

A hybrid semi-Lagrangian cut cell method for advection-diffusion problems with Robin boundary conditions in moving domains

Aaron Barrett^{a,*}, Aaron L. Fogelson^b, Boyce E. Griffith^{c,d,e,f}

^a Department of Mathematics, University of Utah, Salt Lake City, UT, USA

^b Departments of Mathematics and Bioengineering, University of Utah, Salt Lake City, UT, USA

^c Departments of Mathematics, Applied Physical Sciences, and Biomedical Engineering, University of North Carolina, Chapel Hill, NC, USA

^d Carolina Center for Interdisciplinary Applied Mathematics, University of North Carolina, Chapel Hill, NC, USA

^e Computational Medicine Program, University of North Carolina, Chapel Hill, NC, USA

^f McAllister Heart Institute, University of North Carolina, Chapel Hill, NC, USA

ARTICLE INFO

Article history:

Available online 28 October 2021

Keywords:

Irregular domain

Level set method

Robin boundary condition

Cartesian grid method

ABSTRACT

We present a new discretization approach to advection-diffusion problems with Robin boundary conditions on complex, time-dependent domains. The method is based on second order cut cell finite volume methods introduced by Bochkov et al. [8] to discretize the Laplace operator and Robin boundary condition. To overcome the small cell problem, we use a splitting scheme along with a semi-Lagrangian method to treat advection. We demonstrate second order accuracy in the L^1 , L^2 , and L^∞ norms for both analytic test problems and numerical convergence studies. We also demonstrate the ability of the scheme to convert one chemical species to another across a moving boundary.

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

The convective transport and diffusion of chemical species occurs in a broad range of systems. Many applications involve chemical concentrations that evolve within complex, time-dependent regions, including blood flow and clotting in the cardiovascular system [14,25], particulate and chemical vapor transport in the lungs [15,39,43], and drug absorption in the digestive tract [7,33,48]. In some cases, critical interactions occur between fluid-phase and structure-bound chemicals. These interactions can appear in the model equations as a Robin boundary condition for the fluid-bound chemical. For example, the interaction between circulating proteins and membrane-bound proteins plays a pivotal role in thrombus formation [14]. Modeling these interactions becomes even more challenging as one considers the motion of the flow domain itself.

The numerical simulation of PDEs in complex domains has garnered significant attention for decades. Embedded boundary methods are popular approaches to such problems in which a fixed rectangular Cartesian grid is overlaid on the complex structure. Embedded boundary approaches typically alter the PDE to include an additional source term that is non-zero only near the boundary. For instance, the immersed boundary method [17,35] uses an integral transform with a regularized delta function kernel to enforce boundary conditions along irregular interfaces immersed in a background Cartesian grid. These methods typically are designed for Dirichlet boundary conditions, and they have to be modified with special interpolation procedures to allow for other types of boundary conditions [9,32,49]. Volume penalization [22] meth-

* Corresponding author.

E-mail address: barrett@math.utah.edu (A. Barrett).

ods and diffuse domain [27,44,47] methods both introduce phase field models to track the interface. A smoothed version of the interface is then used to modify the original PDE to account for the boundary conditions, ultimately yielding an approach similar to the immersed boundary method. These methods all have the effect of smoothing the boundary of the complex domain over several grid cells. Regularization lowers solution accuracy near the boundary, and this can limit their effectiveness in applications in which boundary interactions are important. One alternative approach is the immersed interface method [24,28,45], which derives jump conditions across the interface and then builds these jump conditions into the discretized equations. Using this method to impose boundary conditions requires relating the boundary conditions to jump conditions along the interface. Many applications have successfully used the immersed interface method, but to our knowledge, the jump conditions for arbitrary spatially dependent Robin boundary conditions have not yet been determined. A closely related approach is the ghost fluid method [10], which uses an interpolation procedure to build stencils near the interface that incorporate the boundary conditions. This approach can lead to non-symmetric and, in some cases, ill-conditioned systems that require specialized solvers [46]. Another alternative to these methods is the immersed boundary smooth extension (IBSE) method [36,40], which can solve PDEs on complex geometries by embedding the geometry in a simpler region and solving the PDE on the extended domain that now includes both “physical” and “non-physical” sub-domains. In the IBSE approach, a body force is incorporated in the “non-physical” regime to extend the physical solution smoothly outside the physical domain. This allows for high order accuracy to the boundary, but at the cost of solving an additional multiharmonic problem, with the order depending on the number of interface conditions.

The approach that we use here to impose boundary conditions along a complex boundary while retaining a Cartesian grid discretization framework is based on a cut-cell finite volume formulation. In these flux-based methods, fluxes are carefully calculated to account for the portion of cells that are inside the physical domain. This approach allows for accurate solutions along the boundary, but comes at the expense of computing cell geometries at the interface. This expense can be alleviated through the use of a level set function to track the surface [30]. Recent work by Helgadottir et al. [18] demonstrated the ability to impose Dirichlet, Neumann, and Robin boundary conditions for Poisson problems. Cut-cell methods can suffer from the so-called “small cell problem,” however, in which cells with small volumes necessitate the use of extremely small time step sizes for conditionally stable time stepping schemes. This becomes a serious problem for hyperbolic equations for which accurate, efficient, and unconditionally stable implicit time-stepping schemes are difficult to create. Approaches to alleviate the small cell problem for advective PDEs involve using an implicit method only for the cut-cells [29], merging small cells with their neighbors to effectively create a larger cell [38], or partitioning fluxes into “shielded” and “unshielded” zones based on the geometry of the cut-cell [23]. These methods have been successfully deployed in two spatial dimensions, but extending them to three spatial dimensions remains a challenge. Recently, cut-cell methods that do not suffer from the small cell problem have been introduced to handle moving boundaries [38,42]; however, a formulation involving Robin boundary conditions that achieves second order accuracy has yet to be developed. The major contribution of this study is the construction of such a method with second order accuracy.

Herein, we develop a cut-cell method for advection-diffusion equations on moving domains that allows for the imposition of general Robin boundary conditions. To avoid the small cell problem, we introduce a split cut-cell semi-Lagrangian scheme, in which the diffusive operator is handled using well established finite volume methods, and the advective operator is treated using a semi-Lagrangian method. The benefit of splitting the two operators is two fold. First, in the diffusive solve, the flux from the boundary does not need to account for the change in location of the boundary. This allows us to leverage recent work by Papac et al. [34] and Bochkov et al. [8] to solve a Poisson-like problem with stationary boundaries. Second, the semi-Lagrangian scheme for advection has no CFL stability constraint, so the small cell problem is no longer an issue. We demonstrate that this method can accurately resolve concentrations near the boundary for both Robin and Neumann boundary conditions, including inhomogeneous Robin boundary conditions. In addition, we show that this method is able to handle conversion of one concentration into another across the boundary, effectively modeling surface reactions.

