \\ & & \mathbf{O}_1^{x,2} & & \\ & & & \ddots & \\ & & & & \mathbf{O}_1^{x,Q^x} \end{bmatrix}, \tag{4}$$

where Q^x is the total number of one dimensional operators required. In the case where the embedded objects do not intersect with the domain boundary, $Q^x = N_y N_z + |\mathcal{R}^x|/2$. This operator has been written as if there are no "holes" in the data, that is, the data in $\mathcal{F}$ and $\mathcal{R}^x$ have been appropriately interleaved. Operators for the y and z spacial derivatives, $\mathbf{O}^y$ and $\mathbf{O}^z$ can be similarly defined. Note that the sizes of the operators will not necessarily be equivalent since $\mathbf{O}^x$ is defined over $\mathcal{F} \cup \mathcal{R}^x$, $\mathbf{O}^y$ is defined over $\mathcal{F} \cup \mathcal{R}^y$, and $\mathbf{O}^z$ is defined over $\mathcal{F} \cup \mathcal{R}^z$. One could define mappings between these various spaces and write the operators assuming a data layout as in equation (4), or one could formulate the operators using a block diagonal matrix similar to (4) but only over $\mathcal{F}$, augmented with a more general sparse matrix over $\mathcal{R}^{x/y/z}$. Such implementation issues will not be explored but are worth noting.

Therefore, formulating cut-cell stencils in the FD framework, is equivalent to designing the boundary schemes, $\mathbf{B}_{l/r}(\psi, \boldsymbol{\alpha})$.

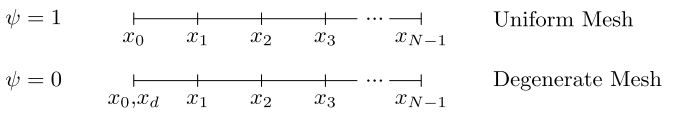

Fig. 5. Uniform mesh with N equidistant points and a corresponding degenerate mesh with degenerate point x_d collocated with the left wall at x_0.

3.1. Design principles

To design the general cut-cell boundary matrix, $\boldsymbol{B}(\psi, \boldsymbol{\alpha})$, attention is first focused on the limiting cases of the uniform mesh, $\boldsymbol{B}^u(\boldsymbol{\alpha}) = \boldsymbol{B}(1, \boldsymbol{\alpha})$ and the degenerate mesh, $\boldsymbol{B}^d(\boldsymbol{\alpha}) = \boldsymbol{B}(0, \boldsymbol{\alpha})$. In the ensuing discussion, the dependence on the free parameters, $\boldsymbol{\alpha}$, is not relevant and is dropped from the notation for brevity.

The width of the scheme used in the interior of the domain sets a minimum value on the number of required boundary stencils. Likewise, the desired order of accuracy of the boundary stencils sets a minimum requirement on their width. However, the number of boundary stencils and their respective widths may be arbitrarily large. Additional rows and columns of coefficients can be added above these baseline requirements to satisfy additional properties, such as discrete conservation [6], or energy stability [36]. Thus, two principles (in addition to discrete conservation) were selected to drive the formulation of the cut-cell boundary stencils, under the assumption that such constraints could be satisfied for suitable sizes of $\mathbf{B}_l^{u/d}$:

1. Given a uniform mesh with $\psi = 1$, the addition of a degenerate point at a wall, such that $\psi = 0$, should not change the truncation error at any point.
2. Do not violate the assumptions of continuity embedded in the FD formulation.

The first principle of truncation error matching is based on the observation of Schneiders et al. [33], described in the context of moving Γ_s: the solution deteriorates (i.e. oscillations develop) when small changes in Γ_s result in abrupt changes to the discrete operator which then result in abrupt changes in truncation error acting as essentially discontinuous forcing terms. The abrupt changes to the discrete operator are associated with 2 events: cell-crossing and cell-classification changes. In the context of [33], a cell-classification change is when a cut-cell changes from a "regular" cell requiring no corrections to a "small" cell which requires corrections. A cell-crossing event is when the geometry of a cell changes over the course of a timestep such that $\psi \approx 1 \rightarrow \psi \approx 0$ (or vice-versa). The events can be triggered by small changes Γ_s. This observation has also been made by other cut-cell researchers, for example, Brehm et al. [8]. The cut-cell method presented here does not do any small cell corrections so there are no cell-classification changes. The goal of the first principle is to avoid the abrupt changes to the discrete operator caused by cell crossing events, ensuring that small changes in Γ_s result in small changes in the truncation error of the scheme. Note that a full mitigation of cell-crossing events require a robust treatment of the "fresh-cell"/"dead-cell" problem for moving Γ_s and is beyond the scope of the present work which is focused on stationary objects.

To focus the discussion of principle 1, (the 'TEM' of TEMO), consider a domain of length L subject to two different discretizations as shown in Fig. 5. The first discretization uses N equidistant points resulting in a uniform mesh of spacing h (labeled "Uniform Mesh" in Fig. 5). The second uses the same N points as the uniform case but contains an additional degenerate point, x_d, collocated with the left wall at x_0 (labeled "Degenerate Mesh" in Fig. 5). We will focus the derivations on the boundary stencils making up $\mathbf{B}_l^{u/d}$ since the corresponding right wall stencils can be computed from a simple mapping applied to $\mathbf{B}_l^{u/d}$ [6]. If each stencil in $\mathbf{B}_l^{u/d}$ is of length t, a discrete approximation of order q of the νth order derivative of some function, f, defined on the grid and evaluated at a point, i, near the boundary, is given by:

$$f_i^{(\nu)} = \frac{1}{h^\nu} \sum_{j=0}^{t-1} \alpha_{ij}^u f_j + \tau_q^u \,, \tag{5a}$$

$$f_i^{(\nu)} = \frac{1}{h^\nu} \left[\alpha_{id}^d f_d + \sum_{j=0}^{t-1} \alpha_{ij}^d f_j \right] + \tau_q^d \,, \tag{5b}$$

where τ is the truncation error and the d and u superscripts indicate terms on the degenerate and uniform meshes, respectively. To discover the form of τ, consider the Taylor series expansion of f_j about f_i in equation (5):

$$f_j = f_i + (j-i)hf_i' + \frac{((j-i)h)^2}{2!} f_i'' + \ldots + \frac{((j-i)h)^n}{n!} f_i^{(n)} + \ldots$$

Expanding each f_j in such a manner in equation (5) yields the truncation errors associated with a qth order approximation on the uniform, τ_q^u, and degenerate, τ_q^d, meshes:

$$\tau_q^u = \frac{1}{h^\nu} \sum_{k=q+\nu} \frac{f_i^{(k)}}{k!} \left[\sum_{j=0}^{t-1} \alpha_{ij}^u \left((j-i)h\right)^k \right],$$

$$\tau_q^d = \frac{1}{h^\nu} \sum_{k=q+\nu} \frac{f_i^{(k)}}{k!} \left[(-ih)^k \alpha_{id}^d + \sum_{j=0}^{t-1} \alpha_{ij}^d \left((j-i)h\right)^k \right].$$

Associated with each truncation error is the $q+\nu$ system of equations, with t unknowns, describing the FD stencils:

$$f_i^{(\nu)} - \frac{1}{h^\nu} \sum_{k=0}^{q+\nu-1} \frac{f_i^{(k)}}{k!} \left[\sum_{j=0}^{t-1} \alpha_{ij}^u \left((j-i)h\right)^k \right] = 0,$$

$$f_i^{(\nu)} - \frac{1}{h^\nu} \sum_{k=0}^{q+\nu-1} \frac{f_i^{(k)}}{k!} \left[(-ih)^k \alpha_{id}^d + \sum_{j=0}^{t-1} \alpha_{ij}^d \left((j-i)h\right)^k \right] = 0,$$

for the uniform and degenerate meshes, respectively. Satisfying the first design principle requires $\tau_q^u = \tau_q^d$. This can be accomplished with the $r \times t$ constraints:

$$\alpha_{id}^d + \alpha_{i0}^d = \alpha_{i0}^u, \tag{6a}$$

$$\alpha_{ij}^d = \alpha_{ij}^u, \quad \text{for all } j \in [1, t). \tag{6b}$$

Note that this principle also constrains the coefficients required to evaluate the discrete derivative at the degenerate point, $f_d^{(\nu)}$. Since this point is collocated with x_0, the discrete derivatives must have the same truncation error (i.e. the same coefficients).

The second design principle: do not violate the assumptions of continuity inherent in the FD formulation, is used to supply one more relation and thus allow for the determination of all α^d in terms of α^u, or equivalently, writing $\mathbf{B}_l^d$ in terms of $\mathbf{B}_l^u$. To illustrate the implications of this constraint, consider a degenerate mesh initialized with some solution, $f(x)$. At this initial time, $f(x_0) = f(x_d)$ since $x_0 = x_d$. Suppose that the temporal evolution of f is governed by some sort of PDE such that the change in f between timesteps is governed by the discrete spatial derivatives constructed according to the first principle. The update to f_0 and f_d will be identical since the discrete derivatives are equal. However, f_0 will be subject to discrete boundary conditions and f_d will not. This will result in a small difference between f_0 and f_d. Thus, the assumption of continuity is violated in any discretization that is constructed based on a Taylor series approximation and includes both of these points. It can be seen that by the formulation of the truncation error given in equation (5), the discrete approximations on the uniform and degenerate meshes are only strictly equal if $f_0 = f_d$. The application of boundary conditions (specifically, Dirichlet) at only x_0, makes them only approximately equal.

In order to avoid this violation, no stencil uses data from both x_0 and x_d as $\psi \to 0$. Thus, the second principle refines equation (6a) and requires:

$$\alpha_{id}^d = \alpha_{i0}^u \qquad \text{or} \qquad \alpha_{i0}^d = \alpha_{i0}^u. \tag{7}$$

Recall that $\mathbf{B}_l^d$ operates on $[f_0, f_d, f_1, \dots]^T$. Therefore, this constraint impacts the first two columns of every row of $\mathbf{B}_l^d$. If there are r rows in $\mathbf{B}_l^d$, this constraint leaves 2^r combinations of parameters to be explored. All combinations meet the desired accuracy constraints by construction, and are judged, in this work, by their stability characteristics on various test problems. These considerations lead to 3 of the possible 2^r combinations being selected. To explore the 3 combinations of parameters, consider $\bar{\mathbf{B}}_l^u$, the 3×3 submatrix of $\mathbf{B}_l^u$:

$$\bar{\mathbf{B}}_l^u = \begin{bmatrix} \alpha_{00}^u & \alpha_{01}^u & \alpha_{02}^u \\ \alpha_{10}^u & \alpha_{11}^u & \alpha_{12}^u \\ \alpha_{20}^u & \alpha_{21}^u & \alpha_{22}^u \end{bmatrix}.$$

If the choices for the degenerate boundary matrices are written as $\mathbf{B}_l^{d,0}$, $\mathbf{B}_l^{d,1}$, and $\mathbf{B}_l^{d,2}$, their respective 3×3 submatrices are:

$$\bar{\mathbf{B}}_l^{d,0} = \begin{bmatrix} \alpha_{00}^u & 0 & \alpha_{01}^u \\ \alpha_{00}^u & 0 & \alpha_{01}^u \\ 0 & \alpha_{10}^u & \alpha_{11}^u \end{bmatrix}, \qquad \bar{\mathbf{B}}_l^{d,1} = \begin{bmatrix} \alpha_{00}^u & 0 & \alpha_{01}^u \\ 0 & \alpha_{00}^u & \alpha_{01}^u \\ 0 & \alpha_{10}^u & \alpha_{11}^u \end{bmatrix}, \qquad \bar{\mathbf{B}}_l^{d,2} = \begin{bmatrix} 0 & \alpha_{00}^u & \alpha_{01}^u \\ \alpha_{00}^u & 0 & \alpha_{01}^u \\ 0 & \alpha_{10}^u & \alpha_{11}^u \end{bmatrix}.$$

Note that the last row is the same in all submatrices. All schemes in this paper use $\alpha_{i0}^d = 0$ for all $i \geq 1$.

It was found that $\mathbf{B}_l^{d,0}$ is suitable for first derivative approximations (as found in hyperbolic problems). Second derivative operators, found in parabolic and elliptic problems, were best discretized with $\mathbf{B}_l^{d,1}$ when Neumann boundary conditions are

used and $\mathbf{B}_l^{d,2}$ when Dirichlet or floating conditions are encountered. Boundary stencils to be used with Neumann conditions are of the form:

$$f_i^{(\nu)} = \frac{1}{h^{\nu-1}}\eta_i f_0' + \frac{1}{h^\nu}\sum_{j=0}^{j=t-1}\alpha_{ij} f_j + \mathcal{O}\left(h^q\right), \tag{8}$$

where the value f_0' is prescribed. In addition to the constraints on α previously discussed, the design principles dictate:

$$\eta_0^d = \eta_d^d = \eta_0^u,$$
$$\eta_i^d = \eta_i^u \text{ for all } i \geq 1.$$

This paper also considers compact finite differences [23] for the 2nd derivative approximations. As opposed to the typical formulations, these boundary schemes use central differences at the first point, that is, equation (5a) at $i = 0$, augmented with:

$$\sum_{k=-1}^{k=1}\beta_{ik} f_{i+k}^{(\nu)} = \frac{1}{h^\nu}\sum_{j=0}^{j=t-1}\alpha_{ij} f_j + \mathcal{O}\left(h^q\right) \quad \text{for } i > 0, \tag{9}$$

where $\beta_{i0} = 1$. On the degenerate mesh, the stencil evaluating $f_d^{(\nu)}$ uses an explicit FD formulation as well which is constrained according to $\mathbf{B}_l^{d,1}$ or $\mathbf{B}_l^{d,2}$. In addition to the constraints on α already discussed, the design principles dictate constraints on β: $\beta_{ik}^d = \beta_{ik}^u$.

3.2. Application of design principles

The two design principles determine the relationship between the boundary operators on a uniform mesh, $\mathbf{B}_l^u$, and a degenerate mesh, $\mathbf{B}_l^d$. The construction of cut-cell boundary schemes, $\mathbf{B}_l(\psi)$, which are valid over the range $\psi \in [0, 1]$, and have the limiting behavior: $\mathbf{B}_l(1) = \mathbf{B}_l^u$ and $\mathbf{B}_l(0) = \mathbf{B}_l^d$, are now considered.

If the size of $\mathbf{B}_l^u$ is given by $r \times t$, the size of $\mathbf{B}_l^d$ must be $(r+1) \times (t+1)$ Therefore, the size of the general $\mathbf{B}_l(\psi)$ must also be $(r+1) \times (t+1)$, with its final row being equivalent to the interior scheme when $\psi = 1$. Assuming the order of accuracy constraints were satisfied with only t points, the extra width, $t+1$, brings with it a free parameter. The design principles inform the choices of the free parameter. Recall that the second principle requires that as $\psi \to 0$ either the coefficient modifying the wall or the degenerate point also has to approach zero. The free parameter choices are therefore:

$$\alpha_{i\delta} = \psi\alpha_{i\delta}^u \quad \text{for } i < r$$
$$\alpha_{r,r-p} = \psi\gamma_{-p}$$

where δ indicates the appropriate coefficient to send to zero which is selected according to the choice of $\mathbf{B}_l^{d,0/1/2}$, and γ_{-p} is the left most coefficient of the centered interior scheme described by the $p+1$ coefficients, $\{\gamma_{-p}, \cdots, \gamma_p\}$. Details on the interior schemes can be found in Ref. [6]. It is possible (and even necessary when enforcing conservation) that there will be more than one free parameter in a given boundary stencil. In such cases, the extra free parameters are written as first degree polynomials in ψ in terms of the uniform mesh coefficients, $\alpha_{ij} = \psi\alpha_{ij}^u + (1-\psi)\alpha_{i-1,j-1}^u$. This ensures that the correct behavior is achieved at $\psi = 0, 1$. This choice of polynomial is not unique but has proved effective.

