Eleventh International Conference on Computational Fluid Dynamics (ICCFD11), Maui, Hawaii, USA, July 11-15, 2022 **ICCFD11-2022-2301**

High-Order Cut-Cell Methods for High-Fidelity Flow Simulations

P. T. Brady*, D. Livescu*
Corresponding author: ptb@lanl.gov

* Los Alamos National Laboratory, USA

Abstract: Cut-cell methods allow for the use of Cartesian meshes to resolve phenomena occurring in complex geometries. The advantage of cut-cell methods over approaches requiring body fitted meshes (either structured or unstructured) is that the former do not require a costly mesh generation step or complicated code infrastructure. However, cut-cell methods have been typically limited to low orders of accuracy. In the present work, we extend our previous efforts to expand our unique high-order, stable and conservative cut-cell method to three-dimensional geometries by exploring the impact of interpolation choices. To test their efficacy for flow problems, the schemes are tested on a hyperbolic system with embedded objects.

Keywords: Cut cell, Computational Fluid Dynamics, Embedded Boundaries.

1 Introduction

The cut-cell method [9] allows for the solution of partial differential equations (PDEs) defined on complicated domains to be computed on simple Cartesian meshes. From a software implementation perspective, a Cartesian solver will generally have a simpler architecture compared to an unstructured mesh solver and will access data in regular, predictable patterns leading to more vectorization and accelerator offload opportunities, making this a desirable approach. The goal of the present approach is to solve the Navier-Stokes equations, 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, as is done in the immersed boundary method [12], the cut-cell approach modifies the discrete derivative operators and imposes boundary conditions directly on Γ_s. The governing equations are not solved within the solid object, Ω_s, indicated by the grayed out portion of the domain in Fig. 1.

The simplicity of cut-cell type methods has attracted the attention and effort of a number of researchers for many years (see [11] for a review). In theory, for compatible resolution requirements with Cartesian and unstructured representations, cut-cell methods (also referred to as embedded boundary or Cartesian grid methods) would always be preferred due to the underlying simplicity and efficiency of a Cartesian solver. However, this has not been the case. It is our conjecture that this is largely due to the limited accuracy of typical cut-cell methods and the other manifestations of the "small-cell problem". In general, the stiffness of the discrete problem becomes very large when the a fluid cell is cut by the embedded object such that the portion of the cell in Ω_f is very small compared to an uncut cell. The solutions to the severe numerical challenges imposed by this small-cell problem (see [3] for a list) lead to significant modifications of both the discrete algorithms and the physical equations. The discrete algorithms are modified by requiring significant extra procedures to evaluate derivatives near the boundary since a straightforward evaluation leads to instabilities. The physical equations are typically modified by requiring some sort of stabilization procedure which manifests itself as a source term in the governing equations (even if not explicitly written as such).

In recent work [2, 6, 3], we have demonstrated an approach to cut-cell methods that uses an offline optimization procedure to locate stable discretizations, rather than relying on ad-hoc stabilization procedures. This was accomplished by pursuing a finite-differences based approach (as was done in [7]) and framing the problem of locating stable cut-cell discretizations as a parameterization of locating suitable numerical boundary stencils [8]. This procedure was successful and allowed for the development of discretizations of up to 8^{th} order for parabolic equations and 4^{th} order for hyperbolic systems while using centered differencing in the interior. To the best of our knowledge, these cut-cell stencils were the highest order reported and were the first that could be reported as simple closed-form coefficients that didn't rely on any in-situ procedures for their evaluation.

The success of the optimization approach relied on using a broad swath of simple test problems such that the tests, in an imprecise sense, "spanned" the space of behaviors that would be encountered in a high-fidelity simulation of fluid flow. The method can be extended such that if situations are encountered in a flow calculation that trigger numerical instabilities, an appropriate test problem can be added to the suite of problems and the optimization procedure rerun. However, this is somewhat of a brute force approach and has led to the exploration of a complementary strategy based on energy stability [13, 14, 16, 15]. In this strategy, we again take the starting point of a finite-difference formulation and view the cut-cell approach as one of finding suitable numerical boundary stencils. However, rather than proceeding directly with optimization, the cut-cell stencils are first constrained to meet the the requirements of energy stability for linear hyperbolic problems. For higher order stencils, the constraints cannot be solved analytically and so one must still pursue an optimization/root-finding strategy, but it is a more limited optimization scope compared to [3], with results that are broadly applicable to hyperbolic problems.