2. Continuous equations

We consider the advection and diffusion of a chemical species with concentration $q(\mathbf{x}, t)$ in an arbitrary domain Ω_t , embedded in a larger rectangular region $\mathcal{B}$. Both $q(\mathbf{x}, t)$ and Ω_t are transported by the same velocity field $\mathbf{u}(\mathbf{x}, t)$. We assume that $q(\mathbf{x}, t)$ diffuses with diffusion coefficient D . We consider the particular case in which $q(\mathbf{x}, t)$ satisfies Robin boundary conditions on the boundary Γ_t of Ω_t , so that

$$\frac{\partial q(\mathbf{x}, t)}{\partial t} + \nabla \cdot (\mathbf{u}(\mathbf{x}, t) q(\mathbf{x}, t) - D \nabla q(\mathbf{x}, t)) = f(\mathbf{x}, t), \mathbf{x} \in \Omega_t, \quad (1a)$$

$$D \nabla q(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}, t) + a(\mathbf{x}, t) q(\mathbf{x}, t) = g(\mathbf{x}, t), \mathbf{x} \in \Gamma_t, \quad (1b)$$

in which $\mathbf{n}(\mathbf{x}, t)$ is the outward unit normal of Γ_t and $f(\mathbf{x}, t)$ is a given volumetric source function. Although $q(\mathbf{x}, t)$ could be extended to be defined on $\mathcal{B}$, in our implementation, $q(\mathbf{x}, t)$ is defined only inside Ω_t , and the bounding region $\mathcal{B}$ is used only to define the cells contained within Ω_t .

We describe the boundary Γ_t using a signed distance function $\phi(\mathbf{x}, t)$ such that

$$\Gamma_t = \{\mathbf{x} \in \mathcal{B} \mid \phi(\mathbf{x}, t) = 0\}. \quad (2)$$

The signed distance function is passively advected by the prescribed velocity,

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} + \nabla \cdot (\mathbf{u}(\mathbf{x}, t) \phi(\mathbf{x}, t)) = 0, \mathbf{x} \in \mathcal{B}. \quad (3)$$

Because $\phi(\mathbf{x}, t)$ will not in general remain a signed distance function under advection by (3), a reinitialization procedure is used to maintain the signed distance property. This can be achieved by computing the steady-state solution to the Hamilton-Jacobi equation

$$\begin{aligned} \frac{\partial \hat{\phi}(\mathbf{x}, \tau)}{\partial \tau} + \text{sgn}(\hat{\phi}(\mathbf{x}, \tau)) \left(\|\nabla \hat{\phi}(\mathbf{x}, \tau)\| - 1 \right) &= 0, \\ \hat{\phi}(\mathbf{x}, 0) &= \phi(\mathbf{x}, t), \end{aligned} \quad (4)$$

after which $\phi(\mathbf{x}, t)$ is set to the steady state solution.

3. Numerical methods

To simplify notation, we describe the method in two spatial dimensions. Extensions to a third spatial dimension are straightforward, with the exception of calculating cut-cell geometries [30].

We use a splitting scheme to split equation (1a) into a diffusion step,

$$\frac{\partial q(\mathbf{x}, t)}{\partial t} - \nabla \cdot (D \nabla q(\mathbf{x}, t)) = f(\mathbf{x}, t), \mathbf{x} \in \Omega_t \quad (5a)$$

$$D \nabla q(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}, t) + a(\mathbf{x}, t) q(\mathbf{x}, t) = g(\mathbf{x}, t), \mathbf{x} \in \Gamma_t, \quad (5b)$$

in which the domain Ω_t remains fixed, followed by an advection step,

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} + \nabla \cdot (\mathbf{u} \phi(\mathbf{x}, t)) = 0 \text{ for } \mathbf{x} \in \mathcal{B} \quad (6a)$$

$$\frac{\partial q(\mathbf{x}, t)}{\partial t} + \nabla \cdot (\mathbf{u} q(\mathbf{x}, t)) = 0 \text{ for } \mathbf{x} \in \Omega_t, \quad (6b)$$

in which we evolve both the level set and the function $q(\mathbf{x}, t)$. We note that because Ω_t and $q(\mathbf{x}, t)$ evolve with the same velocity, no boundary condition is needed with equation (6). The use of this splitting procedure incurs an additional error. This approach can still yield second order temporal accuracy through Strang splitting [41], which will be described later.

We overlay a Cartesian grid on top of $\mathcal{B}$ such that $\mathcal{B}$ consists of rectangular grid cells $\mathbf{c}_{i,j}$ and $\mathcal{B} = \cup \mathbf{c}_{i,j}$ with a grid cell spacing of $\Delta x = \Delta y = h$. The concentration field is approximated at the cell centroid of each full or partial cell contained within Ω_t .

In the following sections, we describe the discretization of equations (5) and (6). In what follows, unless otherwise noted, $q_{i,j}$ refers to the concentration at the cell centroid of the cell $\mathbf{c}_{i,j} \cap \Omega_t$.

3.1. Diffusion step

To solve the diffusion step from equations (5), we employ a cut-cell finite volume method based on the approaches of Papac et al. [34] and Arias et al. [1]. We summarize the derivation here, and refer the interested reader to a more detailed description in previous work. Integrating equation (5a) over a cell $\mathbf{c}_{i,j}$ that is entirely or partially interior to Ω_t and dividing by the volume of the cell, we get

$$\frac{1}{|\mathbf{c}_{i,j} \cap \Omega_t|} \int_{\mathbf{c}_{i,j} \cap \Omega_t} \frac{\partial q(\mathbf{x}, t)}{\partial t} d\mathbf{x} = \frac{1}{|\mathbf{c}_{i,j} \cap \Omega_t|} \int_{\mathbf{c}_{i,j} \cap \Omega_t} D \Delta q(\mathbf{x}, t) d\mathbf{x}. \quad (7)$$

We define $Q_{i,j}$ as the cell average of $q(\mathbf{x}, t)$ in the cell $\mathbf{c}_{i,j} \cap \Omega_t$. Replacing the cell average in the left side of equation (7) and employing the divergence theorem on the right-hand side, we get

$$\frac{dQ_{i,j}}{dt} = \frac{1}{|\mathbf{c}_{i,j} \cap \Omega_t|} \int_{\partial(\mathbf{c}_{i,j} \cap \Omega_t)} D \nabla q(\mathbf{x}, t) \cdot \mathbf{n} \cdot d\mathbf{A}, \quad (8)$$

in which $\mathbf{n}$ is the outward unit normal of $\partial(\mathbf{c}_{i,j} \cap \Omega_t)$. We can further divide the integral in equation (8) into an integral over the cell boundary $\partial \mathbf{c}_{i,j} \cap \Omega_t$ and an integral over the physical boundary $\Gamma_t \cap \mathbf{c}_{i,j}$

$$\int_{\partial(\mathbf{c}_{i,j} \cap \Omega_t)} D \nabla q(\mathbf{x}, t) \cdot \mathbf{n} dA = \left(\int_{\partial \mathbf{c}_{i,j} \cap \Omega_t} + \int_{\mathbf{c}_{i,j} \cap \Gamma_t} \right) D \nabla q(\mathbf{x}, t) \cdot \mathbf{n} \cdot d\mathbf{A}. \quad (9)$$

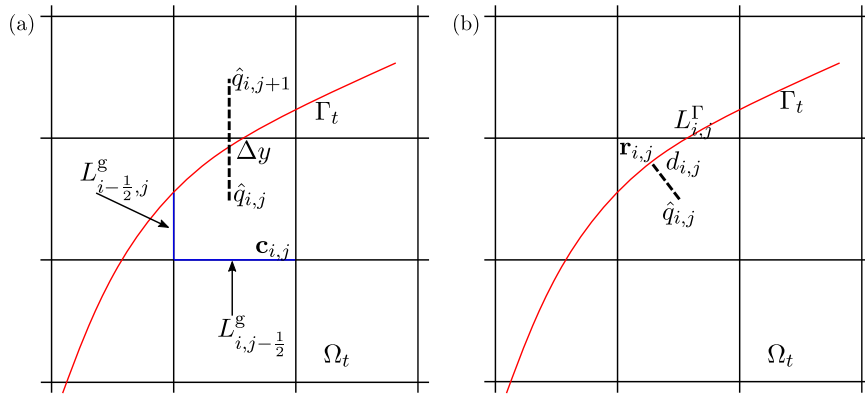


Fig. 1. Depiction of the nomenclature used in the diffusion discretization for the cell length fractions (a) and for the boundary fluxes (b).