Consider again the discrete approximation of order q of the νth order derivative of some function, f, defined on the grid and evaluated at a point, i, near the boundary (i.e. $i \leq r$) written with the constraints imposed by the design principles:

$$f_i^{(\nu)} = \frac{1}{h^\nu}\left[\psi\alpha_{i\delta}^u f_\delta + \sum_{j=0,\, j!=\delta}^{t}\alpha_{ij} f_j\right] + \tau_q,$$

where δ is again used to select the index of the coefficient that will be zeroed out in $\mathbf{B}_l^d$. The data is defined on the cut-cell mesh with x_0 on Γ_s and

$$x_{i+1} - x_i = \begin{cases} \psi h, & \text{for } i = 0 \\ h, & \text{for } i > 0 \end{cases}.$$

With $\Delta_{ij} = (x_j - x_i)$, the system of $(q+\nu)$ equations for the accuracy requirements is given by

$$f_i^{(\nu)} - \frac{1}{h^\nu}\sum_{k=0}^{q+\nu-1}\frac{f_i^{(k)}}{k!}\left[\psi\alpha_{i\delta}^u\Delta_{i\delta}^k + \sum_{j=0,\, j!=\delta}^{j=t}\alpha_{ij}\Delta_{ij}^k\right] = 0.$$

Note that the α^u terms are specified rather than solved for. This avoids any singularities in the coefficients in the range $\psi \in [0, 1]$ and satisfies the design principles by construction. The interested reader can easily verify this property of all schemes presented in this paper.

At this point, it is worthwhile to highlight that any free parameters in $\mathbf{B}_l(\psi)$ will be functions of the uniform mesh coefficients, α^u, by virtue of the above construction. Thus, when writing the full boundary stencil matrix, $\mathbf{B}_l(\psi, \boldsymbol{\alpha})$, it is understood that the vector of free coefficients, $\boldsymbol{\alpha}$ coefficients are the free coefficients in $\mathbf{B}_l^u$. Some of the free coefficients are further constrained by discrete conservation and removed from $\boldsymbol{\alpha}$. The remaining coefficients play a critical role in stability optimization procedure discussed in the following sections.

3.3. Discrete conservation constraints

Discrete conservation is enforced for the first derivative operators. An involved discussion of the constraints which must be satisfied by the boundary stencils to be discretely conservative for hyperbolic systems is given in [6]. In the interest of conciseness, the discussion that follows assumes the reader is familiar with the constraints and will only present the single additional constraint introduced by the cut-cell boundary schemes.

To define a "conservative" approximation, consider a scalar hyperbolic conservation law with the form:

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0\,,$$

for $x \in [0, L]$ where $f = f(u)$ is some flux function. Integrating this equation over the domain gives:

$$\frac{d}{dt}\int_0^L u(x,t)\,dx = f|_{x=0,t=t} \;-\; f|_{x=L,t=t}\,.$$

Thus, the solution to this equation has the property that the total change of u as a function of time is driven solely by the flux function, f, at the domain boundaries. Therefore a conservative approximation is one which satisfies a discrete version of the above integral for some choice of quadrature. It can be shown that the one dimensional operator, $\mathbf{O}$, is discretely conservative if one can derive a set of quadratures, $\mathbf{w}$, such that

$$\mathbf{w}^T\mathbf{O}_i = 0\,, \qquad \text{for } i \in (0, N-1)\,, \tag{10}$$

where $\mathbf{O}_i$ is the ith column of $\mathbf{O}$ and $\mathbf{w}$ is the column vector of quadratures. This does not constrain the first column, $\mathbf{O}_0$ (thus $\mathbf{B}_0$), of the operator. However, an examination of the boundary stencil matrices on uniform and degenerate meshes shows that the coefficients in the first column of $\mathbf{B}_l^u$ move into the second column of $\mathbf{B}_l^d$. Thus, the coefficients are not constrained by equation (10) on the uniform mesh, but are constrained in the degenerate case. This behavior makes it difficult to properly impose the constraint on $\mathbf{B}_l(\psi)$. To address this problem, another constraint is introduced to ensure the proper limiting behavior:

$$w_0\alpha_{00} = \mathbf{w}_u^T\mathbf{B}_{l\,0}^u\,,$$

where the u subscript denotes quantities on the uniform mesh and $\mathbf{B}_{l\,0}^u$ indicates the first column of the uniform mesh boundary matrix, $\mathbf{B}_l^u$. Therefore the conservation constraints considered in this paper form a system of $t+1$ equations and $r+1$ unknowns.

As discussed in [6], the size of $\mathbf{B}_l$ determines the solvability of the system. Following the guidelines therein, for a set of $r+1$ boundary stencils of order q and length $t+1$ to be coupled with an interior scheme of order $2p$, r and t are given by:

$$t = p + q + 1 + n_{extra}, \tag{11a}$$

$$r = q + 1 + n_{extra}\,, \tag{11b}$$

where n_{extra} allows for the addition of extra rows and columns to provide additional free parameters in $\boldsymbol{\alpha}$. This allows for the construction of boundary schemes of any order $q < 2p$ which satisfy conservation constraints. The cut-cell schemes generated in this way will have a number of free parameters which are written as polynomials of ψ and $\boldsymbol{\alpha}$. Note that the conservation constraints must also be solved on the uniform mesh to provide appropriately constrained α^u. In the next section, we discuss how to tune the free $\boldsymbol{\alpha}$ for stability.

3.4. Optimization for stability

The optimization procedure (the 'O' of TEMO) was successfully carried out for 2 families of schemes: $E2_1$ and $E4_1$. The E prefix indicates the schemes are using explicit (as opposed to compact) finite differences. The number indicates the global order of the method, and the subscript indicates the derivative order. Stability optimization is limited to first derivative operators because in the context of fluid dynamics, the second derivative operators are associated with dissipative

Table 1
Conservative cut-cell schemes, for 1st derivatives, described in terms of the interior discretization, p, order of boundary schemes, q, extra rows/columns added for optimization, n_{extra}, free parameters, $\boldsymbol{\alpha}$, and zeroed parameters.

Scheme	p	q	n_{extra}	$\boldsymbol{\alpha}$	zeros
$E2_1$	1	1	1	α^u_{02}, α^u_{03}, α^u_{12}, α^u_{13}	
$E4_1$	2	3	0	α^u_{04}, α^u_{14}, α^u_{24}, α^u_{25}	α^u_{05}, α^u_{15}

phenomena. The form of the conservative cut-cell schemes for 1st derivative operators, constructed according to the design principles, are given in Table 1. The table lists the width, p, of the interior scheme of order $2p$, the boundary scheme order, q, the number of extra rows/columns that were added to facilitate optimization, n_{extra} and the resulting free parameters in the scheme, $\boldsymbol{\alpha}$. The footprint of the boundary scheme can be computed according to equation (11). Schemes $E2_1$ and $E4_1$ utilize boundary schemes that are one order less than the interior scheme, resulting in globally 2nd and 4th order discretizations [19], respectively.

We label a system as "naturally" stable if the boundary and interior schemes can be successfully used in a variety of numerical tests with the free parameters set to 0. These tests will be described in section 4. None of the schemes described in Table 1 have this property. That is, the values of the free parameters must be carefully chosen to ensure stability. To address this issue, we apply a slightly modified version of the optimization strategy presented in [6]. As a brief recap, that strategy consists of:

1. Evaluate the objective function, θ, at N random points in $\boldsymbol{\alpha}$ space.
2. Use the best $\boldsymbol{\alpha}$ as a starting point for a gradient ascent method using a line search approach to take the largest possible steps
3. Record $\boldsymbol{\alpha}$ in a database for further processing once θ reaches a critical value
4. Go back to step 1 when it is no long possible to find a better $\boldsymbol{\alpha}$

The most important aspect of the optimization procedure is the use of an objective function which can serve as a proxy for a stability estimate. The differences in the present optimization procedure and that of Ref. [6] consist of a change in the objective function, specifically, the addition of the $\boldsymbol{\psi}$ parameter to Algorithms 1–3 and its impact on mesh and operator construction. One could follow the same procedure of adding more parameters to the objective function if optimizations over a different (or larger) space were desired. Adding $\boldsymbol{\psi}$ in the present case is sufficient to make the algorithm "cut-cell aware". In contrast to the uniform mesh used in [6], the unoptimized base versions of the cut-cell stencils are far too large to print.

The optimization process to find suitable $\boldsymbol{\alpha}$, such that the resulting operator, $\mathbf{O}$, yields a stable discretization of relevant systems, is based directly on the compressible Euler equations:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} &= 0\,, \\
\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} &= -\frac{\partial p}{\partial x_i}\,, \\
\frac{\partial \rho E}{\partial t} + \frac{\partial \rho E u_i}{\partial x_i} &= -\frac{\partial p u_i}{\partial x_i}\,,
\end{aligned}$$

where ρ is the density, p the pressure, E the total energy, u_i the ith component of the velocity vector, and the Einstein summation convention is assumed. The system is then closed by an equation of state assuming a calorically perfect gas,

$$p = (\gamma - 1)\,(\rho E - \rho u_i u_i/2)\,.$$

The speed of sound is $a_0 = \sqrt{\gamma p/\rho}$. For the numerical tests, the ratio of specific heats, γ, is assumed equal to 1.4, corresponding to air.

The test problem is one-dimensional and starts with an initially quiescent fluid with a Gaussian density distribution:

$$\rho_I = \rho(x,0) = 1 + \frac{\exp(-\frac{(x-\mu)^2}{2\sigma})}{\sqrt{2\pi\sigma}}\,, \tag{12a}$$

$$E_I = E(x,0) = \frac{\rho^{\gamma-1}}{\gamma - 1}\,, \tag{12b}$$

where the energy has been initialized using isentropic relations. The one dimensional computational domain is depicted in Fig. 6. The mesh has a uniform grid spacing, h, excepting the spacing between the embedded objects and first/last fluid nodes in $\mathcal{F}$, with respective spacings $\psi_l h$ and $\psi_r h$. The embedded objects form the boundaries of the domain such that

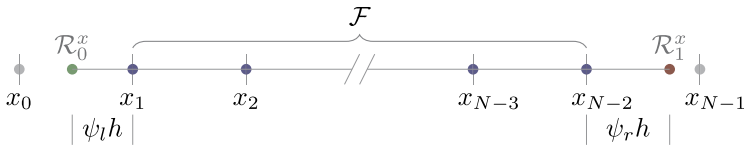

Fig. 6. One dimensional domain with N points and regular spacing h, used in optimization simulations. There are two intersection points near the left and right boundaries of the domain, indicated with the sets $\mathcal{R}_0^x$ and $\mathcal{R}_1^x$, respectively. The interior $N-2$ points make up the fluid node set, $\mathcal{F}$. The one dimensional fluid line fractions, ψ_l and ψ_r are varied independently. A uniform mesh is achieved when $\psi = 1$, a degenerate mesh when $\psi = 0$. In contrast to the domain in Fig. 5, the fully degenerate mesh will be $2h$ shorter than the uniform mesh in the present case. This allows for the optimization simulations to easily explore the full range $\psi \in [0, 1]$.

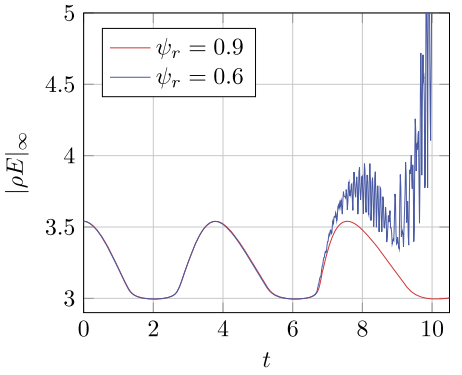

Fig. 7. Maximum energy, $|\rho E|_\infty$, over the domain as a function of time, t, for cut-cell meshes with $\psi_l = 1$ and the indicated value of ψ_r for scheme $E4_1$ with $\boldsymbol{\alpha} = 0$. The domain is discretized with $N = 121$ points. The RK4 method is used to integrate the equations in time with $C = 0.5$. The simulation is not stable for $\psi_r = 0.6$.

cut-cell boundary schemes are relevant at both ends of the domain. The fluid line fraction between the embedded wall on the left and first fluid point is ψ_l. Likewise, ψ_r is the fluid line fraction between the last point in $\mathcal{F}$ and the embedded right wall. Both ψ_l and ψ_r are varied independently. The only physical boundary condition for this inviscid flow is that the normal component of the velocity be zero at the walls [2]. In the one-dimensional case this becomes:

$$u(x_1 - \psi_l h, t) = u(x_{N-2} + \psi_r h, t) = 0\,.$$

The boundary conditions for this problem lead to the conservation of the total mass and energy within the domain:

$$\frac{d}{dt}\int \rho\, dx = 0, \qquad \frac{d}{dt}\int \rho E\, dx = 0\,.$$

The use of conservative schemes ensures that these relations are satisfied discretely (to within machine precision). This is verified using the quadrature weights, $\mathbf{w}$, derived with each scheme. The timestep restriction is given by the well-known CFL constraint,

$$\Delta t = \frac{C\,h}{\max(|u| + a_0)}\,, \tag{13}$$

where typically $C \leq 1$. Note that the timestep restriction is written in terms of the uniform mesh spacing, h, and makes no reference to ψ. Since the equations are non-linear and there is no dissipation, infinitely thin shocks will develop. The simulations are stopped before this happens when there are still about 15 points resolving the wavefront on the coarsest grid. The maximum energy at a given time, $|\rho E|_\infty$, is shown in Fig. 7 for $E4_1$ with $\boldsymbol{\alpha} = 0$ and parameters: $L = x_{N-1} - x_0 = 5$, $\mu = 5$, $\sigma = 2$, $N = 121$, $C = 0.5$, for the cases of $(\psi_l = 1, \psi_r = 0.9)$ and $(\psi_l = 1, \psi_r = 0.6)$. The simulation becomes unstable for decreasing ψ, highlighting the need for an optimization procedure to chose $\boldsymbol{\alpha}$ that yield stable schemes over the range $\psi \in [0, 1]$.

As in Ref. [6], the objective function, θ, is chosen so as to provided a reasonable quantification of numerical stability, such that maximizing θ yields stable discretizations for systems of interest. It is split into two helper functions. The first of these, $\mathcal{T}(\mathbf{O}, \boldsymbol{\alpha}, \boldsymbol{N}, \boldsymbol{C}, \boldsymbol{\psi}, I, t_c) \to t_r$, quantifies the run time, t_r, of a set of simulations of the desired equations with a particular boundary and interior scheme described by $\mathbf{O}$ and a particular set of free parameters $\boldsymbol{\alpha}$ for a set of grid resolutions, CFL numbers, embedded wall distances, and initial conditions given by $\boldsymbol{N}$, $\boldsymbol{C}$, $\boldsymbol{\psi}$ and I, respectively. The computation of t_r is described in Algorithm 1. Using this algorithm, it is clear that $t_r \in (0, t_c]$.