Both strategies have yielded successful approaches to high-order cut-cell methods and have been demonstrated on a variety of test problems, including the compressible Euler equations in 3D [15]. In the next section, we discuss the remaining challenges for these methods to be applied to the compressible Navier-Stokes equations with general boundary conditions as well as some preliminary results.

2 Problem Statement

We will be making use of the notations introduced in [3] to describe the present cut-cell approach. By maintaining a finite difference representation, the method does not require expensive computational geometry utilities to determine arbitrary volume/volume intersections of the embedded object with the Cartesian mesh. Instead, all that is required is a simple ray-tracing calculation along mesh lines to locate intersection points. Figure 2 shows an example of the construction of set of points, $\mathcal{R}^x$, associated with the intersection of rays in the x-direction described by various (j, k) coordinates, $\mathcal{R}^x_{jk}$, for 3 embedded objects arrayed on a domain. These intersection calculations are one dimensional calculations and are therefore relatively fast compared to the

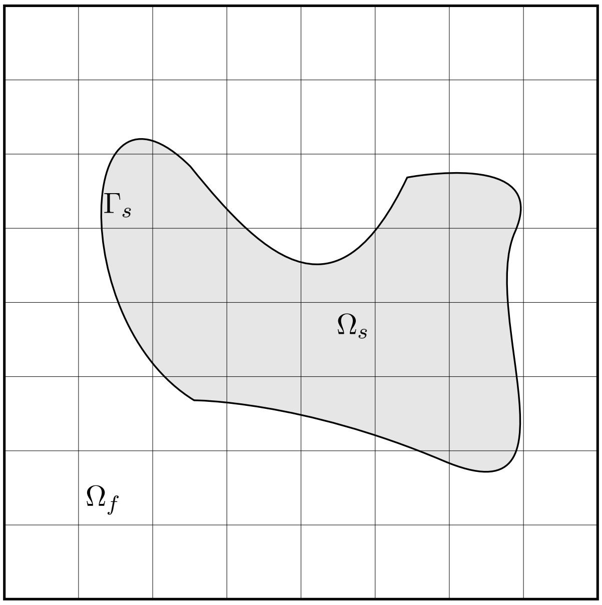


Figure 1: Schematic of solid object, bounded by Γ_s, embedded in a fluid domain, Ω_f.