We approximate the first integral by

$$\int_{\partial \mathbf{c}_{i,j} \cap \Omega_t} D \nabla q(\mathbf{x}, t) \cdot \mathbf{n} dA \approx L_{i+1/2,j}^g \frac{\hat{q}_{i+1,j} - \hat{q}_{i,j}}{\Delta x} - L_{i-1/2,j}^g \frac{\hat{q}_{i,j} - \hat{q}_{i-1,j}}{\Delta x} + L_{i,j+1/2}^g \frac{\hat{q}_{i,j+1} - \hat{q}_{i,j}}{\Delta y} - L_{i,j-1/2}^g \frac{\hat{q}_{i,j} - \hat{q}_{i,j-1}}{\Delta y}, \quad (10)$$

in which $\hat{q}_{i,j}$ is the point-wise concentration at the center of the cell $\mathbf{c}_{i,j}$ and $L_{i+1/2,j}^g$ is the length fraction of the face $(i + \frac{1}{2}) \times [j - \frac{1}{2}, j + \frac{1}{2}]$ covered by the irregular domain, see Fig. 1a. It is challenging to compute $L_{i+1/2,j}^g$ exactly. The smoothness of the level set is determined by the zero contour. For shapes that have sharp features, the level set will inherit these features. However, to achieve second order accuracy for smooth level sets, it suffices to use a linear approximation

$$L_{i+1/2,j}^g = \begin{cases} \Delta y \left| \frac{\phi_{i+1/2,j-1/2}}{\phi_{i+1/2,j-1/2} - \phi_{i+1/2,j+1/2}} \right|, & \text{if } \phi_{i+1/2,j-1/2} \phi_{i+1/2,j+1/2} < 0, \\ \Delta y, & \text{if } \phi_{i+1/2,j-1/2} < 0 \text{ and } \phi_{i+1/2,j+1/2} < 0, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

While in the evolution of the level set $\phi(\mathbf{x}, t)$, the degrees of freedom live at cell centers, in equation (11), we require the value of $\phi(\mathbf{x}, t)$ at cell nodes. In our computations, we use a simple bi-linear interpolant to find these nodal values. In three spatial dimensions, evaluating the corresponding quantity $L_{i+1/2,j,k}^g$ would involve computing the surface area.

We emphasize that $\hat{q}_{i,j}$ refers to the cell center of the (possibly cut) grid cell $\mathbf{c}_{i,j}$ regardless of the location of the physical boundary Γ_t . In cases where the cell center does not correspond to the location of degrees of freedom, e.g., near cut-cells, we must reconstruct these values. Here, we perform this reconstruction using either a moving least squares (MLS) approximation, or a radial basis function (RBF) interpolant. Both procedures are described in more detail in Section 3.3. We note that this reconstruction is not required in the methods of Arias et al. [1] and Bochkov et al. [8], as those methods always define the degrees of freedom at cell centers.

As done in Bochkov et al. [8], the second integral is approximated using a linear approximation to $q(\mathbf{x}, t)$ on the boundary in the direction normal to Γ_t

$$\int_{\mathbf{c}_{i,j} \cap \Gamma_t} D \nabla q(\mathbf{x}, t) \cdot \mathbf{n} dA = \int_{\mathbf{c}_{i,j} \cap \Gamma_t} (g(\mathbf{x}, t) - a q(\mathbf{x}, t)) dA \approx L_{i,j}^\Gamma (g(\mathbf{r}_{i,j}, t) - a q(\mathbf{r}_{i,j}, t)), \quad (12)$$

in which $\mathbf{r}_{i,j}$ is the closest point on Γ_t to $\mathbf{x}_{i,j}$ and $L_{i,j}^\Gamma = |\mathbf{c}_{i,j} \cap \Gamma_t|$ (see Fig. 1b). The value $q(\mathbf{r}_{i,j}, t)$ is found using a Taylor series expansion

$$q(\mathbf{x}_{i,j}, t) = q(\mathbf{r}_{i,j}, t) + d_{i,j} \frac{\partial q(\mathbf{r}_{i,j}, t)}{\partial \mathbf{n}} + \mathcal{O}(h^2), \quad (13)$$

in which $d_{i,j} = \frac{\phi(\mathbf{x}_{i,j}, t)}{|\nabla \phi(\mathbf{x}_{i,j}, t)|}$ is the distance between $\mathbf{r}_{i,j}$ and cell center $\mathbf{x}_{i,j}$. On the boundary, we have that

$$D \frac{\partial q(\mathbf{r}_{i,j}, t)}{\partial \mathbf{n}} + a q(\mathbf{r}_{i,j}, t) = g(\mathbf{r}_{i,j}, t). \quad (14)$$

We solve equations (13) and (14), dropping the $\mathcal{O}(h^2)$ terms, for $q(\mathbf{r}_{i,j}, t)$ and use this value in the approximation to the integral in equation (12). As shown previously [8], this system is well posed for a sufficiently refined grid.

The cut-cell volume $|\mathbf{c}_{i,j} \cap \Omega_t|$ and physical boundary length $|\mathbf{c}_{i,j} \cap \Gamma_t|$ are found by decomposing the cut-cell and boundary into simplices, for which analytic formulas for the volume exist [30].

We approximate equation (5) using the implicit trapezoidal rule:

$$\begin{aligned} & \frac{1}{\Delta t} Q_{i,j}^{n+1} - \frac{D}{2|\mathbf{c}_{i,j} \cap \Omega^-|} \left(L_{i+\frac{1}{2},j}^g \frac{q_{i+1,j}^{n+1} - q_{i,j}^{n+1}}{\Delta x} - L_{i-\frac{1}{2},j}^g \frac{q_{i,j}^{n+1} - q_{i-1,j}^{n+1}}{\Delta x} \right. \\ & \quad \left. + L_{i,j+\frac{1}{2}}^g \frac{q_{i,j+1}^{n+1} - q_{i,j}^{n+1}}{\Delta y} - L_{i,j-\frac{1}{2}}^g \frac{q_{i,j}^{n+1} - q_{i,j-1}^{n+1}}{\Delta y} \right) \\ &= \frac{1}{\Delta t} Q_{i,j}^n + \frac{D}{2|\mathbf{c}_{i,j} \cap \Omega^-|} \left(L_{i+\frac{1}{2},j}^g \frac{q_{i+1,j}^n - q_{i,j}^n}{\Delta x} - L_{i-\frac{1}{2},j}^g \frac{q_{i,j}^n - q_{i-1,j}^n}{\Delta x} \right. \\ & \quad \left. + L_{i,j+\frac{1}{2}}^g \frac{q_{i,j+1}^n - q_{i,j}^n}{\Delta y} - L_{i,j-\frac{1}{2}}^g \frac{q_{i,j}^n - q_{i,j-1}^n}{\Delta y} \right) + \frac{1}{|\mathbf{c}_{i,j} \cap \Omega^-|} f_{i,j}^{n+\frac{1}{2}}. \end{aligned} \quad (15)$$

In our computational examples, we use GMRES without preconditioning to solve this system of equations. The solver typically converges to a solution with tolerance 10^{-12} after approximately 20 to 50 iterations.