The second helper function, $\mathcal{E}(\boldsymbol{O}, \boldsymbol{\alpha}, \boldsymbol{N}, \boldsymbol{C}, \boldsymbol{\psi}, I, t_c, \boldsymbol{R}) \to \epsilon$, quantifies the solution smoothness for the time and space interval specified by $\boldsymbol{R}$. We make use of the monotonic nature of the energy per unit volume, ρE, near the walls at a late

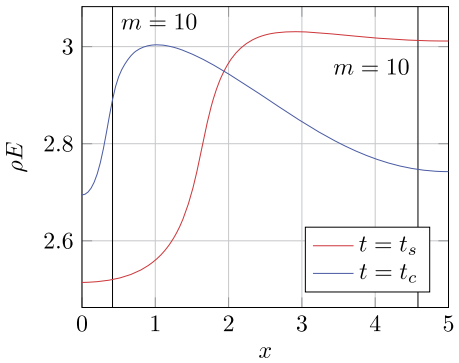

Fig. 8. Energy, ρE at times $t = t_s = 9.5$ and $t = t_c = 10.5$ over the spacial domain. The areas near the left and right boundaries delimited by the black vertical lines correspond to the boundary domains used for the monotonicity error when $m = 10$ in equation (14).

input : Boundary and interior scheme, $\mathbf{O}$
input : Set of free parameters, $\boldsymbol{\alpha}$
input : Set of grid resolutions for spatial discretization, $\boldsymbol{N}$
input : Set of time constraints for temporal discretization, $\boldsymbol{C}$
input : Set of embedded wall distances, $\boldsymbol{\psi}$
input : Initial conditions, I
input : Completion time, t_c
output: Average run time: t_r

$t_r \leftarrow 0$;
foreach $\{\psi, N, C\}$ *of* $\boldsymbol{\psi} \times \boldsymbol{N} \times C$ **do**
  Initialize simulation of N grid points and initial conditions I;
  Set coefficients for derivative operator, $\mathbf{O}(\boldsymbol{\alpha}, \psi_l = \psi_r = \psi)$;
  Integrate equations in time using time constraint, C, until a time of t_d where $t_d = t_c$ if the simulation completed successfully or $t_d < t_c$ if the simulation diverged at t_d;
  $t_r \leftarrow t_r + t_d$;
end
$t_r \leftarrow t_r/(|\boldsymbol{N}||\boldsymbol{C}||\boldsymbol{\psi}|)$;

Algorithm 1: $\mathcal{T}(O, \boldsymbol{\alpha}, N, C, \psi, I, t_c) \rightarrow t_r$.

time ($t \in [9.5, 10.5]$) when the initial density, ρ_I and energy, E_I, are given by equation (12), with $\sigma = 2$ and $\mu = L = 5$. The monotonicity of the energy at times $t = 9.5$ and $t = 10.5$ can be seen in Fig. 8. Thus, for a grid with N points, the monotonicity error on the left and right boundaries is:

$$\mathcal{M}_L^m(f) = \left| TV_0^m(f) - (f_{m+1} - f_0) \right| \tag{14a}$$

$$\mathcal{M}_R^m(f) = \left| TV_{N-m}^N(f) - (f_{N-m-2} - f_{N-1}) \right|, \tag{14b}$$

where total variation is given by:

$$TV_j^k(f) = \sum_{i=j}^{i=k} |f_{i+1} - f_i|, \tag{15}$$

and f_i is the computed solution at point i. In equations (14) and (15), f_0 and f_{N-1} refer to f evaluated at the left and right embedded wall: $\mathcal{R}_{0,0}^x$ and $\mathcal{R}_{1,0}^x$, respectively, rather than at the points x_0 and x_{N-1} which are in $\mathcal{S}$. Note that $\mathcal{M}_L^m$ has been defined for a monotonically increasing function and $\mathcal{M}_R^m$ has been defined for a monotonically decreasing function to reflect the different behavior of ρE near the left and right boundaries, respectively. With these definitions, the procedure for computing ϵ is given in Algorithm 2.

The objective function $\theta(\boldsymbol{O}, \boldsymbol{\alpha}, \boldsymbol{N}, \boldsymbol{C}, \boldsymbol{\psi}, I, t_c, \boldsymbol{R}) \rightarrow \nu$ (where ν is a measure of the error ϵ) can then be defined in terms of $\mathcal{T}$ and $\mathcal{E}$ and is given in Algorithm 3. With this, the optimization problem is stated as: For a given stencil, $\boldsymbol{O}$, set of grid resolutions, $\boldsymbol{N}$, set of temporal resolutions $\boldsymbol{C}$, set of embedded wall distances, $\boldsymbol{\psi}$, initial conditions, I, completion time, t_c and interval for smoothness calculations, $\boldsymbol{R}$, choose $\boldsymbol{\alpha}$ such that θ is maximized for the desired equations. The parameters used for optimization in the present paper are:

$$\boldsymbol{N} = \{121, 151, 201\}$$

$$\boldsymbol{C} = \{0.8, 0.1\}$$

input : Boundary and interior scheme, $\boldsymbol{O}$
input : Set of free parameters, $\boldsymbol{\alpha}$
input : Set of grid resolutions for spatial discretization, $\boldsymbol{N}$
input : Set of time constraints for temporal discretization, $\boldsymbol{C}$
input : Set of embedded wall distances, $\boldsymbol{\psi}$
input : Initial conditions, I
input : Completion time, t_c
input : Space-time interval for smoothness calculations, $\boldsymbol{R} = (t_s, t_e, m)$
output: Maximum error: ϵ

$\epsilon \leftarrow 0$;
foreach $\{\psi, N, C\}$ *of* $\boldsymbol{\psi} \times \boldsymbol{N} \times C$ **do**
  Initialize simulation with N grid points and initial conditions I;
  Set coefficients for derivative operator, $\mathbf{O}(\boldsymbol{\alpha}, \psi_l = \psi_r = \psi)$;
  Integrate equations in time using time constraint, C, until a time of t_d where $t_d = t_c$ if the simulation completed successfully or $t_d < t_c$ if the simulation diverged at t_d;
  if $t_d < t_c$ **then**
    $\epsilon \leftarrow$ SENTINEL;
    return;
  end
  Write $\boldsymbol{\alpha}$ to database;
  $\epsilon_L \leftarrow$ temporal average of $\mathcal{M}_L^m(\rho E)$ over time interval $t \in [t_s, t_e]$;
  $\epsilon_R \leftarrow$ temporal average of $\mathcal{M}_R^m(\rho E)$ over time interval $t \in [t_s, t_e]$;
  $\epsilon \leftarrow \max(\epsilon, \epsilon_L + \epsilon_R)$;
end

Algorithm 2: $\mathcal{E}\,(O, \boldsymbol{\alpha}, N, \psi, C, I, t_c, R) \rightarrow \epsilon$.

input : Boundary and interior scheme, $\boldsymbol{O}$
input : Set of free parameters, $\boldsymbol{\alpha}$
input : Set of grid resolutions for spatial discretization, $\boldsymbol{N}$
input : Set of CFL constraints for temporal discretization, $\boldsymbol{C}$
input : Set of embedded wall distances, $\boldsymbol{\psi}$
input : Initial conditions, I
input : Completion time, t_c
input : Space-time interval for smoothness calculations, $\boldsymbol{R} = (t_s, t_e, m)$
output: Quantification of stability: ν

$\nu \leftarrow \mathcal{T}\,(\boldsymbol{O}, \boldsymbol{\alpha}, \boldsymbol{N}, \boldsymbol{C}, \boldsymbol{\psi}, I, t_c)$;
$\epsilon \leftarrow \mathcal{E}\,(\boldsymbol{O}, \boldsymbol{\alpha}, \boldsymbol{N}, \boldsymbol{C}, \boldsymbol{\psi}, I, t_c, \boldsymbol{R})$;
if $\epsilon \neq$ *SENTINEL* **then** $\nu \leftarrow \nu + \log(1/\epsilon)$;

Algorithm 3: $\theta\,(O, \boldsymbol{\alpha}, N, C, \psi, I, t_c, R) \rightarrow \nu$.

$$\boldsymbol{\psi} = \{10^{-6}, 10^{-3}, 10^{-2}, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$$
$$I = (\rho_I, E_I, \sigma = 2, \mu = 5, L = 5)$$
$$t_c = 10.5$$
$$\boldsymbol{R} = (t_s = 9.5, t_e = t_c, m = 10)$$

A gradient ascent method is applied to move through $\boldsymbol{\alpha}$ space using a line search approach and arrive at suitable schemes. It should be noted that the objective function described here is not unique but has proven effective in finding schemes that behave well in systems that are similar to those encountered in studying fluid dynamics. The optimization that is done here is based on solving the compressible Euler equations. This tends to yield a great number of candidate boundary stencils which are then further refined by the stability tests in the next section. The candidate boundary stencils are all written to a database for further processing via the stability tests in the next section. Note that all $\boldsymbol{\alpha}$ for which $t_d == t_c$ are written to the database rather than just those which maximize θ. This is done because θ is not an exact quantification of stability, and so, beyond a certain threshold, it is not clear that the $\boldsymbol{\alpha}$ which yield the largest values of θ are strictly better than those which yield slightly lower values. Thus, the "effective" objective function is a combination of all these tests, and provides a reasonable proxy for stability over a range of systems. Software limitations prevented this "effective" objective function from being directly implemented as such. The benefits of combining the present method of separate objective function and refining steps into a single objective function are being explored.

It is worthwhile to discuss the n_{extra} parameter in Table 1. The most obvious choice for this parameter is 0. However, during the course of optimizing $E2_1$, no schemes with $n_{extra} = 0$ were found to be stable over the whole range of hyperbolic tests considered in Section 4. Adding one extra row and column to the boundary scheme was enough to enable the optimization process to find stable $E2_1$ schemes. If other systems of equations were found for which the present batch of cut-cell schemes were unstable, one could increase n_{extra} until stable schemes were found.

The optimization and refining process discovered 5300 stable $E2_1$ schemes and 2 stable $E4_1$ schemes. These are included in a database in the supplementary materials. The first scheme for each family in the database is shown in Appendix A. In terms of accuracy on the test problems of Section 4, the schemes are roughly equivalent. Choosing the "best" scheme for a particular application would involve using the schemes directly for the desired application or designing a test problem to use a proxy for the application and sorting the schemes based on whichever criteria is deemed most important.

The optimization process was unnecessary for constructing second derivative approximations in parabolic and elliptic equations. The schemes labeled: $E2_2^d$, $E4_2^d$, $E6_2^d$, and $E8_2^d$ refer to approximations of the second derivative using explicit (central) differences such that the global orders are 2, 4, 6, and 8, respectively, for Dirichlet or floating boundaries. Likewise, $E2_2^n$, $E4_2^n$, $E6_2^n$, and $E8_2^n$ are boundary closures designed for Neumann boundary conditions. Compact differences are also used for second derivative approximations resulting the $T4_2^d$, $T6_2^d$, and $T8_2^d$ of orders 4, 6, and 8, respectively. Similarly, the boundary schemes for Neumann conditions are: $T4_2^n$, $T6_2^n$, $T8_2^n$. The coefficients for each scheme are given in the supplementary material described in Appendix B.

4. Results

The stability and accuracy of the schemes for parabolic and hyperbolic equations is demonstrated through a variety of tests and analysis. The asymptotic stability of the second derivative approximations is explored through an eigenvalue analysis in Section 4.1. The accuracy and stability of these schemes is demonstrated by solving the unsteady heat equation in Section 4.2 for a variety of embedded wall distances to highlight the behavior of the schemes on degenerate meshes. The behavior of the schemes for convex geometries is explored in Section 4.3. The behavior of the schemes when applied to elliptic problems is examined in Sections 4.4 and 4.5. The asymptotic stability of the conservative first derivative approximations given in Appendix A are also explored through an eigenvalue analysis presented in Section 4.6. The accuracy and long-time stability of the conservative cut-cell discretizations is assessed through two challenging hyperbolic problems. In the first, in Section 4.7, a varying coefficient advection equation is simulated for long times over a convex geometry. For the second test, in Section 4.8, the compressible Euler equations are solved with a moving (supersonic) isentropic vortex. The embedded wall distances are again varied at the supersonic inflow/outflow boundaries. Centered differences are used in the interior of the domain. These non-dissipative interior schemes make the hyperbolic tests especially difficult. In all tests considered, the schemes demonstrate the advertised order of accuracy and excellent stability properties over the whole range of $\psi \in [0, 1]$.

4.1. Asymptotic stability: eigenvalue analysis for parabolic terms

An asymptotically stable scheme is one for which the error does not grow unphysically with time [12]. To illustrate, consider the linear parabolic equation,

$$\frac{\partial u}{\partial t} = k \frac{\partial^2 u}{\partial x^2} ,$$

defined on the cut-cell mesh given in Fig. 6. As a notational convenience, x_l is the x-coordinate of the point describing the left embedded wall, $\mathcal{R}^x_{0,0}$. Similarly, x_r is the x-coordinate of the right embedded wall, $\mathcal{R}^x_{1,0}$. With this notation, the initial and boundary conditions for $u = u(x, t)$ are:

$$u(x, 0) = f(x) , \qquad u(x_l, t) = a(t) , \qquad \left.\frac{\partial u}{\partial x}\right|_{x=x_r} = b(t) ,$$

where it is assumed that the boundary conditions are consistent with the initial conditions. In its typical formulation, the discrete second derivative operator, $\mathbf{O}^{x,2}$, allows,

$$U'' = \mathbf{O}^{x,2} U ,$$

where $\mathbf{O}^{x,2}$ has dimensions $N \times N$ and U, U'' are column vectors of length N. However, the derivative information at x_l is not needed due to Dirichlet boundary conditions. Therefore, let the $(N-1) \times (N-1)$ submatrix of $\mathbf{O}^{x,2}$, which does not include the first row or column of $\mathbf{O}^{x,2}$, be denoted as $\mathbf{Q}$ and let $\hat{U} = [u_1, u_2, \ldots, u_{N-2}, u_r]^T$. With this notation, the semi-discrete parabolic equation can be written as,

$$\frac{\partial \hat{U}}{\partial t} = \frac{k}{h^2} \mathbf{Q} \hat{U} + G , \tag{16}$$

where G is a column vector of length $N-1$ giving the appropriate weights of the stencils on the boundary data, $a(t)$ and $b(t)$. The stability of this semi-discrete system is governed by the eigenvalues, λ, of the spatial discretization matrix, $\mathbf{Q}$. If the real part of an eigenvalue is given by $\mathrm{re}(\lambda)$ and the maximum real part of all the eigenvalues is given by $\max_\lambda \mathrm{re}(\lambda)$, the semi-discrete system is then stable if [37],

Table 2
For the 8th order tridiagonal compact scheme, $T8_2$, with a given resolution, N, the maximum value of $\max_\lambda \mathrm{re}(\lambda)$ over 169 combinations of (ψ_l, ψ_r) is given in the second column.