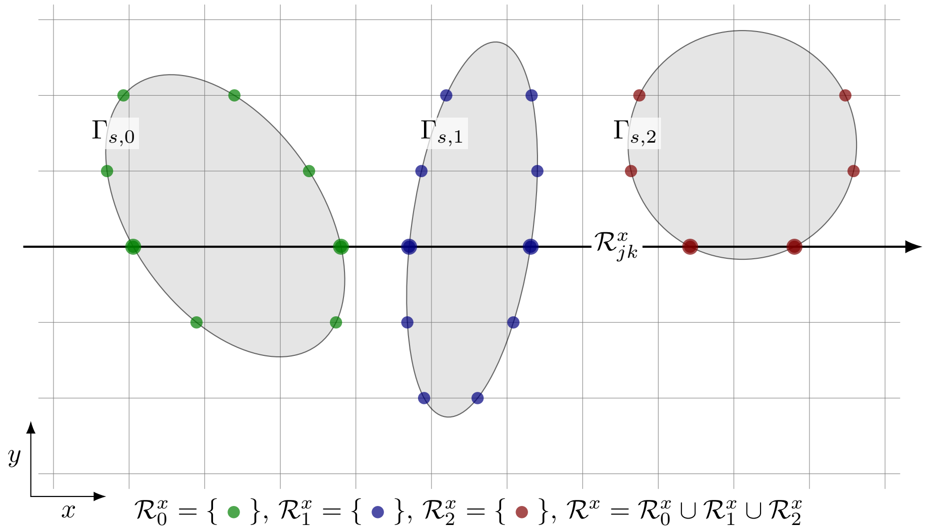
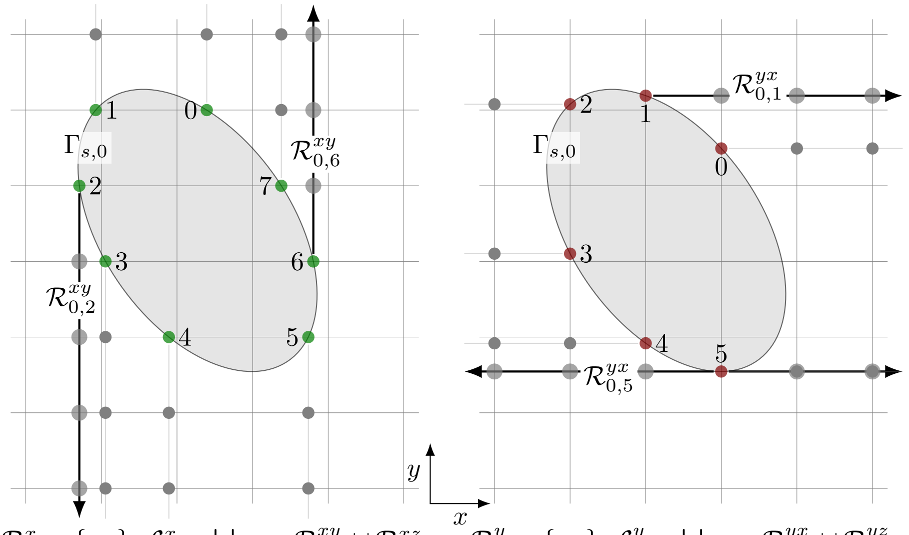

$\Gamma_{s,0}$
$\Gamma_{s,1}$
$\Gamma_{s,2}$
$\mathcal{R}^x_{jk}$
y
x
$\mathcal{R}^x_0 = \{$ ● $\}$, $\mathcal{R}^x_1 = \{$ ● $\}$, $\mathcal{R}^x_2 = \{$ ● $\}$, $\mathcal{R}^x = \mathcal{R}^x_0 \cup \mathcal{R}^x_1 \cup \mathcal{R}^x_2$

Figure 2: Taken from [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.



$\mathcal{R}^x_0 = \{$ ● $\}$, $\mathcal{L}^x_0 = \bigcup_{p\in\mathcal{P}^x_0} \mathcal{R}^{xy}_{0,p} \cup \mathcal{R}^{xz}_{0,p}$ $\mathcal{R}^y_0 = \{$ ● $\}$, $\mathcal{L}^y_0 = \bigcup_{p\in\mathcal{P}^y_0} \mathcal{R}^{yx}_{0,p} \cup \mathcal{R}^{yz}_{0,p}$

Figure 3: Taken from [3], 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 p^{th} point in $\mathcal{R}^x_m$ and extending in the $\pm y$ and $\pm z$ directions as appropriate.

three dimensional volume intersection calculations required for finite volume cut-cell schemes. The construction of $\mathcal{R}^y$ and $\mathcal{R}^z$ follow similarly. Grouping the intersection points by mesh-line ray-direction works well with the dimensionally split nature of finite differences and allows one to directly frame the problem of locating stable cut-cell discretizations as a parameterized numerical boundary stencil optimization problem.

Knowledge of the points in $\mathcal{R}^{x/y/z}$ is all that is needed when Dirichlet boundary conditions are associated with each embedded object. However, when solving the Navier-Stokes equations in 3D, a variety of other situations needs to be handled. These include: 1) cross derivatives at the object boundary, 2) Neumann conditions, and 3) floating values for which no physical boundary condition may be applied. Handling these situations requires interpolation.