3.2. Advection step

For the advection step, we use a semi-Lagrangian method [11,37] to advance both the level set $\phi(\mathbf{x}, t)$ and the concentration field $q(\mathbf{x}, t)$. We solve equation (6) by first finding the preimage $\mathbf{X}_{i,j} = \chi^{-1}(\mathbf{x}_{i,j}, t^{n+1})$ of the fluid parcel located at $\mathbf{x}_{i,j}$ at time t^{n+1} . The mapping $\chi(\mathbf{x}, t)$ satisfies the differential equation

$$\frac{\partial \chi(\mathbf{x}, t)}{\partial t} = \mathbf{u}(\mathbf{x}, t) \text{ for } \mathbf{x} \in \mathcal{B}. \quad (16)$$

The preimage $\mathbf{X}_{i,j}$ can be found by integrating equation (16) backwards in time. We use an explicit two step Runge-Kutta method:

$$\mathbf{X}_{i,j}^* = \mathbf{x}_{i,j} - \frac{\Delta t}{2} \mathbf{u}(\mathbf{x}_{i,j}, t^{n+1}), \quad (17a)$$

$$\mathbf{X}_{i,j}^n = \mathbf{x}_{i,j} - \Delta t \mathbf{u}(\mathbf{X}_{i,j}^*, t^{n+\frac{1}{2}}). \quad (17b)$$

Having found the preimage, we can interpolate the solution at time t^n at the preimage locations $\mathbf{X}_{i,j}^n$.

Our interpolation procedure consists of one of two possible methods depending on whether $\mathbf{X}_{i,j}^n$ is near cut-cells. Away from cells pierced by the zero level set, we use a tensor product of special piecewise Hermite polynomials called Z-splines [6]. The Z-spline $Z_m(x)$ interpolating the data $(x_i, f_i = f(x_i))_{i=1}^n$ is a piecewise polynomial function that satisfies

$$Z_m(x) \in C^m([x_1, x_n]), \quad (18a)$$

$$\left. \frac{d^p}{dx^p} Z_m(x) \right|_{x_j} = f_{m,j}^p \text{ for } p = 0, \dots, m \text{ and } j = 1, \dots, n, \quad (18b)$$

$$Z_m(x) \in \pi_{2m+1}([x_i, x_{i+1}]) \text{ for } i = 1, \dots, n-1, \quad (18c)$$

in which $f_{m,j}^p$ is the approximation of the p^{th} order derivative of $f(x)$ computed from high-order finite differences of f_j using $2m+1$ points, and π_{2m+1} is the space of polynomials of degree less than or equal to $2m+1$.

It is possible to define Z_m for a general set of data points (or abscissae) via the cardinal Z-splines [6]

$$Z_m(x) = \sum_i f_i \tilde{Z}_m(x - x_i). \quad (19)$$

The cardinal Z-splines have two key properties. First, the cardinal Z-splines have compact support, $\tilde{Z}_m(x) = 0$ for $|x| > m+1$, which allows for efficient local evaluation of interpolants. Second, cardinal Z-splines are interpolatory, which makes the interpolant trivial to form. Because data values are defined on a regular Cartesian grid, the computation of the full interpolant uses a tensor product of cardinal Z-splines

$$Z_m(\mathbf{x}) = \sum_i \sum_j q_{i,j} \tilde{Z}_m(x - x_i) \tilde{Z}_m(y - y_j). \quad (20)$$

In our computations, we use the quintic Z-splines, which are defined in terms of

$$Z_2(x) = \begin{cases} 1 - \frac{15}{12}x^2 - \frac{35}{12}x^3 + \frac{63}{12}x^4 - \frac{25}{12}x^5, & \text{if } |x| \leq 1, \\ -4 + \frac{75}{4}x - \frac{245}{8}x^2 + \frac{545}{24}x^3 - \frac{63}{8}x^4 + \frac{25}{24}x^5, & \text{if } 1 < |x| \leq 2, \\ 18 - \frac{153}{4}x + \frac{255}{8}x^2 - \frac{313}{24}x^3 + \frac{21}{8}x^4 - \frac{5}{24}x^5, & \text{if } 2 < |x| \leq 3, \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

In cases where we do not have enough points to define the Z-spline, e.g., near cut-cells, we use the procedure described in Section 3.3. We note that although the interpolation procedure described here has a discontinuous switch between operators, this does not appear to affect the overall convergence rates in our numerical tests of the methodology.

We note that this form of the semi-Lagrangian method is not conservative, but, in our experiments, the change in the amount of material was very small. Conservative versions of semi-Lagrangian methods have been developed [21,26]. The core change is to advect *grid cells* instead of individual points, and then integrate the resulting interpolating polynomial over the grid cell. Because conservation is not critical to our ultimate problem of interest, we use the simpler non-conservative approach in this work.

The evolution of the level set and the concentration use the same procedure, except for the choice of reference grid. The reference grid for the level set consists of the entire domain $\mathcal{B}$, whereas the reference grid for the concentration consists of the domain $\Omega_{t^{n+1}} = \{\mathbf{x} \in \mathcal{B} | \phi(\mathbf{x}, t^{n+1}) < 0\}$. Therefore, we update the level set prior to updating the concentration.

3.3. Moving least squares and radial basis function interpolation

Near cut-cells, it is necessary to form interpolants on unstructured data. In this study, we compare the accuracy of a moving least squares (MLS) approximation and a local radial basis function (RBF) interpolant. This procedure is used both in the diffusion step, to extrapolate data from cut-cell centroids to full-cell centers, and also in the advection step, for cells where the Z-spline interpolant can not be formed. We describe the procedure in the context of reconstructing a function $f(\mathbf{x})$ from the data points $\{\mathbf{x}_i, f(\mathbf{x}_i)\}$ with $i = 1, 2, \dots, n$ in which n is an arbitrary number of points.