N	$\max_\psi(\max_\lambda \mathrm{re}(\lambda))$
31	$-2.74 \cdot 10^{-3}$
41	$-1.54 \cdot 10^{-3}$
51	$-9.87 \cdot 10^{-4}$
61	$-6.85 \cdot 10^{-4}$
71	$-5.04 \cdot 10^{-4}$
81	$-3.86 \cdot 10^{-4}$
91	$-3.05 \cdot 10^{-4}$

$$\max_\lambda \mathrm{re}(\lambda) \leq 0\,. \tag{17}$$

The discretization matrix $\mathbf{Q}$ can be constructed for any scheme by consulting Appendix B. The compact schemes require solving a system in order to form $\mathbf{O}$ and, consequently, $\mathbf{Q}$. The stability of each scheme is assessed by examining the eigenvalues of 1183 cases defined by the parameter space, $\{\mathbf{N} \times \boldsymbol{\psi}_l \times \boldsymbol{\psi}_r\}$. The set of discretizations examined is $\mathbf{N} = \{31, 41, 51, 61, 71, 81, 91\}$. The ψ values considered are: $\boldsymbol{\psi}_l = \boldsymbol{\psi}_r = \{0, 0.001, 0.01, 0.1, 0.2, \ldots, 1.0\}$. Recall that ψ_l affects the Dirichlet boundary stencil, while ψ_r impacts the Neumann boundary stencil. A summary of the results for $T8_2$ are given in Table 2. Similar results demonstrating stability can be shown for all schemes and cases. The data for every case is in the supplementary material.

4.2. Two-dimensional concave geometry: constant coefficient heat equation

The stability and accuracy of the discretizations for the 2nd derivatives given in the appendix for Dirichlet and Neumann boundary conditions are demonstrated by considering the heat equation:

$$\frac{\partial T}{\partial t} = k\nabla^2 T\,,$$

on the domain (x, y) for $x \in [x_l, x_r]$, $y \in [y_l, y_r]$, where

$$x_l = h(1-\psi_l) \qquad y_l = h(1-\psi_l)$$
$$x_r = L - h(1-\psi_r) \qquad y_r = L - h(1-\psi_r)\,.$$

This domain is a two dimensional extension of that shown in Fig. 6. The length of the domain is $L = 2$ and the uniform mesh spacing is given by $h = L/(N-1)$. A manufactured solution, T_M, was chosen for this test:

$$T_M(x, y, t) = \sum_n A_n \cos(f_n t) \exp\left(-\frac{(x-\mu_{n,x})^2}{2\sigma_{n,x}^2} - \frac{(y-\mu_{n,y})^2}{2\sigma_{n,y}^2}\right) \tag{18}$$

with parameters:

n	A_n	f_n	μ_n	σ_n
0	2	0.5	(0.4, 0.5)	(0.5, 0.2)
1	0.5	1	(0.2, 2)	(1/3, 0.8)
2	−1.2	0.8	(1.5, 1.6)	(0.3, 0.8)
3	3	0.2	(1.8, 0.3)	(2/3, 0.9)

Following the method of manufactured solutions [5,31], the modified governing equation for this problem becomes:

$$\frac{\partial T}{\partial t} = k\nabla^2 T + \frac{\partial T_M}{\partial t} - k\nabla^2 T_M \tag{19}$$

The boundary conditions are given by:

$$T(x_l, y, t) = T_M(x_l, y, t)\,, \qquad T(x, y_l, t) = T_M(x, y_l, t)\,,$$
$$\left.\frac{\partial T}{\partial x}\right|_{x=x_r} = \left.\frac{\partial T_M}{\partial x}\right|_{x=x_r}, \qquad \left.\frac{\partial T}{\partial y}\right|_{y=y_r} = \left.\frac{\partial T_M}{\partial y}\right|_{y=y_r}.$$

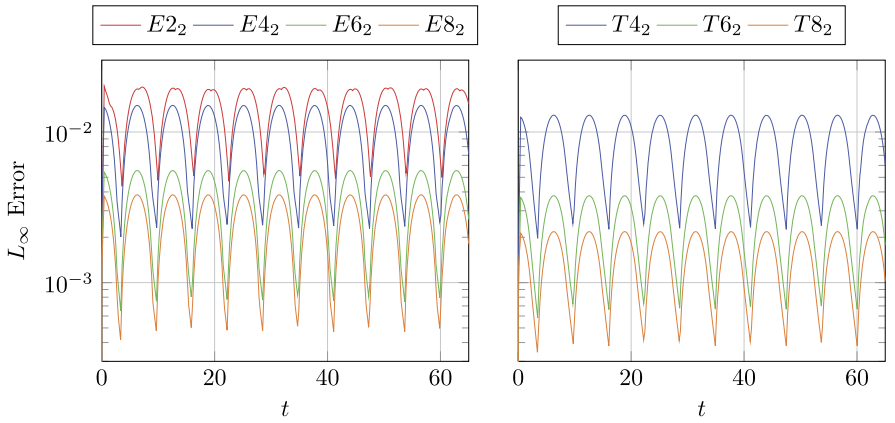

Fig. 9. Time history of L_∞ for $N = 31$ and $(\psi_l, \psi_r) = (0, 0)$ for (left) central schemes and (right) compact schemes when solving the heat equation (19).

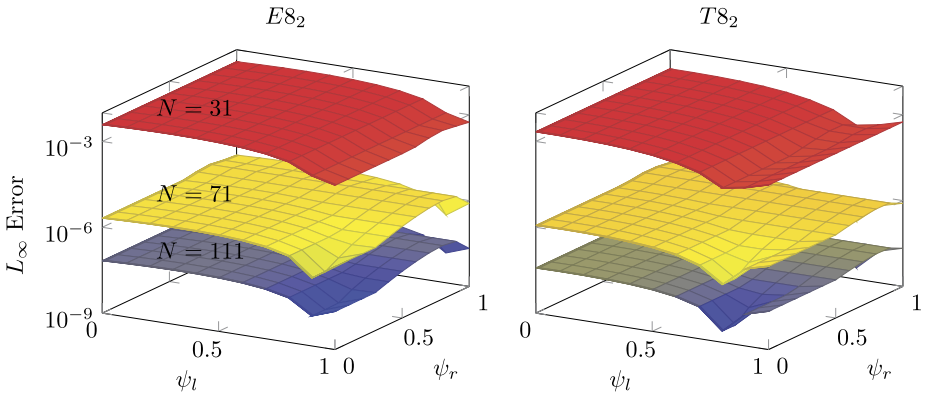

Fig. 10. Maximum L_∞ for $t > 50$ as a function of ψ_l and ψ_r for $N = 31$, 71 and 111 using the 8th order $E8_2$ and $T8_2$ schemes when solving the heat equation (19).

This set of conditions was chosen to ensure that all combinations of corner boundary conditions would be encountered. The discretized heat equation is advanced in time using the standard 4th order Runge-Kutta (RK4) method with the timestep constraint given by the well known parabolic stability constraint:

$$\Delta t = C\frac{h^2}{4k}.$$

For schemes $E2_2$, $E4_2$, $E6_2$, and $T4_2$, $C = 0.4$ was chosen. The stiffness of the higher order schemes necessitated slightly smaller timesteps with $C = 0.2$ for $E8_2$ and $T6_2$, and $C = 0.1$ for $T8_2$. A thermal diffusivity of $k = 1/30$ was used. An implicit time integration scheme could have been used to circumvent the parabolic timestep restrictions. However, an explicit integration scheme has been used to highlight the fact that small cells do not lead to an excessively stiff system.

Fig. 9 shows the evolution of the L_∞ error norm for all second derivative operators with $N = 31$ and $\psi_l = \psi_r = 0$. As can be seen, the operators result in stable discretizations even for this very challenging case and require no reduction in the timestep compared to the uniform mesh case.

Fig. 10 shows the maximum L_∞ error for $t > 50$ for the 8th order second derivative operators $E8_2$ and $T8_2$ as functions of ψ_l and ψ_r for $N = 31$, 71, and 111. The set of sampled values for (ψ_l, ψ_r), is adopted from the eigenvalue examination in Section 4.1. There is some variation in the measured L_∞ error with ψ_l and ψ_r. The supplementary material contains the results for $\mathbf{N} = \{31, 51, 71, 91, 111\}$, for all schemes. The schemes remain well behaved in all cases.

The accuracy of the schemes is demonstrated in Fig. 11, which shows the maximum L_∞ error for $t > 50$ for all second derivative operators at $N = 31, 51, 71, 91, 111$ in the fully degenerate case of $\psi_l = \psi_r = 0$ (filled symbols) and the uniform case with $\psi_l = \psi_r = 1$ (open symbols). The labeled solid black lines indicate the observed order of convergence for each scheme on the degenerate mesh. It can be observed that under mesh refinement the 2nd and 4th order schemes yield the same results in the uniform and degenerate cases while small differences persist for the 6th and 8th order schemes. The cause of this discrepancy is twofold. Firstly, the domain size and the location of boundary condition application are functions of ψ and are therefore different on the two meshes. Secondly, according to the discussion in Section 3.1, the application of boundary conditions leads to discrete operators with slightly different truncation errors in the uniform and degenerate cases. The sum of these differences is washed out by the truncation errors of the 2nd and 4th order schemes but does

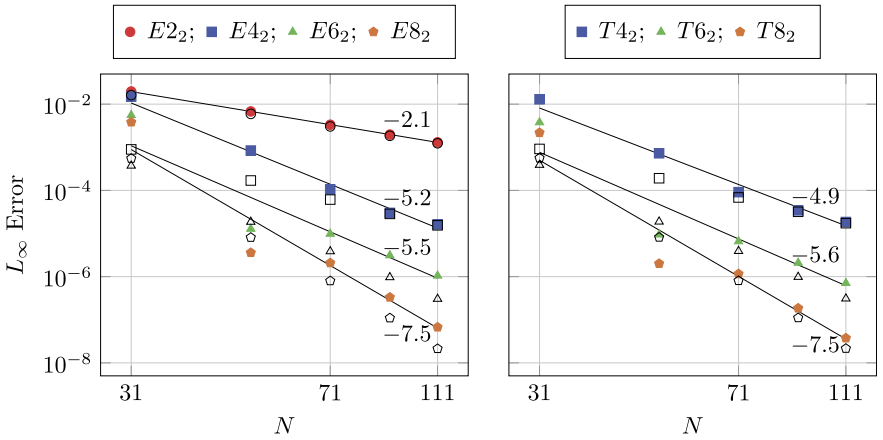

Fig. 11. Maximum L_∞ for $t > 50$ for the central and compact second derivative operators applied to equation (19). The filled data points correspond to $\psi_l = \psi_r = 0$. For comparison, the open data points correspond to $\psi_l = \psi_r = 1$. The shapes of the open data points match the shapes (and therefore schemes) of the filled points. A line (on the log-log plot) is fit to the $\psi = 0$ data and annotated with its slope, indicating the convergence of the scheme.

have an impact on the higher order schemes. All schemes demonstrate the expected convergence on both the uniform and degenerate cases under grid refinement.

This test case has allowed for fully characterizing the behavior of the cut-cell operators over the full range $\psi \in [0, 1]$, for both Dirichlet and Neumann boundary conditions. In the next test case, a convex geometry is considered to explore the impact of having stencils with different values ψ throughout their neighborhood.

4.3. Two-dimensional convex geometry: constant coefficient heat equation

In this section, the constant coefficient heat equation with the manufactured solution term given by equation (19) is again solved but on a different domain. Rather than planes of varying ψ, the embedded portion of the domain is described by,

$$G(x, y) = \sqrt{(x - x_c)^2 + (y - y_c)^2} - r\,, \tag{20a}$$

$$\Omega_s = \{(x, y) \mid G(x, y) < 0\}\,, \tag{20b}$$

$$\Gamma_s = \{(x, y) \mid G(x, y) = 0\}\,, \tag{20c}$$

where the object subscript has been dropped since there is only one object. The initial conditions are given by the manufactured solution field. The boundary conditions are:

$$T(x, y, t)\big|_{(x,y)\in\Gamma_s} = T_M(x, y, t)\,,$$

$$\left.\frac{\partial T(x, y, t)}{\partial x}\right|_{x=0|2} = \left.\frac{\partial T_M(x, y, t)}{\partial x}\right|_{x=0|2}\,,$$

$$\left.\frac{\partial T(x, y, t)}{\partial y}\right|_{y=0|2} = \left.\frac{\partial T_M(x, y, t)}{\partial y}\right|_{y=0|2}\,.$$

To avoid any fortuitous cancellation of errors due to problem symmetry, the embedded object is placed off-center with respect to the mesh with $x_c = 16/17$, $y_c = 25/22$, and $r = \sqrt{3}/10$. The initial conditions for this geometry and manufactured solution are shown in Fig. 12. With the given initial conditions, and the timestepping described in the previous section, the solution is advanced in time until $t = 65$. An equidistant uniform mesh is used with $N_x = N_y = N$. Fig. 13 shows the L_∞ error norm as a function of time for all second derivative operators with $N = 51$. The operators are all well behaved and yield stable simulations.

The order of accuracy of the schemes is examined by observing the maximum L_∞ error norm for $t > 50$ for $N = 51$, 71, 91, 111. The number of points per diameter, N_p, for the grid sizes are: $N_p = 8.7$, 12.1, 15.6, and 19.1. The error for the different schemes is shown in Fig. 14. The observed order of each schemes is given by a labeled solid black line. In all cases, the schemes achieve or are very close to their design order of accuracy. Note again that this is accomplished without any geometry modifications or in-situ correction procedure.

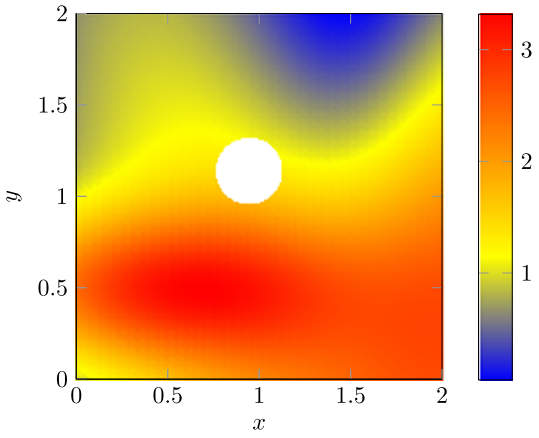

Fig. 12. Initial conditions for convex geometry test in Section 4.3.

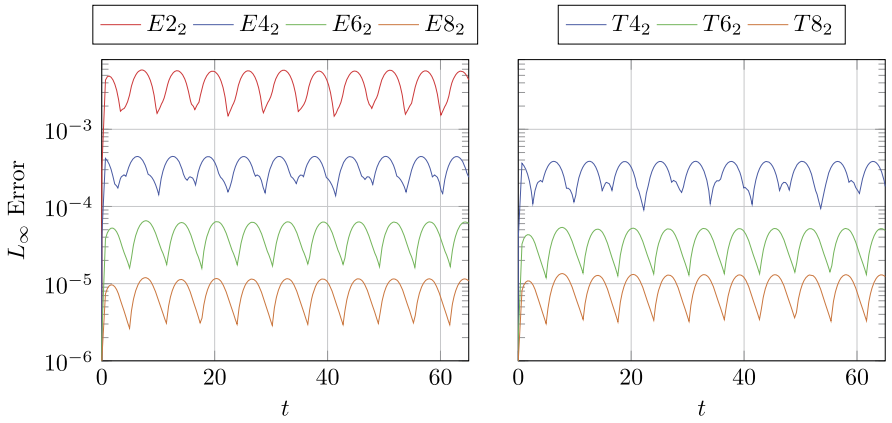

Fig. 13. Time history of L_∞ for test case in Section 4.3 with $N_x = N_y = 51$ for (left) central schemes and (right) compact schemes.