For example, consider the points in $\mathcal{R}^x$. They are aligned with the mesh-lines in the x-direction but not with the y or z directions. Computing derivatives in those directions at the embedded object requires that we first construct some auxiliary lines of data in those directions, interpolate field data to the appropriate locations on those lines and apply the derivative stencil. Figure 3 highlights some of the considerations involved in this procedure, where a dimensionally split approach is being pursued rather than constructing a polynomial surface and performing a least-squares fit.

An initial exploration of these ideas was done within the context of an energy stable approach [15] in simulating the flow past a cylinder. There it was observed that anything higher than linear interpolation led to instabilities. Similar results were found for the approach in [3]. This highlights the need for further optimization. Specifically, the interpolation stencils must be included in the optimization procedure.

While it is possible to stick with the previously optimized derivative stencils and simply add in interpolation stencils, we decided to instead pursue the development of a unified framework that started with the interpolation stencils, and then to derive the derivative stencils from those. This puts all the free parameters in a consistent system. The procedure does require new tests to

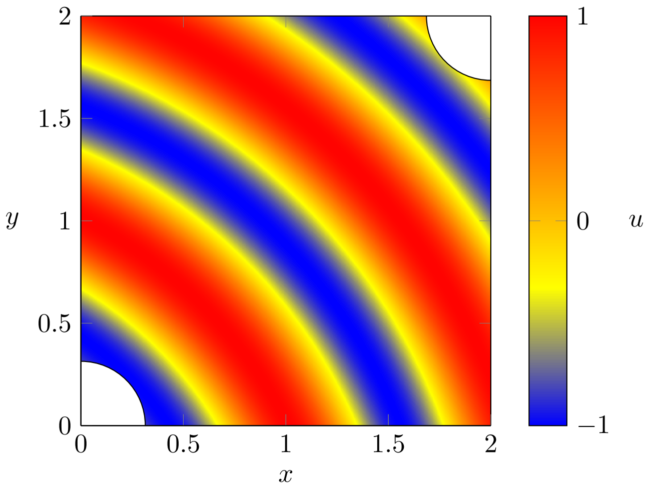


Figure 4: Initial conditions for the variable coefficient advection test. The circular pulses migrate from bottom-left to top-right, providing an opportunity to test the interpolation/derivative stencils needed for outflow at the top-right embedded object.

be added to the software optimization infrastructure, and so a new finite-difference optimizer [4] was developed on top of nlopt [10] and legion [1] that utilizes the newly developed cut-cell solver [5].

As an example, one new test added to the optimizer is a variable coefficient advection equation with embedded circles at the inflow and outflow walls. The governing equation for the system is:

$$\frac{\partial u}{\partial t} + \nabla G \cdot \nabla u = 0\,, \qquad L_x = L_y = 2\,, \tag{1}$$

with

$$G(x, y) = \sqrt{(x+1)^2 + (y+1)^2} + 1\,, \tag{2}$$

$$u(x, y, 0) = \sin(2\pi G)\,, \tag{3}$$

and a solution given by:

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

The domain and initial condition are shown in Fig. 4. The characteristics of the system point from the bottom left to the top right. Thus, inflow boundary conditions are imposed on the bottom and left walls as well as the lower-left embedded object. Outflow conditions occur at the top and right walls as well as along the top-right embedded object, providing a good test for the new interpolation/derivative stencils.

Preliminary results for this challenging hyperbolic system are demonstrated in Fig. 5, which shows the L_∞ norm of the error as a function of time over 500 periods of the solution for a grid resolution of 50 × 50. While further analysis is required, the success of the optimizer in finding a stable 2^{nd} order derivative/interpolation scheme for this challenging hyperbolic problem is promising.