The MLS approximation is formed by finding the best approximation $q(\mathbf{x})$ at a point $\mathbf{x}_c$ of the form

$$q(\mathbf{x}) = \sum_{j=1}^N c_j p_j(\mathbf{x}) \quad (22)$$

in which the $p_j(\mathbf{x})$ form a basis for the space of polynomials up to a certain degree and N is the number of polynomials in the basis. The approximation $q(\mathbf{x})$ is chosen to be optimal with respect to the standard weighted L^2 inner product with weight function $w(\mathbf{x})$

$$\langle f(\mathbf{x}), g(\mathbf{x}) \rangle = \int_{\Omega} f(\mathbf{x}) g(\mathbf{x}) w(\mathbf{x}) d\mathbf{x}. \quad (23)$$

We use a stencil width of approximately two grid cells. It is known that the weight function must be singular at the data locations $\mathbf{x}_i$ for the MLS approximation to interpolate the data [5]. Here, we use

$$w(\mathbf{x}) = e^{-\|\mathbf{x} - \mathbf{x}_c\|^2}. \quad (24)$$

Because this weighting function is not singular at $\mathbf{x}_c$, the reconstructed polynomial will not be interpolatory. The MLS calculation reduces to solving a linear system of the form

$$\Lambda \mathbf{A} \mathbf{c} = \Lambda \mathbf{f} \quad (25)$$

in which $\mathbf{A}$ is a matrix whose entries consist of $a_{i,j} = p_j(\mathbf{x}_i)$ and Λ is a diagonal matrix with $\lambda_{i,i} = w(\mathbf{x}_i)$. In preliminary computational tests, we observed that using a lower order reconstruction for the diffusion step is more robust without affecting the order of accuracy while quadratic reconstructions can be used in the advection step. Consequently, we use a quadratic polynomial in the reconstructions for the advection step and a linear reconstruction with the diffusion step for the tests reported in section 4.

The RBF approximation is constructed via a polyharmonic spline of the form

$$q(\mathbf{x}) = \sum_{j=1}^n \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) + \sum_{j=1}^s \beta_j p_j(\mathbf{x}) \quad (26)$$

in which $\phi(\mathbf{x})$ is a polyharmonic radial basis function of degree m and $p_j(\mathbf{x})$ are a set of s polynomial basis functions [5,13]. The degree of the polynomials k is chosen such that $m = 2k + 1$. The coefficients are chosen so that

$$q(\mathbf{x}_j) = f_j \text{ for } j = 1, \dots, n, \quad (27)$$

$$\sum_{k=1}^n \lambda_j p_k(\mathbf{x}_j) = 0 \text{ for } j = 1, \dots, n. \quad (28)$$

This leads to the linear system for the coefficients

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \beta \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix} \quad (29)$$

in which A is a square matrix with elements $A_{i,j} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$ and P is a rectangular matrix with elements $P_{i,j} = p_j(\mathbf{x}_i)$. In our computations, we use the polyharmonic spline $\phi(\mathbf{x}) = \|\mathbf{x}\|^3$ combined with linear polynomials.

3.4. Full procedure and implementation

We summarize the full procedure we use to advance the solution from time t^n to time t^{n+1} :

- 1) If needed, reinitialize the level set $\phi(\mathbf{x}, t)$ into the signed distance function by iterating equation (4) to steady state using the procedure as described by Nangia et al. [31].
- 2) Update the diffusion equation to half time $t^{n+\frac{1}{2}}$ using the methods described in Section 3.1.
- 3) Update the level set using the prescribed velocity $\mathbf{u}$ to time t^{n+1} by the methods described in Section 3.2.
- 4) Advect the concentration using the prescribed velocity $\mathbf{u}$ by the methods described in Section 3.2.
- 5) Update the diffusion equation to full time t^{n+1} using the concentration and level sets from the previous two steps.

The above procedure is implemented using the SAMRAI [19] infrastructure, which provides an efficient, parallelized environment for structured adaptive mesh refinement. The diffusion solve is computed using matrix free solvers with operators provided by IBAMR [16,20] and Krylov methods provided by PETSc [2–4].

4. Results

Here we demonstrate the capabilities of the method both using a prescribed level set and an level set advected with the fluid. We start with diffusion dominated examples before exploring inclusion of advection and spatially varying boundary conditions.

4.1. Diffusion from a point source

We consider the advection-diffusion of a point source within a disk of radius R . The disk is advected with velocity $\mathbf{u} = (\cos(\frac{\pi}{4}), \sin(\frac{\pi}{4}))$. The exact concentration is

$$q(\mathbf{x}, t) = \frac{10}{4D(t + \frac{1}{2})} e^{-\frac{\|\mathbf{x} - \mathbf{x}_c(t)\|^2}{4D(t + \frac{1}{2})}} \text{ for } \mathbf{x} \in \mathcal{B}, \quad (30)$$

in which $\mathbf{x}_c(t)$ is the center of the disk. We apply Robin boundary conditions of the form

$$D \frac{\partial q}{\partial \mathbf{n}} = g(\mathbf{x}, t) - a(\mathbf{x}, t) q(\mathbf{x}, t). \quad (31)$$

We set $a(\mathbf{x}, t) = 1$ and use the method of manufactured solutions to determine the value of $g(\mathbf{x}, t)$. For this example, we specify the location of the level set ϕ at each timestep. This procedure allows us test the convergence of the advection-diffusion step without having to account for numerical error in the evolution of the level set function. We note that whereas this example includes a trivial semi-Lagrangian backwards integration step, the interpolation procedure and diffusion step are non-trivial.

The extended domain $\mathcal{B}$ is the box $[0, 12] \times [0, 12]$ and is discretized using N grid cells in each direction. We use a disk radius of $R = 1$ and initial center $\mathbf{x}_c(0) = (1.521, 1.503)$. The choice of center slightly offsets the disk from the background grid so that different cut-cells are generated on opposite sides of the disk. We use a diffusion coefficient of $D = 0.01$. The simulations are run at an advective CFL number of $C_{\text{CFL}} = \frac{\Delta x}{|\mathbf{u}| \Delta t} = 0.5$.

Fig. 2 shows convergence plots, which demonstrate second order accuracy in the L^1 , L^2 , and L^∞ norms. The coarsest simulation uses $N = 128$ grid cells in each direction of the background Cartesian mesh, which corresponds to approximately 26 grid cells covering the diameter of the disk. Despite the relatively coarse description of the disk, we still see errors that are on the order of one percent.

We also consider the case where the level set ϕ is advected with the same velocity. In this case, numerical issues make it difficult to assess the pointwise error on the Cartesian grid because the piecewise linear reconstruction used to determine