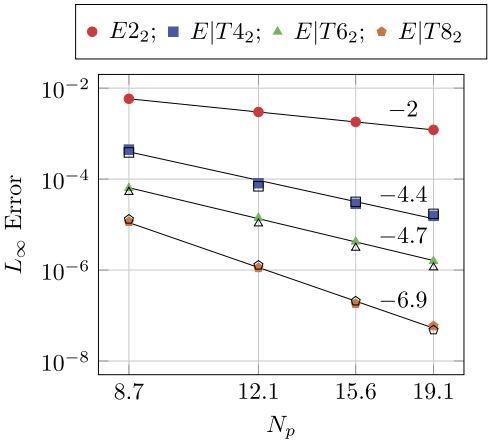

Fig. 14. Maximum L_∞ for $t > 50$ for all second derivative operators for the concave geometry test in Section 4.3. The filled points correspond to the central schemes. The open points correspond to the compact schemes. A line (on the log-log plot) is fit to the data for the central schemes and annotated with its slope, indicating the observed convergence of the schemes over the range of grids.

4.4. Two-dimensional concave geometry: constant coefficient Poisson equation

In this section we examine the behavior of the second derivative cut-cell operators in the context of a Poisson equation with walls offset by ψ_l and ψ_r as in Section 4.2. The equation to be solved is:

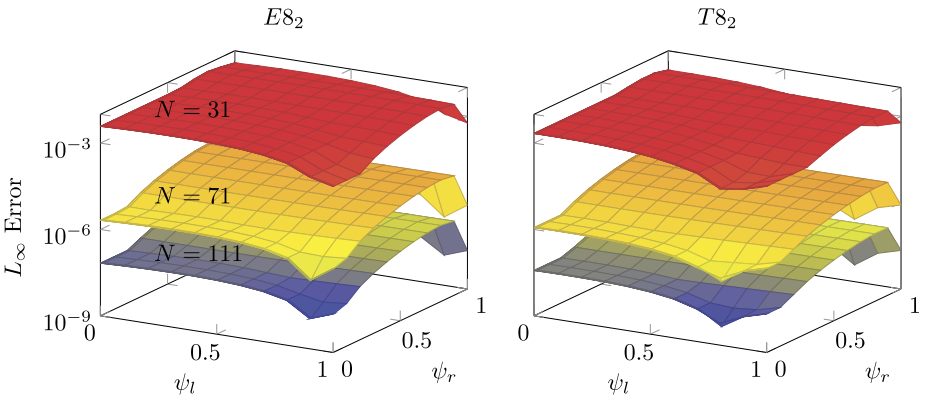

Fig. 15. L_∞ error norm as a function of ψ_l and ψ_r for $N = 31$, 71 and 111 using the 8th order $E8_2$ and $T8_2$ schemes.

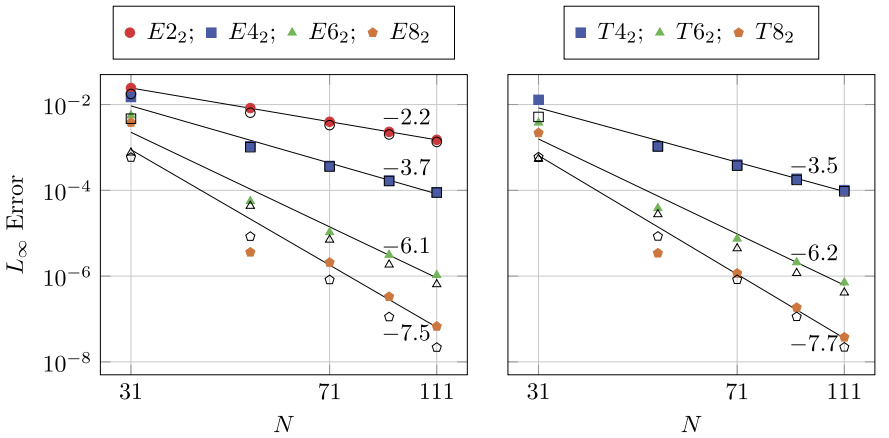

Fig. 16. L_∞ error norm for (left) central differences and (right) compact differences applied to equation (21). The filled data points correspond to $\psi_l = \psi_r = 0$. For comparison, the open data points correspond to $\psi_l = \psi_r = 1$. The shapes of the open data points match the shapes (and therefore schemes) of the filled points. A line (on the log-log plots) is fit to the $\psi = 0$ data and annotated with its slope, indicating the convergence of the scheme.

$$\nabla^2 T = \nabla^2 T_M(x, y, 0) \tag{21}$$

where T_M is given by equation (18). As in Section 4.2, the domain is given by (x, y) for $x \in [x_l, x_r]$, $y \in [y_l, y_r]$ where

$$x_l = h(1 - \psi_l)\,, \qquad y_l = h(1 - \psi_l)\,,$$
$$x_r = L - h(1 - \psi_r)\,, \qquad y_r = L - h(1 - \psi_r)\,.$$

The boundary conditions are:

$$T(x_l, y) = T_M(x_l, y, 0)\,, \qquad T(x, y_l) = T_M(x, y_l, 0)\,,$$
$$\left.\frac{\partial T(x,y)}{\partial x}\right|_{x=x_r} = \left.\frac{\partial T_M(x,y,0)}{\partial x}\right|_{x=x_r}, \qquad \left.\frac{\partial T(x,y)}{\partial y}\right|_{y=y_r} = \left.\frac{\partial T_M(x,y,0)}{\partial y}\right|_{y=y_r}.$$

A systematic study of the behavior of the schemes is conducted by solving equation (21) and examining the L_∞ errors of 845 cases defined by the parameter space, $\{\mathbf{N} \times \boldsymbol{\psi}_l \times \boldsymbol{\psi}_r\}$. The set of discretizations examined is $\mathbf{N} = \{31, 51, 71, 91, 111\}$. As in Sections 4.1 and 4.2, the ψ values considered are $\boldsymbol{\psi}_l = \boldsymbol{\psi}_r = \{0, 0.001, 0.01, 0.1, 0.2, \ldots, 1.0\}$. This is done to force the small-cell issue.

Fig. 15 shows the L_∞ error for the 8th order second derivative operators $E8_2$ and $T8_2$ as functions of ψ_l and ψ_r for $N = 31$, 71, and 111. Compared to the results for the heat equation in Fig. 10, there is more variation with the observed L_∞ error with ψ_l and ψ_r, but the errors remain well-behaved in all cases. Similar results for all schemes can be seen by examining the supplementary materials.

The accuracy of the schemes is demonstrated in Fig. 16, which shows the L_∞ error norm for all second derivative operators for all N in the fully degenerate case of $\psi_l = \psi_r = 0$ (filled points), as well as the uniform case with $\psi_l = \psi_r = 1$ (open point). The variation of L_∞ with ψ is not as strong with the 2nd and 4th order methods so the filled and open points

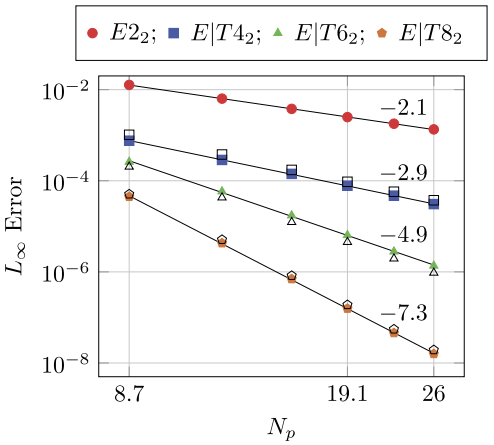

Fig. 17. L_∞ error norm for all second derivative operators as a function of points-per-diameter, N_p, applied to the Poisson equation for the concave geometry test in Section 4.5. The filled points correspond to the central schemes. The open points correspond to the compact schemes. A line (on the log-log plot) is fit to the data for the central schemes and annotated with its slope, indicating the observed converged of the schemes over the range of grids.

are largely collocated. There is a noticeable difference in the error magnitude between the uniform and degenerate cases for the 6th and 8th order schemes as per the discussion in Section 4.2, however, the order of the convergence of the schemes is as expected in all cases.

4.5. Two-dimensional convex geometry: constant coefficient Poisson equation

In this section, the constant coefficient Poisson equation with the manufactured solution term given by equation (21) is solved but on a domain described by equation (20), allowing for a circle in the domain centered at (x_c, y_c). The boundary conditions are:

$$
\begin{aligned}
T(x,y)\big|_{(x,y)\in\Gamma_s} &= T_M(x,y,0)\,,\\
\left.\frac{\partial T(x,y)}{\partial x}\right|_{x=0|2} &= \left.\frac{\partial T_M(x,y,0)}{\partial x}\right|_{x=0|2}\\
\left.\frac{\partial T(x,y)}{\partial y}\right|_{y=0|2} &= \left.\frac{\partial T_M(x,y,0)}{\partial y}\right|_{y=0|2}.
\end{aligned}
$$

As before, the embedded object is placed off-center with respect to the mesh with $x_c = 16/17$, $y_c = 25/22$, and $r = \sqrt{3}/10$. The solution for this geometry and manufactured solution are given by the initial conditions of the problem described in Section 4.3 and shown in Fig. 12.

The order of accuracy of the schemes is examined by computing the L_∞ error norm for $N = 51, 71, 91, 111, 131, 151$. The number of points per diameter, N_p, for each of the grid sizes are: $N_p = 8.7, 12.1, 15.6, 19.1, 22.5$, and 26. The error for the different schemes is shown in Fig. 17. The data corresponding the central schemes: $E2_2$, $E4_2$, $E6_2$ and $E8_2$ are shown with filled points. The compact schemes make use of the same stencil for first point and are therefore not very different. The open data points are used to indicate the L_∞ error for $T4_2$, $T6_2$, and $T8_2$. A least squares line fit is done for the central schemes and is annotated with the slope. While $E2_2$ converges with the desired accuracy, the 4th, 6th and 8th order schemes converge with one order less than expected. The intended order could be achieved by increasing the order of the cut-cell boundary schemes by 1, but such schemes are unstable for parabolic problems. Future work may explore optimizing such schemes for stability following the procedure in Section 3.4. It should again be noted that there were no geometry modifications to avoid the small cell problem or any other corrective procedure.

4.6. Asymptotic stability: eigenvalue analysis for hyperbolic terms

Similar to the eigenvalue analysis performed on the heat equation, the asymptotic stability of the $E2_1$ and $E4_1$ schemes is assessed in the context of the linear hyperbolic equation:

$$
\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0\,, \qquad \text{for}\, x_l \le x \le x_r,\ \text{and}\, t \ge 0\,,
$$

where the domain is once again given by Fig. 6, with a uniform mesh spacing of h throughout the domain excepting the near wall distances of $\psi_l h$ and $\psi_r h$. The initial and boundary conditions are:

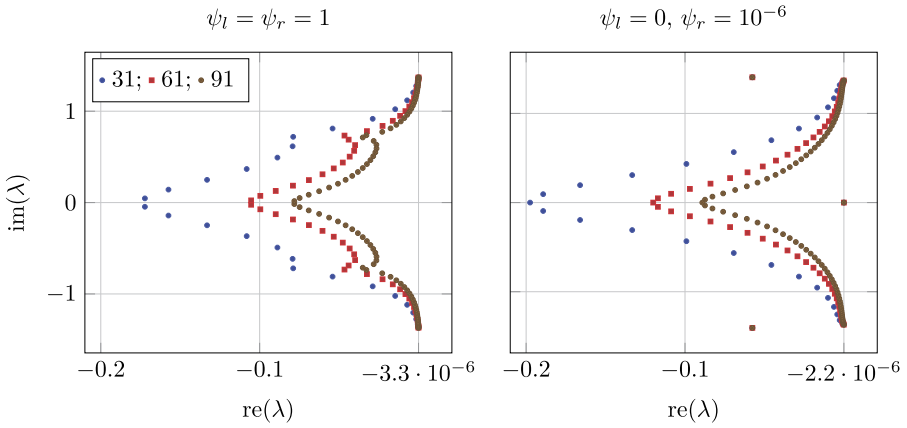

Fig. 18. Eigenvalue spectrum for $E4_1$ applied to equation (22) at the indicated (ψ_l/ψ_r) combinations and grid resolutions. The left boundary, ψ_l, is associated with an inflow boundary condition. The right boundary, ψ_r is associated with an outflow boundary condition. For each plot, $\max_\lambda \mathrm{re}(\lambda)$ is indicated by the right most tick mark. It remains negative, indicating stability.

$$u(x,0) = f(x)\,, \qquad u(x_l,t) = a(t)\,.$$

In its typical formulation, the discrete first derivative operator, $\mathbf{O}^{x,1}$, allows,

$$U' = \mathbf{O}^{x,1} U\,,$$

where $\mathbf{O}^{x,1}$ has dimensions $N \times N$ and U, U' are column vectors of length N. However, the derivative information at x_l is not needed due to Dirichlet (inflow) boundary conditions. Therefore, let the $(N-1)\times(N-1)$ submatrix of $-\mathbf{O}^{x,1}$, which does not include the first row or column of $\mathbf{O}^{x,1}$, be denoted as $\mathbf{Q}$ and let $\hat{U} = [u_1, u_2, \ldots, u_{N-2}, u_r]^T$. With this notation, the semi-discrete linear hyperbolic equation can be written as:

$$\frac{\partial \hat{U}}{\partial t} = \frac{1}{h}\mathbf{Q}\hat{U} + G\,, \tag{22}$$

where G is a column vector of length $N-1$ giving the appropriate weights of the stencils on the boundary data, $a(t)$. The stability of this semi-discrete system is governed by the eigenvalues, λ, of the spatial discretization matrix, $\mathbf{Q}$ and is stable if condition (17) is satisfied. The supplementary material contains the information for constructing 5302 discretization matrices. Fig. 18 highlights the eigenvalue spectrum for the $E4_1$ scheme given in Table A.4. The real (re) and imaginary (im) components of the eigenvalues are shown for $\psi_l = \psi_r = 1$ and $\psi_l = 0, \psi_r = 10^{-6}$. Both plots indicate the largest real part of the eigenvalue spectra with the right most tick label. In both cases the stability condition (17) is satisfied. Note that the outflow stencil will contain a repeated row at $\psi_r = 0$, yielding $\lambda = 0$. Fig. 18 demonstrates that the scheme approaches this limit in a stable manner.

The stability of the 5302 schemes is assessed by examining the eigenvalues of 676 cases defined by the parameter space, $\{\mathbf{N} \times \boldsymbol{\psi}_l \times \boldsymbol{\psi}_r\}$. The set of discretizations examined is $\mathbf{N} = \{31, 61, 91, 121\}$. The ψ considered are $\boldsymbol{\psi}_l = \boldsymbol{\psi}_r = \{10^{-6}, 0.001, 0.01, 0.1, 0.2, \ldots, 1.0\}$ Recall that ψ_l impacts the inflow stencil while ψ_r impacts the outflow stencil. The supplementary material records $\max_\lambda \mathrm{re}(\lambda)$ for all cases and shows that no eigenvalues with positive real parts are found.