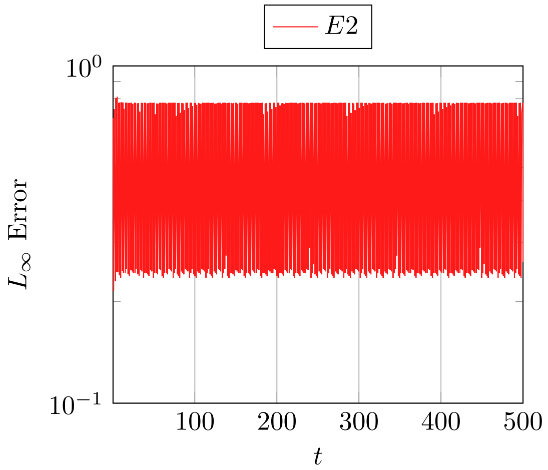


Figure 5: L_∞ norm of the error as a function of time for the variable coefficient advection test. 2^{nd} order central differencing is used in the interior. The error varies smoothly with the period of the solution and does not grow over 500 solution periods with the optimized interpolation/derivative stencils.

3 Conclusion and Future Work

Our novel optimization approach to discover stable cut-cell discretizations has allowed for the derivation of stable, conservative and high-order derivative approximations based on finite differences. These approaches address the "small-cell problem" by construction and do not require any ad-hoc stabilization. Generalizing our previous work to the Navier-Stokes equations highlights the need for optimizing interpolation stencils in addition to the derivative stencils. In this work, we have explored a unified framework to accomplish this and preliminary results, though only second order, showcase the stability of the resulting method and provide a path forward for pursuing higher-order stencils.

Acknowledgments

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 numbers ER20190227 and DR20220104. Computational resources were provided by the LANL Institutional Computing (IC) Program.

References

[1] The legion programming system, 2022. https://legion.stanford.edu/.

[2] P. Brady and D. Livescu. High-order, stable, and conservative boundary schemes for central and compact finite differences. *Computers & Fluids*, 183:84–101, 2019.

[3] P. Brady and D. Livescu. Foundations for high-order, conservative cut-cell methods: Stable discretizations on degenerate meshes. *Journal of Computational Physics*, 426:109794, 2021.

[4] P. T. Brady. Finite difference optimizer, 2022. https://github.com/lanl/fido.

[5] P. T. Brady. Stable, high-order cut-cell solver, 2022. https://github.com/lanl/shoccs.

[6] P. T. Brady and D. Livescu. Stable, high-order and conservative cut-cell methods. *AIAA Scitech 2019 Forum*, AIAA 2019-1991, 2019.

[7] C. Brehm, C. Hader, and H. Fasel. A locally stabilized immersed boundary method for the compressible Navier-Stokes equations. *Journal of Computational Physics*, 295:475–504, aug 2015.

[8] M. H. Carpenter, D. Gottlieb, and S. Abarbanel. The Stability of Numerical Boundary Treatments for Compact High-Order Finite-Difference Schemes. *J. Comput. Phys.*, 108(2):272–295, 1993.

[9] D. Clarke, H. Hassan, and M. Salas. Euler calculations for multielement airfoils using Cartesian grids. *AIAA Journal*, 24(3):353–358, 1986.

[10] S. G. Johnson. The nlopt nonlinear-optimization package, 2021. https://github.com/stevengj/nlopt.

[11] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37(1):239–261, jan 2005.

[12] C. S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972.

[13] N. Sharan, P. T. Brady, and D. Livescu. Stable and conservative boundary treatment for difference methods, with application to cut-cell discretizations. *AIAA Scitech 2020 Forum*, AIAA 2020-0807, 2020.

[14] N. Sharan, P. T. Brady, and D. Livescu. Finite-difference cartesian cut-cell method for hyperbolic systems. *AIAA Scitech 2021 Forum*, AIAA 2021-0746, 2021.

[15] N. Sharan, P. T. Brady, and D. Livescu. High-order dimensionally-split cartesian embedded boundary method for non-dissipative schemes. *J. Comput. Phys.*, 464:111341, 2022.

[16] N. Sharan, P. T. Brady, and D. Livescu. Time stability of strong boundary conditions in finite-difference schemes for hyperbolic systems. *SIAM J. Numerical Analysis*, 60:1331–1362, 2022.