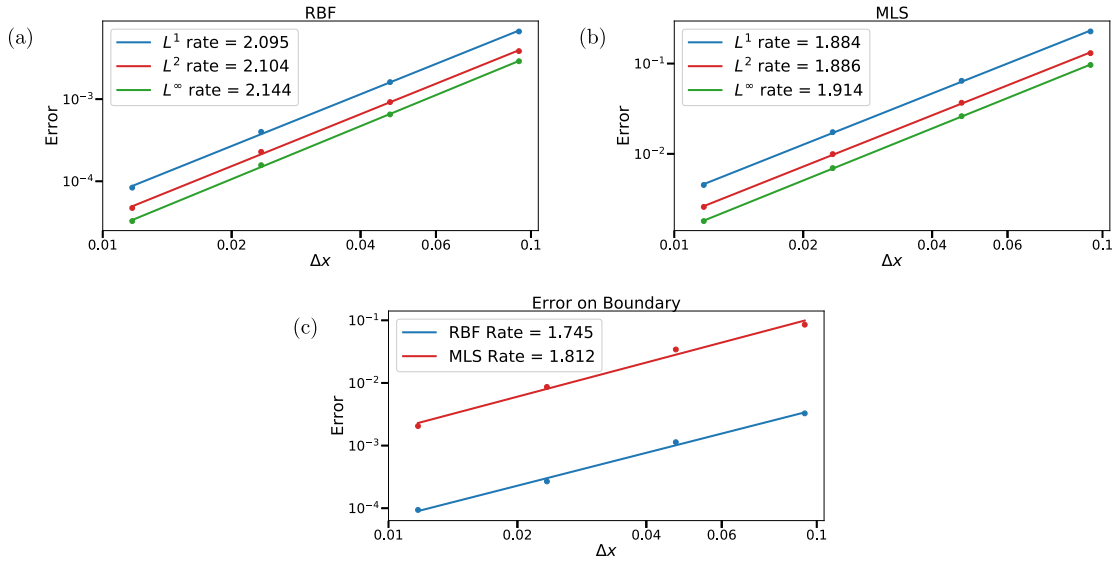


Fig. 2. Convergence rates for an advecting and diffusion concentration field initialized from a point source in a constant flow using either a RBF (a) or MLS (b) reconstruction. Also shown is the convergence rate for the integral in equation (32) using either an RBF or MLS reconstruction (c). In all cases, the error using RBF reconstructions is more than an order of magnitude smaller than MLS reconstructions. Simulations are run to a final time of $T = 10$ and at a CFL number of $C_{\text{CFL}} = 0.5$. We use a diffusion coefficient of $D = 0.01$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

the cut-cells gives different cut-cell geometries than that of a prescribed level set. Instead, we compare the solution at the boundary to the imposed boundary condition. We compute the solution on the boundary using a moving least squares linear extrapolation from cut-cells to the cell boundary. Further, to assess the convergence rate, we compute the error of the surface integral

$$E = \left| \int_0^{2\pi} q(\mathbf{x}, t) d\theta - \int_0^{2\pi} \tilde{q}(\mathbf{x}, t) d\theta \right|, \quad (32)$$

in which θ is the angle along the boundary from the center of the disk, $q(\mathbf{x}, t)$ is the exact solution from (30), and $\tilde{q}(\mathbf{x}, t)$ is the approximate solution computed using the moving least squares extrapolation as described in Section 3.3. A convergence plot is shown in Fig. 2 and indicates that the method achieves between first and second order convergence rates. In both the evolved and prescribed level set cases, the RBF reconstruction yields errors that are more than an order of magnitude smaller than the MLS reconstructions.

4.2. Solid body rotation

In this section, we again consider diffusion from a point source, but instead apply a solid body rotation to the disk. In this case, the semi-Lagrangian sub-step contains a non-trivial integration in time, and therefore tests all components of the solver. We again prescribe the initial condition as in equation (30), and apply boundary conditions as in equation (31).

The computational domain $\mathcal{B}$ is $[-4, 4] \times [-4, 4]$ and is again discretized using N grid cells in each direction. We use a disk radius of $R = 1$ and initial center $\mathbf{x}_c(0) = (1.521, 1.503)$. We use a diffusion coefficient of $D = 0.1$. The prescribed velocity is $\mathbf{u} = 2\pi(-y, x)$. Simulations are run at an advective CFL number of 0.5 to a final time of $T = 1$. Fig. 3 shows the initial and final solution. We can see that the solution maintains symmetry throughout the simulation. Fig. 4 shows convergence rates for both RBF and MLS reconstructions. Figs. 4a and 4b show convergence rates in which the level set is prescribed. We see second order convergence rates in all norms for the prescribed level set case. Fig. 4c shows the convergence rate for the solution extrapolated to the boundary for the case in which the level set is evolved with the fluid velocity. In this case, we see approximately second order convergence rates. In both of these cases, the error for the RBF reconstruction is almost an order of magnitude lower than that of the MLS reconstruction.

4.3. Advection diffusion in oscillatory Couette flow

We now consider the advection and diffusion of a concentration inside a vesicle under oscillating Couette flow with no flux boundary conditions. Specifically, we specify the rotational component of the velocity field as

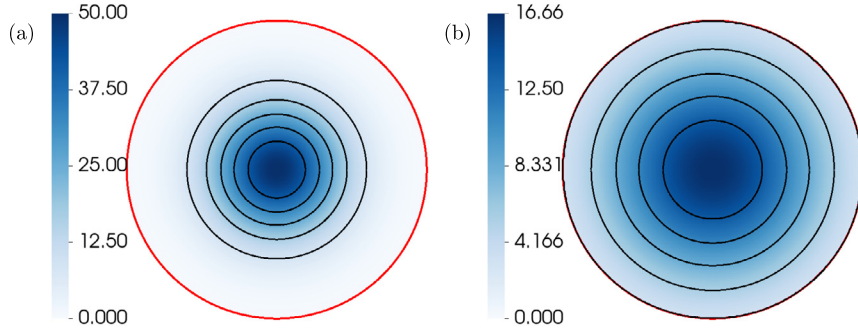


Fig. 3. Initial (a) and final (b) configuration for a diffusing concentration field initialized from a point source in a rigid body rotational flow field. The zero contour of ϕ is shown in red. Simulations are run at an advective CFL number of $C_{\text{CFL}} = 0.5$ to a final time of $T = 1$.

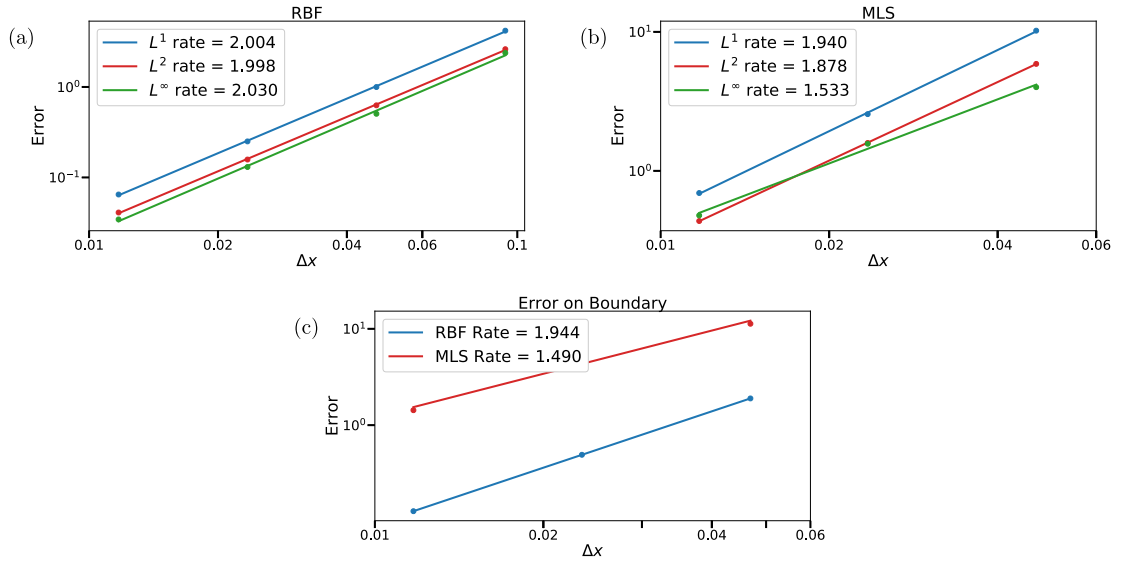


Fig. 4. Convergence rates for advecting and diffusion concentration field initialized from a point source in a solid body rotational flow using either a RBF (a) or MLS (b) reconstruction. Also shown is the convergence rate for the integral in equation (32) using both an RBF or MLS reconstruction (c). Simulations are run at an advective CFL number of $C_{\text{CFL}} = 0.5$ to a final time of $T = 1$.

$$u_{\theta}(r) = \left(ar + \frac{b}{r} \right) \sin(\pi t), \quad (33a)$$

$$a = \frac{\Omega_2 R_2^2 - \Omega_1 R_1^2}{R_2^2 - R_1^2}, \quad (33b)$$

$$b = \frac{(\Omega_1 - \Omega_2) R_1^2 R_2^2}{R_2^2 - R_1^2}, \quad (33c)$$

where R_1 and R_2 are the radii of the coaxial cylinders, and Ω_1 and Ω_2 are the angular velocities of the cylinders. Here, we set the radii to be $R_1 = 0.5$ and $R_2 = 3.75$ with respective angular velocities $\Omega_1 = 10.0$ and $\Omega = 1.0$.