4.7. Two-dimensional convex geometry: varying coefficient hyperbolic equation

In this section, the accuracy and time stability of the conservative cut-cell schemes are demonstrated by solving a varying coefficient hyperbolic equation:

$$\frac{\partial u}{\partial t} + \nabla G \cdot \nabla u = 0\,, \qquad L_x = L_y = 2\,, \tag{23}$$

where $G(x, y)$ and the embedded object are given by equation (20). For this test, 4 different embedded circles, $l_n = (x_{c_n}, y_{c_n})$, are used in order to trigger different geometry errors. Each object has a radius of $r = 0.2$ and centers: $l_0 = (1.053, 0.901)$, $l_1 = (1.033, 1.101)$, $l_2 = (0.989, 1.037)$, $l_3 = (0.933, 0.999)$. The initial and boundary conditions for each shape are given by:

$$u(x, y, 0) = \sin(2\pi G)\,, \qquad u(x, y, t)\big|_{G=0} = -\sin(2\pi t)$$

where (x_c, y_c) in G are taken from the particular l_n being used. Fig. 19 shows the initial conditions for geometry l_0. The exact solution is a circular pulse radiating out from the embedded object with a period of 1:

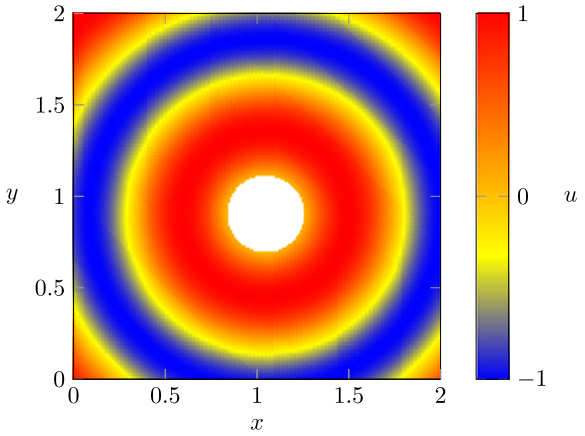

Fig. 19. Initial conditions hyperbolic scalar equation (23) for geometry l_0. The circular pulse radiates out from the embedded object as time evolves.

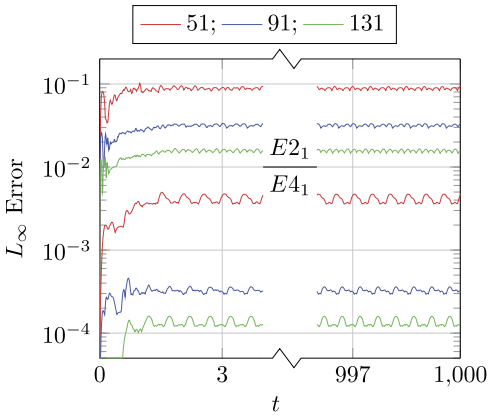

Fig. 20. L_∞ error as a function of time for $E2_1$ (top 3 curves) and $E4_1$ (bottom 3 curves) for grid sizes $N = 51, 91, 131$ with time step constraint $C = 0.5$. The break in the graph allows for focusing on the initial and final times of the simulation.

$$u(x, y, t) = \sin(2\pi(G - t))$$

This series of tests consisted of constant CFL tests with $C = 0.5$ and constant timestep tests with $\Delta t = 8 \cdot 10^{-4}$. The constant CFL tests allow for demonstrating that the cut-cell schemes do not require special timestep restrictions or implicit integration when compared to the uniform mesh case. However, larger timesteps (in the context of the RK4 time integration scheme) introduce numerical dissipation. The very small constant timestep series of tests minimizes this dissipation and allows for demonstrating the order of accuracy through convergence studies on an unsteady problem.

The constant CFL tests are carried out for all 5302 schemes over 32 cases defined by the parameter space, $\{\mathbf{N} \times \mathbf{l}\}$. The set of discretizations examined is $\mathbf{N} = \{51, 61, 71, 81, 91, 111, 131, 151\}$. The set of geometries is $\mathbf{l} = \{l_0, l_1, l_2, l_3\}$. To demonstrate the asymptotic stability of the schemes, the simulations are run for a full 1000 periods (until $t = 1000$). Fig. 20 shows the L_∞ error as a function of time for the $E2_1$ and $E4_1$ schemes in the appendix. The plot is broken to highlight the early time behavior, $t \in [0, 4]$, and the late time behavior, $t \in [996, 1000]$. The L_∞ error norm settles into a periodic steady state around $t \approx 2$ after brief transient. The curve continues, unchanged, for the duration of the simulation, demonstrating the time stability of the schemes. Integration in time is done using the standard RK4 method. The supplementary material records the maximum L_∞ over the time $t \in [980, 1000]$ for all schemes and cases. The constant Δt tests are carried out for all 5302 schemes over 24 cases defined by the parameter space, $\{\mathbf{N} \times \mathbf{l}\}$. The set of discretizations examined is $\mathbf{N} = \{51, 71, 91, 111, 131, 151\}$. The set of geometries is $\mathbf{l} = \{l_0, l_1, l_2, l_3\}$. Once again, the schemes are run until a time of $t = 1000$ and the maximum L_∞ over the time $t \in [980, 1000]$ is recorded for all schemes in the supplementary material. Fig. 21 shows the L_∞ error as a function of the number of points per diameter, N_p, for each geometry using the $E2_1$ and $E4_1$ schemes in the appendix. There is no apparent impact of the geometry on the behavior of $E2_1$, which converges with second order accuracy. There is some spread in the observed L_∞ for $E4_1$ at the finest two meshes but approximately 4th order convergence is still observed. This was achieved through the design principles and offline optimization process without any geometry corrections or in-situ procedures.

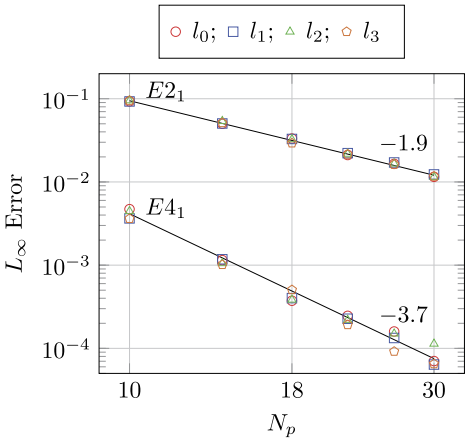

Fig. 21. Maximum error, L_∞, over $980 \leq t \leq 1000$ plotted against points-per-diameter, N_p, for $\Delta t = 8 \cdot 10^{-4}$ and the indicated schemes/geometries in solving equation (23). A line (on the log-log plot) is fit to the results for l_0 and annotated with its slope, indicating the convergence of the scheme. The 2nd order $E2_1$ is very insensitive to changes in geometry. The 4th order $E4_1$ shows some sensitivity but remains well-behaved over long times.

4.8. Two-dimensional nonlinear test: inviscid vortex / numerical reflection

In this section the two-dimensional compressible Euler equations are solved to examine the transport of an inviscid vortex through a domain and it's numerical collision with a supersonic outflow boundary. This collision with the outflow boundary generates very high frequency errors which propagate back into the domain with the potential to destabilize the simulation over long periods of time. The transport of an inviscid vortex through a periodic domain has been studied to quantify the impact of dissipation in upwind schemes [46]. The supersonic inflow/outflow case has been examined in Refs. [21] and [38] for relatively short times (1.5 flow through times based on the background streamwise velocity). In more recent work, this challenging case was run for 50 flow through times to highlight the stability of high order boundary schemes on uniform meshes [6]. In the present case, the conservative cut-cell schemes are used at the inflow/outflow boundaries and periodic boundary conditions are imposed in the cross-stream direction.

We adopt the notations from Ref. [38] to describe the analytic solution for a vortex of non-dimensional circulation, ϵ, centered at (x_c, y_c) at time $t = 0$ and propagating in the x direction. The solutions are repeated below for convenience:

$$\begin{aligned}
\frac{\rho}{\rho_\infty} &= \left(1 - \frac{\gamma - 1}{2} f^2\right)^{1/(\gamma-1)} , \\
\frac{u}{a_\infty} &= M_\infty + K\, y\, f , \\
\frac{v}{a_\infty} &= -K\, x\, f , \\
\frac{p}{p_\infty} &= \left(\frac{\rho}{\rho_\infty}\right)^{\gamma} ,
\end{aligned}$$

where $f = \frac{\epsilon}{2\pi} \exp((1 - K^2(x^2 + y^2))/2)$, M_∞ is the free stream Mach number, and $\gamma = c_p/c_v = 1.4$. As with the previous tests, time integration is carried out using a standard RK4 method. The tests were run on a computational domain of $x \in [x_l, x_r]$, $y \in [0, 10]$ with $(x_c, y_c) = (10, 5)$, $K = 1$, $\epsilon = 1.5$ and $M_\infty = 2.0$. The meshes are chosen such that the grid spacing in the x and y directions are equivalent. The x-direction mesh is once again described by Fig. 6, with uniform mesh spacing, h, throughout the domain excepting the near wall distances of $\psi_l h$ and $\psi_r h$. When $\psi_l = \psi_r = 1$, $x \in [0, 20]$. Fig. 22 shows contours of the pressure at $t = 0, 5, 10$. On the coarsest mesh, the vortex represented by these contours is resolved by 8 points. Constant CFL tests with $C = 0.5$, are carried out for all 5302 schemes over 6 cases defined by the parameter space, $\{\mathbf{N} \times \boldsymbol{\psi}\}$. The set of discretizations examined is $\mathbf{N} = \{51 \times 26, 101 \times 51, 201 \times 101\}$. The set of ψ values considered is $\boldsymbol{\psi} = \{\psi_l = \psi_r = 1, \psi_l = \psi_r = 0\}$. The reduced parameter space of the present test is due to the larger simulation cost compared to previous tests. The supplementary material records maximum L_∞ error in pressure (which occurs as the vortex leaves the domain through the outflow boundary) as well as the final L_∞ pressure error for all schemes and cases.

To illustrate the robustness of the conservative cut-cell schemes, Fig. 23 shows the evolution of L_∞ error of the pressure for $E2_1$ and $E4_1$ over the parameter space. The colored lines are used to indicate the degenerate mesh results. The light gray lines are used to indicate the results for the uniform mesh. Note that degenerate mesh results track very closely to the uniform mesh results. The spikes in error around $t = 10$ correspond to the vortex leaving the domain. The cut-cell schemes remain well behaved even when encountering this difficult non-linear event on a degenerate mesh.

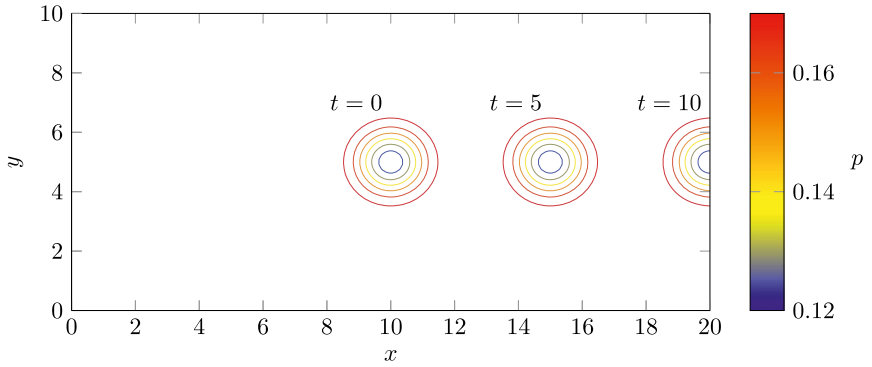

Fig. 22. Pressure contours of the solution at times $t = 0, 5, 10$ as the vortex moves from left to right. The vortex has completely left the domain by a time of $t = 12$.

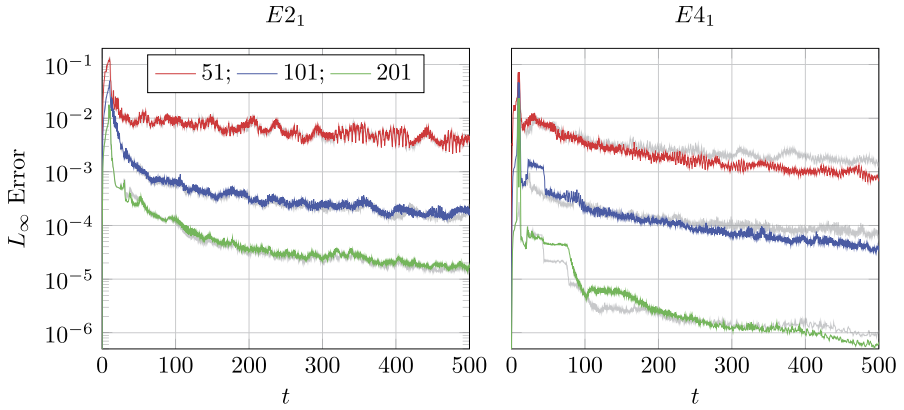

Fig. 23. L_∞ norm of error in pressure as a function of time, t, for the indicated conservative cut-cell scheme. The color lines correspond to the degenerate mesh simulations with $\psi_l = \psi_r = 0$. The light gray lines correspond to the uniform mesh simulation with $\psi_l = \psi_r = 1$. The uniform mesh simulations tend to closely follow the path of the degenerate mesh simulations. Large spikes in error occur around $t = 10$, when the vortex is leaving the domain.

5. Conclusions

The numerical challenges facing cut-cell methods stem from the highly irregular grids which naturally arise near the embedded boundaries of a non-Cartesian object embedded in a Cartesian mesh. This paper has focused exclusively on the stability issues that result from approximating derivatives on this irregular, and at times, degenerate mesh. This has been termed the "small-cell problem" in the literature. Previous attempts to solve this classic problem have employed either geometric manipulation algorithms to redistribute the irregularities or some kind of dynamic correction procedure which often results in numerical source terms being added to the governing equations. These approaches have been largely limited to 2nd order accuracy for unsteady problems.

In this paper, a fundamentally different approach, TEMO (truncation error matching and optimization), has been taken. This method is based on the intuitive design principles that a degenerate mesh ought to yield the same solution as the equivalent uniform mesh and that the assumptions of continuity, which form the foundation of finite differences, ought not be violated in the process. These two principles are all that is needed to construct the stable 4th, 6th and 8th order accurate approximations to parabolic and elliptic equations using both explicit and compact finite differences.

The construction of stable and conservative approximations to hyperbolic problems requires that these two design principles be supplemented with discrete conservation constraints and a novel non-linear optimization strategy. This process produced stable and conservative approximations of 2nd and 4th order.