The vesicle is described by the level set $\phi_v(\mathbf{x}, t)$ so that the initial condition is

$$\phi_v(\mathbf{x}, 0) = \|\mathbf{x} - \mathbf{x}_{v,c}\| - R \quad (34)$$

where $R = 1$ is the radius of the disk, and $\mathbf{x}_{v,c} = (1.521, 1.503)$ is the center of the disk. We initialize the concentration field q_v via

$$q_v(\mathbf{x}, 0) = (\cos(\pi \|\mathbf{x} - \mathbf{x}_{v,c}\|) + 1)^2 \quad (35)$$

For comparison, we define another concentration field $q_0(\mathbf{x}, t)$ outside the vesicle, but inside the cylinders. The second level set is defined by

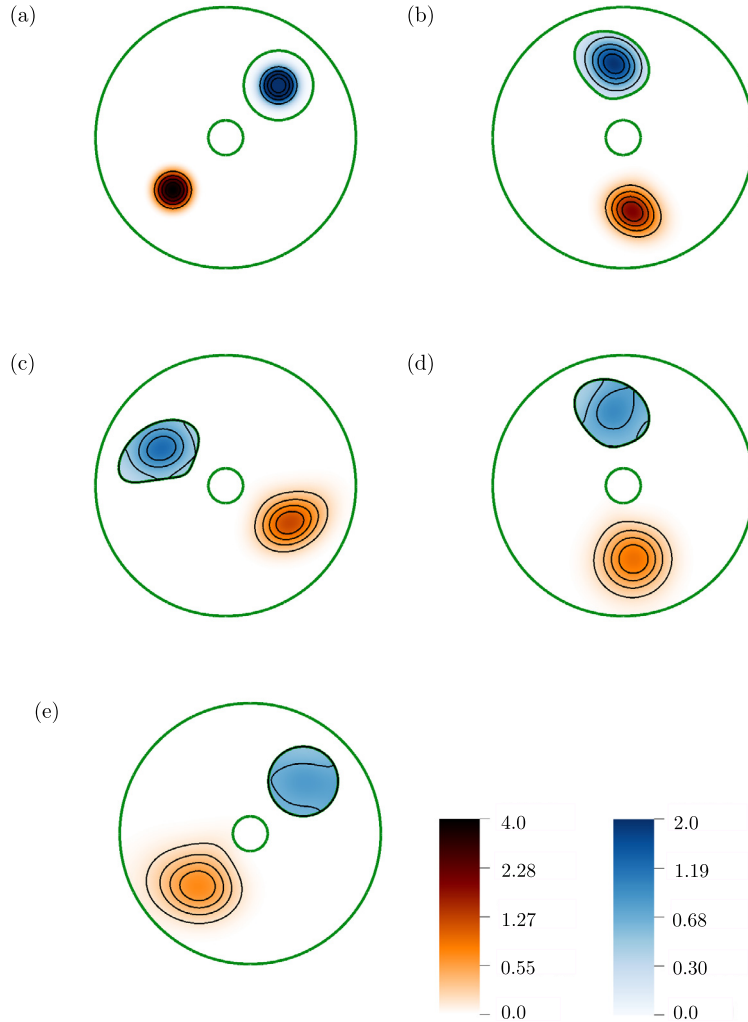


Fig. 5. Concentrations q_o (orange) and q_v (blue) at time points 0.0 (a), 0.5 (b), 1.0 (c), 1.5 (d), and 2.0 (e). The zero contour of the level set ϕ_o is shown in green. The diffusion coefficient is fixed at $D = 0.05$ with 1024 grid cells in each direction. We use a fixed time-step corresponding to a maximum CFL number of $C_{\text{CFL}} = 0.5$ to a final time of $T = 2$.

$$\phi_o(\mathbf{x}, t) = \max(-\phi_v(\mathbf{x}, t), \|\mathbf{x}\| - R_1, R_2 - \|\mathbf{x}\|). \quad (36)$$

The initial condition is given by

$$q_o(\mathbf{x}, 0) = \begin{cases} (\cos(\pi \|\mathbf{x} - \mathbf{x}_{o,c}\|) + 1)^2 & \text{if } \|\mathbf{x} - \mathbf{x}_{o,c}\| \leq 1 \\ 0 & \text{otherwise,} \end{cases} \quad (37)$$

in which $\mathbf{x}_{o,c} = -\mathbf{x}_{v,c}$.

The computational domain $\mathcal{B}$ is the box $[-4, 4] \times [-4, 4]$ and is discretized using N grid cells in each direction, and the diffusion coefficient for both the interior and exterior concentration fields is $D = 0.05$. Fig. 5 shows the solution at five different time values during the simulation. Over the course of the simulations, the concentration field inside the disk stays completely contained within the disk, while the outside concentration field is free to diffuse between the coaxial cylinders.

Figs. 6 and 7 show numerical convergence results for the concentrations inside the vesicle and outside the vesicle but inside the two cylinders respectively. We estimate the convergence rate r using Richardson extrapolation via

$$r = \log_2 \frac{\|q_{4h} - q_{2h}\|}{\|q_{2h} - q_h\|}, \quad (38)$$

in which q_h is the solution with a grid spacing of h . To compute the difference $\|q_{2h} - q_h\|$, we first interpolate the fine solution q_h onto the coarser grid q_{2h} . Because the interpolation from a fine grid to a coarse grid is nontrivial near cut-cells, we exclude all cut-cells and any cells that neighbor cut-cells in the computation of the norm. To assess the error on the