The stability and accuracy of the schemes for parabolic, elliptic, and hyperbolic equations is demonstrated through a variety of tests and analysis. Asymptotic stability is explored through an eigenvalue analysis in which all schemes yielded stable eigenvalues. The accuracy and stability of the second derivative approximations was demonstrated by solving the Poisson equation and the unsteady heat equation for a variety of embedded wall distances as well as an embedded circle test case. The accuracy and long-time stability of the conservative cut-cell discretizations was assessed through two challenging hyperbolic problems. In the first, a varying coefficient advection equation is simulated for very long times with an interior, embedded circle serving as the inflow boundary. For the second test, the compressible Euler equations are solved (without any numerical dissipation) for a moving (supersonic) isentropic vortex. The embedded wall distances are again varied at

the supersonic inflow/outflow boundaries. In all tests considered, the schemes demonstrate the advertised order of accuracy and excellent stability properties over the whole range of geometries, from uniform to degenerate meshes.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the US Department of Energy through the Los Alamos National Laboratory. Los Alamos National Laboratory is operated by Triad National Security, LLC, for the National Nuclear Security Administration of U.S. Department of Energy (Contract No. 89233218CNA000001). Research presented in this article was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number ER20190227. Computational resources were provided by the LANL Institutional Computing (IC) Program.

Appendix A. Conservative cut-cell discretizations for first derivatives

The conservative cut-cell discretizations are given by the elements of the boundary matrix, $\mathbf{B}(\psi, \boldsymbol{\alpha})$. The general form of this matrix is too large to print but is available in the supplementary material. Tables A.3 and A.4 give specific instances of the $E2_1$ and $E4_1$ families that were found by the optimization procedure and used in the test cases in this paper.

Table A.3
Scheme $E2_1$: explicit conservative cut-cell boundary closures for first derivatives with 2nd order interior scheme.

$$\alpha_{0,0} = -2.769707867422022$$
$$\alpha_{0,1} = 4.394343202305503\psi$$
$$\alpha_{0,2} = 4.394343202305503 - 5.873906004650443\psi$$
$$\alpha_{0,3} = -1.47956280234494 + 1.3344902698063992\psi$$
$$\alpha_{0,4} = -0.145072532538541 + 0.145072532538541\psi$$
$$\alpha_{1,0} = (-2.769707867422022 + 0.3948457470551643\psi) / (1 + \psi)$$
$$\alpha_{1,1} = 1.150196406378214\psi$$
$$\alpha_{1,2} = \left(4.394343202305503 - 2.5501660272730517\psi^2 - 1.3203764394447342\psi\right) / (1 + \psi)$$
$$\alpha_{1,3} = -1.47956280234494 + 1.2548970883562962\psi$$
$$\alpha_{1,4} = -0.145072532538541 + 0.145072532538541\psi$$
$$\alpha_{2,0} = 0.17185962499047314\psi$$
$$\alpha_{2,1} = -1.187431060183429 + 0.34371181020248276\psi - 0.17185962499047316\psi^2$$
$$\alpha_{2,2} = 1.150196406378214 + 0.17185962499047314\psi^2 - 0.8064771563972674\psi$$
$$\alpha_{2,3} = 0.261900367793859 + 0.0662400072156786\psi$$
$$\alpha_{2,4} = -0.224665713988644 + 0.224665713988644\psi$$
$$\alpha_{3,0} = 0$$
$$\alpha_{3,1} = \left(-5.854968681519409 + 7.238307504438035\psi + 0.6392337735764906\psi^3 + 0.36877946535523365\psi^4 - 2.391352061850366\psi^2\right) / e_0$$
$$\alpha_{3,2} = \left(34.59906599295791 + 6.190942468885872\psi^2 - 1.6063383960657007\psi^4 - 2.6066103934877307\psi^3 - 30.765971981157694\psi\right) / e_0$$
$$\alpha_{3,3} = \left(-17.564906044558235 + 21.71492251331414\psi + 1.9177013207294775\psi^3 + 1.1063383960657005\psi^4 - 7.174056185551086\psi^2\right) / e_0$$
$$\alpha_{3,4} = \left(-11.17919126688027 + 1.8127419634055244\psi + 3.3744657785155803\psi^2 + 0.04967529918176233\psi^3 + 0.13122053464476655\psi^4\right) / e_0$$
$$e_0 = -34.068319896799366 + 18.102098935687152\psi + 1.9662274333304404\psi^2 + 1.3778181455165115\psi^3 + \psi^4$$

Table A.4
Scheme $E4_1$: explicit conservative cut-cell boundary closures for first derivatives with 4th order interior scheme.

$$\alpha_{0,0} = \left(-19.877376814069642 - 3.\psi^2 - 56.63213044220892\psi\right) / \left(6. + 11.\psi + 6.\psi^2 + \psi^3\right)$$
$$\alpha_{0,1} = 8.918251209379761\psi$$
$$\alpha_{0,2} = \left(8.918251209379761 - 27.73431643048422\psi^2 - 1.93868840703482\psi\right) / (1. + \psi)$$
$$\alpha_{0,3} = \left(-20.75475362813928 + 9.316065221104461\psi + 30.1934420351741\psi^2\right) / (2 + \psi)$$
$$\alpha_{0,4} = \left(18.75475362813928 - 12.856939616414 58\psi^2 - 11.816065221104461\psi\right) / (3. + \psi)$$
$$\alpha_{0,5} = -1.47956280234494 + 1.47956280234494\psi$$
$$\alpha_{0,6} = 0$$
$$\alpha_{1,0} = (-19.877376814069642 + 18.162985641122255\psi) / \left(6 + 11\psi + 6\psi^2 + \psi^3\right)$$
$$\alpha_{1,1} = -1.547601471175436\psi$$
$$\alpha_{1,2} = \left(8.918251209379761 + 3.163241611181368\psi^2 - 6.938688407034821\psi\right) / (1. + \psi)$$
$$\alpha_{1,3} = \left(-20.75475362813928 + 17.31606522110446\psi - 0.2041160064914882\psi^2\right) / (2. + \psi)$$
$$\alpha_{1,4} = \left(18.75475362813928 - 2.8910869358593843\psi^2 - 14.816065221104461\psi\right) / (3. + \psi)$$
$$\alpha_{1,5} = -1.47956280234494 + 1.47956280234494\psi$$
$$\alpha_{1,6} = 0$$
$$\alpha_{2,0} = -0.8770687218264502\psi$$

Table A.4 (*continued*)

$$\alpha_{2,1} = -0.07143296553947431 + 0.3905026400439473\psi + 1.6079593233484921\psi^2 + 0.8770687218264505\psi^3 + 0.14617812030440838\psi^4$$
$$\alpha_{2,2} = -1.547601471175436 + 1.946264327130012\psi - 0.4385343609132251\psi^4 - 2.1926718045661255\psi^3 - 2.6312061654793513\psi^2$$
$$\alpha_{2,3} = 2.571402206763154 + 1.3156030827396754\psi^2 + 1.7541374436529005\psi^3 + 0.4385343609132251\psi^4 - 2.919396490695018\psi$$
$$\alpha_{2,4} = -1.2142681378421027 + 1.946264327130012\psi - 0.14617812030440838\psi^4 - 0.43853436091322523\psi^3 - 0.29235624060881676\psi^2$$
$$\alpha_{2,5} = 0.261900367793859 - 0.486566081782503\psi$$
$$\alpha_{2,6} = 0$$
$$\alpha_{3,0} = 0.6961450379207412\psi$$
$$\alpha_{3,1} = -0.8770687218264502 + 0.42462426012860605\psi - 0.11602417298679019\psi^4 - 0.6961450379207411\psi^3 - 1.2762659028546923\psi^2$$
$$\alpha_{3,2} = 2.9502758399838243 + 2.0884351137622237\psi^2 + 1.740362594801853\psi^3 + 0.3480725189603706\psi^4 - 4.258411478208746\psi$$
$$\alpha_{3,3} = -4.863749475004126 + 5.825952932341509\psi - 0.34807251896037056\psi^4 - 1.3922900758414822\psi^3 - 1.0442175568811118\psi^2$$
$$\alpha_{3,4} = 3.1602806033739372 + 0.2320483459735804\psi^2 + 0.34807251896037056\psi^3 + 0.1160241729867902\psi^4 - 3.135082908265525\psi$$
$$\alpha_{3,5} = -0.145072532538541 + 0.2221064420947713 7\psi$$
$$\alpha_{3,6} = -0.224665713988644 + 0.224665713988644\psi$$
$$\alpha_{4,0} = 0$$
$$\alpha_{4,1} = \big(-17.540945247724775 + 136.8872106322367\psi + 332.657888919291\psi^2 + 13.474345105867513\psi^3 - 0.6778464862382794\psi^9$$
$$- 10.817315419280217\psi^8 - 66.55351613252738\psi^7 - 174.35167249687643\psi^6 - 166.11737202522914\psi^5 - 46.96077684951908\psi^4\big)/e_0$$
$$\alpha_{4,2} = \big(64.02319500448931 + 158.33667525651526\psi^4 + 653.4217049549081\psi^5 + 703.0444457818155\psi^6 + 273.9984433548364\psi^7$$
$$+ 45.79767101871374\psi^8 + 2.9639178813369282\psi^9 - 66.08934182445557\psi^3 - 1258.2734987586696\psi^2 - 533.2984309736585\psi\big)/e_0$$
$$\alpha_{4,3} = \big(-72.28424196120802 + 776.9504344019497\psi + 1651.5251149385265\psi^2 + 62.84926186977691\psi^3 - 5.077206662964922\psi^9$$
$$- 75.35086321538616\psi^8 - 433.28173917411794\psi^7 - 1077.3359887996094\psi^6 - 971.2160503086135\psi^5 - 208.1769746550032\psi^4\big)/e_0$$
$$\alpha_{4,4} = \big(45.918893441692546 + 570.1283631129575\psi^5 + 718.2208287916981\psi^6 + 308.68564685840187\psi^7 + 57.939717726678 19\psi^8$$
$$+ 4.2265775632559865\psi^9 - 55.85254863749133\psi^4 - 227.1961303834413\psi^3 - 978.96820052106\psi^2 - 443.10314795269153\psi\big)/e_0$$
$$\alpha_{4,5} = \big(-18.175858040589983 + 41.99911392203342\psi + 208.19563395503837\psi^2 + 262.53615381792383\psi^3 + 204.37907808912647\psi^4$$
$$- 1.6879742317735251\psi^9 - 19.930952785651765\psi^8 - 89.22165019129679\psi^7 - 170.8779037507821\psi^6 - 65.81738721737831\psi^5\big)/e_0$$
$$\alpha_{4,6} = \big(-1.9410431966590622 + 20.56481997013003\psi + 44.863061466873575\psi^2 + 1.3002904737542476\psi^6 + 6.3728152847038775\psi^7$$
$$+ 2.3617426749262207\psi^8 + 0.25253193638381144\psi^9 - 20.39925851664461\psi^5 - 51.72545320362801\psi^4 - 45.5742885856713\psi^3\big)/e_0$$
$$e_0 = -25.197256738504368 - 37.8864504890495\psi + 164.96997270972506\psi^2 + 200.2939631081134\psi^3 + 133.31412637240055\psi^4$$
$$+ 56.108852223817124\psi^5 + 26.024791923333847\psi^6 + 8.469381240138295\psi^7 + \psi^8$$

Table B.5
Each scheme (or family of schemes) is associated with an ascii text file for the corresponding boundary conditions as well a sqlite database recording the performance of the schemes on the appropriate tests in Section 4.

Scheme	Dirichlet/Floating	Neumann	Database
$E2_1$	E2_1D.txt		E2_1.db
$E4_1$	E4_1D.txt		E4_1.db
$E2_2$	E2_2D.txt	E2_2N.txt	E2_2.db
$E4_2$	E4_2D.txt	E4_2N.txt	E4_2.db
$E6_2$	E6_2D.txt	E6_2N.txt	E6_2.db
$E8_2$	E8_2D.txt	E8_2N.txt	E8_2.db
$T4_2$	T4_2D.txt	T4_2N.txt	T4_2.db
$T6_2$	T6_2D.txt	T6_2N.txt	T6_2.db
$T8_2$	T8_2D.txt	T8_2N.txt	T8_2.db

Appendix B. Supplementary material

Included with this manuscript are two types of supplementary data. The first are ascii text files containing the α_{ij} of the cut-cell boundary schemes for both first and second derivatives that can be easily parsed and implemented in a particular programming language. The second are sqlite databases recording the performance of the schemes over the relevant parameter spaces for the various tests listed in Section 4. The relationship between the schemes, text files and databases are given in Table B.5.

The text files associated with the second derivative operators contain the coefficients as functions of ψ in a format that is easy to copy/paste or parse. The text files associated with the first derivative operators are the general coefficients written in terms of their free parameters, $\boldsymbol{\alpha}$, and the cut-cell distance, ψ. The general forms are too large to include directly in the text but can be easily parsed from the included ".txt" files. Values for the free parameters, $\boldsymbol{\alpha}$ are given the "nbs" table in the databases. The given id is unique to each scheme. Tables A.3 and A.4 use schemes with an id of 1 from the appropriate table to generate particular instances of the $E2_1$ and $E4_1$ families. An example of using python to read a row of free parameters from the database and generate boundary scheme coefficients for a particular ψ is shown in Listing 1.

Table B.6 lists the relationship between the stability and accuracy tests for the second derivative operators in Section 4 and the corresponding sqlite tables that contain the results of the tests over the appropriate parameter space. The tables are normalized, such that the parameter space id, "pid", is used as a foreign key in the results tables. The "eigenvalues" table

Table B.6
Relationship between the sqlite tables recording the results and parameter space for the indicated parabolic and elliptic equation tests in Section 4.

Test	Results	Parameters
Section 4.1: eigenvalues	eigenvalues	eigen_p
Section 4.2: embedded planes	heat_planes	heat_planes_p
Section 4.3: embedded circle	heat_circle	heat_circle_p
Section 4.4: embedded planes	elliptic_planes	elliptic_planes_p
Section 4.5: embedded circle	elliptic_circle	elliptic_circle_p

Table B.7
Relationship between the sqlite tables recording the results and parameter space for the indicated hyperbolic tests in Section 4.

Test	Results	Parameters
Section 4.6: eigenvalues	eigenvalues	eigen_p
Section 4.7: constant CFL	scalar_wave_cfl	scalar_wave_cfl_p
Section 4.7: constant Δt	scalar_wave_dt	scalar_wave_dt_p
Section 4.8: vortex advection	vortex_cfl	vortex_cfl_p

records $\max_\lambda \mathrm{re}(\lambda)$ as "lambda". The other tables record the L_∞ error norm for the indicated simulation (the maximum error over the time interval specified in the appropriate section) as "Linf".

Table B.7 lists the relationship between the stability and accuracy tests for the first derivative operators in Section 4 and the corresponding sqlite tables that contain the results of the tests over the appropriate parameter space. The tables are normalized, such that the scheme id, "id", and parameter space id, "pid", are used as a foreign key in the results tables. The data is encoded as for the second derivative tests with the exception of the vortex advection tests. The "vortex_cfl" table records the maximum L_∞ error in the pressure as "Linf" and the final observed L_∞ error in the pressure as "Linf_final".

```
from fractions import Fraction as F
import sqlite3 as db

# connect to database and read free parameters for scheme with id=1
scheme_id = 1
conn = db.connect("E2_1.db")
conn.row_factory = db.Row
cur = conn.cursor()
cur.execute("select * from nbs where id=?", (scheme_id, ))
row = cur.fetchone()

# choose a particular psi value to evaluate all coefficients
params = {"psi" : 0}

# add free parameters from database to params dictionary
# this will serve as the local environment for eval
for k, v in zip(row.keys()[1:], row[1:]):
    params[k] = F(v)  # use rationals to avoid roundoff errors


with open("E2_1D.txt") as f:
    for line in f:
        k, v = line.split('=')
        params[k] = eval(v.strip(), globals(), params)

# print coefficients
for k, v in sorted(params.items()):
    print("{:10s} = {:10f}".format(k, float(v)))
```

Listing 1: Example python script to use free parameters from database to generate coefficients.

Description of E2_1D.txt[1]

Ascii text description of the 2nd order $E2_1$ family of cut-cell schemes for first derivative approximations as functions of ψ. The free parameters are listed as `alphau_0_2`, `alphau_0_3`, `alphau_1_2`, and `alphau_1_3`, corresponding to the column names of the `nbs` table in the associated database `E2_1.db`

Description of E2_1.db[1]

Sqlite3 database for the $E2_1$ family of cut-cell schemes containing the tables listed in Table B.7. The scheme used for the paper and given in Table A.3 corresponds to the choice free parameters recorded in the `nbs` table with `id=1`:

```
alphau_0_2=-1.47956280234494
alphau_0_3=-0.145072532538541
alphau_1_2=0.261900367793859
alphau_1_3=-0.224665713988644
```

Description of E4_1D.txt[1]

Ascii text description of the 4^{th} order $E4_1$ family of cut-cell schemes for first derivative approximations as functions of ψ. The free parameters are listed as `alphau_0_4`, `alphau_1_4`, `alphau_2_4`, and `alphau_2_5`, corresponding to the column names of the `nbs` table in the associated database `E4_1.db`

Description of E4_1.db[1]

Sqlite3 database for the $E4_1$ family of cut-cell schemes containing the tables listed in Table B.7. The scheme used for the paper and given in Table A.4 corresponds to the choice free parameters recorded in the `nbs` table with `id=1`:

```
alphau_0_4=-0.0228413686378388
alphau_1_4=-0.0582315114712024
alphau_2_4=0.0258161783004396
alphau_2_5=-0.0111107179663226
```

Description of E2_2D.txt[1]

Ascii text description of the 2nd order $E2_2^d$ family of cut-cell schemes for second derivative approximations as functions of ψ when Dirichlet boundary conditions are required.

Description of E2_2N.txt[1]

Ascii text description of the 2nd order $E2_2^n$ family of cut-cell schemes for second derivative approximations as functions of ψ when Neumann boundary conditions are required.

Description of E2_2.db[1]

Sqlite3 database for the $E2_2$ family of cut-cell schemes containing the result tables listed in Table B.6.

Description of E4_2D.txt[1]

Ascii text description of the 4th order $E4_2^d$ family of cut-cell schemes for second derivative approximations as functions of ψ when Dirichlet boundary conditions are required.

Description of E4_2N.txt[1]

Ascii text description of the 4th order $E4_2^n$ family of cut-cell schemes for second derivative approximations as functions of ψ when Neumann boundary conditions are required.

Description of E4_2.db[1]

Sqlite3 database for the $E4_2$ family of cut-cell schemes containing the result tables listed in Table B.6.

[1] Please see https://data.mendeley.com/datasets/x5z98g7mz7/1.

Description of E6_2D.txt[1]

Ascii text description of the 6th order $E6_2^d$ family of cut-cell schemes for second derivative approximations as functions of ψ when Dirichlet boundary conditions are required.

Description of E6_2N.txt[1]

Ascii text description of the 6th order $E6_2^n$ family of cut-cell schemes for second derivative approximations as functions of ψ when Neumann boundary conditions are required.

Description of E6_2.db[1]

Sqlite3 database for the $E6_2$ family of cut-cell schemes containing the result tables listed in Table B.6.

Description of E8_2D.txt[1]

Ascii text description of the 8th order $E8_2^d$ family of cut-cell schemes for second derivative approximations as functions of ψ when Dirichlet boundary conditions are required.

Description of E8_2N.txt[1]

Ascii text description of the 8th order $E8_2^n$ family of cut-cell schemes for second derivative approximations as functions of ψ when Neumann boundary conditions are required.

Description of E8_2.db[1]

Sqlite3 database for the $E8_2$ family of cut-cell schemes containing the result tables listed in Table B.6.

Description of T4_2D.txt[1]

Ascii text description of the 4th order compact $T4_2^d$ family of cut-cell schemes for second derivative approximations as functions of ψ when Dirichlet boundary conditions are required.

Description of T4_2N.txt[1]

Ascii text description of the 4th order compact $T4_2^n$ family of cut-cell schemes for second derivative approximations as functions of ψ when Neumann boundary conditions are required.

Description of T4_2.db[1]

Sqlite3 database for the $T4_2$ family of cut-cell schemes containing the result tables listed in Table B.6.

Description of T6_2D.txt[1]

Ascii text description of the 6th order compact $T6_2^d$ family of cut-cell schemes for second derivative approximations as functions of ψ when Dirichlet boundary conditions are required.

Description of T6_2N.txt[1]

Ascii text description of the 6th order compact $T6_2^n$ family of cut-cell schemes for second derivative approximations as functions of ψ when Neumann boundary conditions are required.

Description of T6_2.db[1]

Sqlite3 database for the $T6_2$ family of cut-cell schemes containing the result tables listed in Table B.6.

Description of T8_2D.txt[1]

Ascii text description of the 8th order compact $T8_2^d$ family of cut-cell schemes for second derivative approximations as functions of ψ when Dirichlet boundary conditions are required.

Description of T8_2N.txt[1]

Ascii text description of the 8th order compact $T8_2^n$ family of cut-cell schemes for second derivative approximations as functions of ψ when Neumann boundary conditions are required.

Description of T8_2.db[1]

Sqlite3 database for the $T8_2$ family of cut-cell schemes containing the result tables listed in Table B.6.

References

[1] A. Almgren, J. Bell, P. Colella, T. Marthaler, A Cartesian grid projection method for the incompressible Euler equations in complex geometries, SIAM J. Sci. Comput. 18 (1997) 1289–1309.
[2] J.D. Anderson, Computational Fluid Dynamics: The Basics with Applications, McGraw-Hill Series in Mechanical Engineering, McGraw-Hill, 1995.
[3] P. Barton, B. Obadia, D. Drikakis, A conservative level-set based method for compressible solid/fluid problems on fixed grids, J. Comput. Phys. 230 (2011) 7867–7890, https://doi.org/10.1016/j.jcp.2011.07.008.
[4] M. Berger, R. LeVeque, An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries, AIAA Paper 89-1930, 1989.
[5] P. Brady, M. Herrmann, J. Lopez, Code verification for finite volume multiphase scalar equations using the method of manufactured solutions, J. Comput. Phys. 231 (2012) 2924–2944, https://doi.org/10.1016/j.jcp.2011.12.040, http://www.sciencedirect.com/science/article/pii/S0021999112000083.
[6] P. Brady, D. Livescu, High-order, stable, and conservative boundary schemes for central and compact finite differences, Comput. Fluids 183 (2019) 84–101.
[7] P.T. Brady, D. Livescu, Stable, high-order and conservative cut-cell methods, AIAA Scitech 2019 Forum, AIAA 2019–1991, https://doi.org/10.2514/6.2019-1991, 2019.
[8] C. Brehm, M. Barad, C. Kiris, Development of immersed boundary computational aeroacoustic prediction capabilities for open-rotor noise, J. Comput. Phys. 388 (2019) 690–716, https://doi.org/10.1016/j.jcp.2019.02.011.
[9] C. Brehm, H. Fasel, A novel concept for the design of immersed interface methods, J. Comput. Phys. 242 (2013) 234–267, https://doi.org/10.1016/j.jcp.2013.01.027.
[10] C. Brehm, C. Hader, H. Fasel, A locally stabilized immersed boundary method for the compressible Navier–Stokes equations, J. Comput. Phys. 295 (2015) 475–504, https://doi.org/10.1016/j.jcp.2015.04.023.
[11] E. Burman, S. Claus, P. Hansbo, M.G. Larson, A. Massing, Cuffem: discretizing geometry and partial differential equations, Int. J. Numer. Methods Eng. 104 (2015) 472–501, https://doi.org/10.1002/nme.4823.
[12] M.H. Carpenter, D. Gottlieb, S. Abarbanel, The stability of numerical boundary treatments for compact high-order finite-difference schemes, J. Comput. Phys. 108 (1993) 272–295, https://doi.org/10.1006/jcph.1993.1182.
[13] D. Cecere, E. Giacomazzi, An immersed volume method for Large Eddy Simulation of compressible flows using a staggered-grid approach, Comput. Methods Appl. Mech. Eng. 280 (2014) 1–27, https://doi.org/10.1016/j.cma.2014.07.018.
[14] Y-L. Chiang, B. van Leer, K. Powell, Simulation of unsteady inviscid flow on an adaptively refined Cartesian grid, AIAA Paper 92-0443, 1992.
[15] M.H. Chung, A level set approach for computing solutions to inviscid compressible flow with moving solid boundary using fixed Cartesian grids, Int. J. Numer. Methods Fluids 36 (2001) 373–389, https://doi.org/10.1002/fld.32.
[16] M.H. Chung, Cartesian cut cell approach for simulating incompressible flows with rigid bodies of arbitrary shape, Comput. Fluids 35 (2006) 607–623, https://doi.org/10.1016/j.compfluid.2005.04.005.
[17] D. Clarke, H. Hassan, M. Salas, Euler calculations for multielement airfoils using Cartesian grids, AIAA J. 24 (1986) 353–358.
[18] R. Crockett, P. Colella, D. Graves, A Cartesian grid embedded boundary method for solving the Poisson and heat equations with discontinuous coefficients in three dimensions, J. Comput. Phys. 230 (2011) 2451–2469, https://doi.org/10.1016/j.jcp.2010.12.017.
[19] B. Gustafsson, The convergence rate for difference approximations to mixed initial boundary value problems, Math. Comput. 29 (1975) 396–406.
[20] D. Hartmann, M. Meinke, W. Schröder, A strictly conservative Cartesian cut-cell method for compressible viscous flows on adaptive grids, Comput. Methods Appl. Mech. Eng. 200 (2011) 1038–1052, https://doi.org/10.1016/j.cma.2010.05.015.
[21] J.W. Kim, Optimised boundary compact finite difference schemes for computational aeroacoustics, J. Comput. Phys. 225 (2007) 995–1019, https://doi.org/10.1016/j.jcp.2007.01.008.
[22] M.P. Kirkpatrick, S.W. Armfield, J.H. Kent, A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional Cartesian grid, J. Comput. Phys. 184 (2003) 1–36.
[23] S.K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1992) 16–42, https://doi.org/10.1016/0021-9991(92)90324-R.
[24] P. McCorquodale, P. Colella, H. Johansen, A Cartesian grid embedded boundary method for the heat equation on irregular domains, J. Comput. Phys. 173 (2001) 620–635, https://doi.org/10.1006/jcph.2001.6900.
[25] M. Meyer, A. Devesa, S. Hickel, X. Hu, N. Adams, A conservative immersed interface method for Large-Eddy Simulation of incompressible flows, J. Comput. Phys. 229 (2010) 6300–6317, https://doi.org/10.1016/j.jcp.2010.04.040.
[26] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261, https://doi.org/10.1146/annurev.fluid.37.061903.175743.
[27] L. Monasse, V. Daru, C. Mariotti, S. Piperno, C. Tenaud, A conservative coupling algorithm between a compressible flow and a rigid body using an embedded boundary method, J. Comput. Phys. 231 (2012) 2977–2994, https://doi.org/10.1016/j.jcp.2012.01.002.
[28] B. Muralidharan, S. Menon, A high-order adaptive Cartesian cut-cell method for simulation of compressible viscous flow over immersed bodies, J. Comput. Phys. 321 (2016) 342–368, https://doi.org/10.1016/j.jcp.2016.05.050.
[29] R. Pember, J. Bell, An adaptive Cartesian grid method for unsteady compressible flow in irregular regions, J. Comput. Phys. 120 (1995) 278–304.
[30] J. Quirk, An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies, Comput. Fluids 23 (1994) 125–142.
[31] J. Roache, Code verification by the method of manufactured solutions, J. Fluids Eng. 124 (2002) 4–10.
[32] L. Schneiders, C. Günther, M. Meinke, W. Schröder, An efficient conservative cut-cell method for rigid bodies interacting with viscous compressible flows, J. Comput. Phys. 311 (2016) 62–86, https://doi.org/10.1016/j.jcp.2016.01.026.
[33] L. Schneiders, D. Hartmann, M. Meinke, W. Schröder, An accurate moving boundary formulation in cut-cell methods, J. Comput. Phys. 235 (2013) 786–809, https://doi.org/10.1016/j.jcp.2012.09.038.
[34] P. Schwartz, M. Barad, P. Colella, T. Ligocki, A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions, J. Comput. Phys. 211 (2006) 531–550, https://doi.org/10.1016/j.jcp.2005.06.010.

[35] J.H. Seo, R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, J. Comput. Phys. 230 (2011) 7347–7363, https://doi.org/10.1016/j.jcp.2011.06.003.
[36] N. Sharan, P.T. Brady, D. Livescu, Stable and conservative boundary treatment for difference methods, with application to cut-cell discretizations, AIAA Scitech 2020 Forum, AIAA 2020–0807, https://doi.org/10.2514/6.2020-0807, 2020.
[37] G. Strang, Linear Algebra and Its Applications, Thomson Brooks/Cole, 2006.
[38] J.M. Turner, S. Haeri, J.W. Kim, Improving the boundary efficiency of a compact finite difference scheme through optimising its composite template, Comput. Fluids 138 (2016) 9–25, https://doi.org/10.1016/j.compfluid.2016.08.007.
[39] H. Udaykumar, H.C. Kan, W. Shyy, R. Tran-Son-Tay, Multiphase dynamics in arbitrary geometries on fixed Cartesian grids, J. Comput. Phys. 137 (1997) 366–405, https://doi.org/10.1006/jcph.1997.5805.
[40] H. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, J. Comput. Phys. 174 (2001) 345–380, https://doi.org/10.1006/jcph.2001.6916.
[41] H. Udaykumar, W. Shyy, M. Rao, ELAFINT: a mixed Eulerian–Lagrangian method for fluid flows with complex and moving boundaries, Int. J. Numer. Methods Fluids 22 (1996) 691–712.
[42] H.S. Udaykumar, R. Mittal, P. Rampunggoon, Interface tracking finite volume method for complex solid-fluid interactions on fixed meshes, Commun. Numer. Methods Eng. 18 (2001) 89–97, https://doi.org/10.1002/cnm.468.
[43] G. Yang, D. Causon, D. Ingram, Calculation of compressible flows about complex moving geometries using a three-dimensional Cartesian cut cell method, Int. J. Numer. Methods Fluids 33 (2000) 1121–1151.
[44] G. Yang, D. Causon, D. Ingram, R. Saunders, P. Batten, A Cartesian cut cell method for compressible flows. A. Static body problems, Aerosp. Eng. Technol. 101 (1997) 47–56.
[45] T. Ye, R. Mittal, H. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (1999) 209–240, https://doi.org/10.1006/jcph.1999.6356.
[46] H.C. Yee, N.D. Sandham, M.J. Djomehri, Low-dissipative high-order shock-capturing methods using characteristic-based filters, J. Comput. Phys. 150 (1999) 199–238.