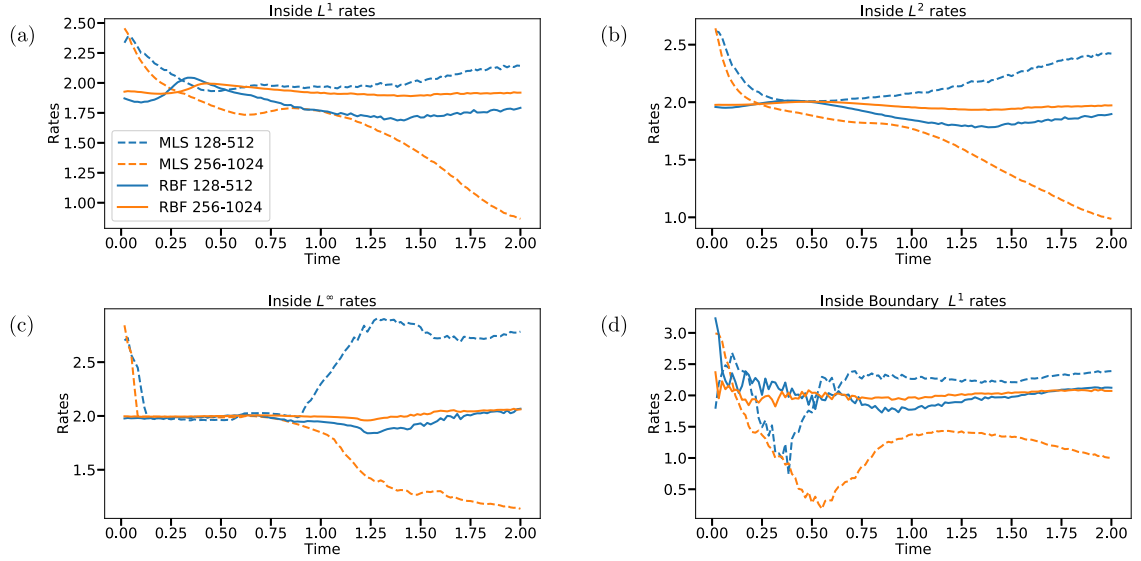


Fig. 6. Numerical convergence study for the interior of the disk in the L^1 (a), L^2 (b), and L^∞ (c) norms. The convergence rates are computed from simulations with $N = 128, 256, 512$ grid cells (blue) and $N = 256, 512, 1024$ grid cells (orange). Also shown is the convergence rate for the extrapolation of the solution to the boundary (d).

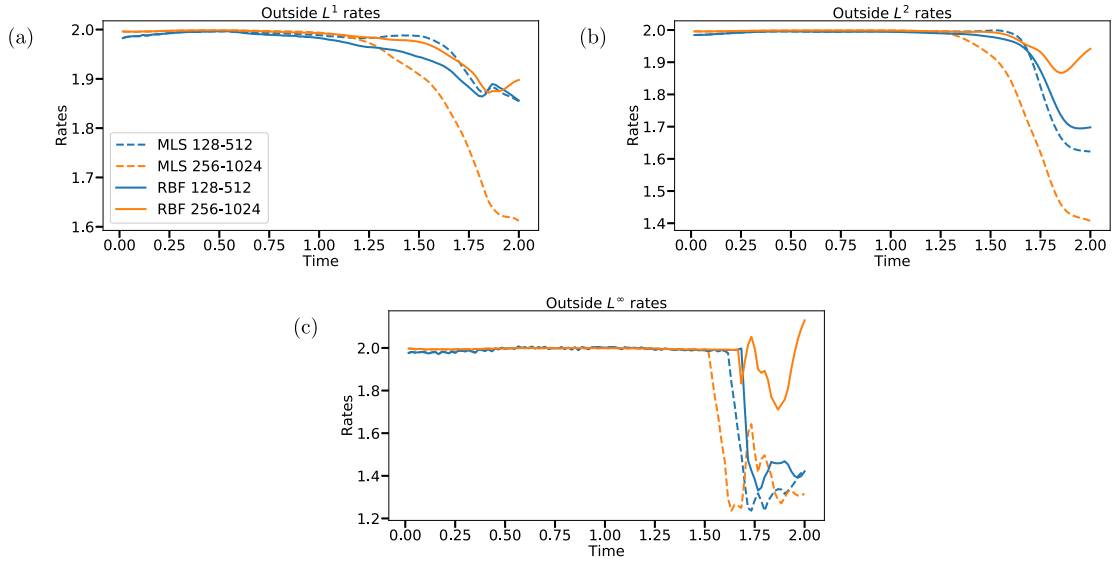


Fig. 7. Numerical convergence study for the exterior of the disk but inside the two cylinders in the L^1 (a), L^2 (b), and L^∞ (c) norms. The convergence study is computed from simulations with $N = 128, 256, 512$ grid cells and $N = 256, 512, 1024$ grid cells. The dip in convergence rates near the end of the simulation occurs when there is a significant value of q_0 near the boundaries.

cut-cells, we perform a reconstruction of the solution on the surface of the disk by extrapolating the solution to the disk in each cut-cell, then computing a cubic spline of these data points. Fig. 6d shows convergence of this reconstruction for the solution inside the vesicle. We obtain second order convergence rates using RBFs at all points in time for the finest grids. In contrast, the MLS reconstruction does not show optimal convergence rates for the range of grid spacings considered. We expect that under additional grid refinement the MLS approximation will settle to second order accuracy, but this is not achieved over the range of grid spacings considered herein. The dips in convergence rates in Fig. 7 occur when significant values of q_0 begin to appear in cut cells. Additionally, Figs. 8 and 9 show the norms of the differences. To quantify the difference in accuracy between MLS and RBF reconstructions, we estimate the error coefficient C in the expression

$$\|q_h - q\| \approx Ch^p, \quad (39)$$

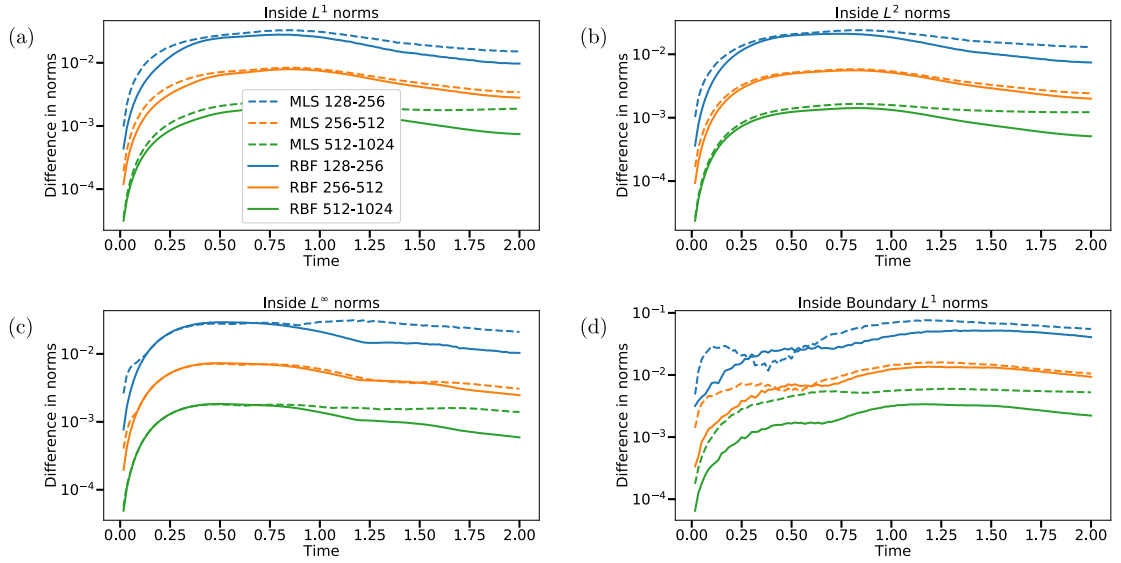


Fig. 8. Norms of solution differences for the concentration field inside the domain for the L^1 (a), L^2 (b), and L^∞ (c) norms. The dashed lines correspond to MLS reconstruction while the solid lines correspond to a RBF reconstruction. Also shown are the norms of solution differences for the extrapolation to the boundary (d).

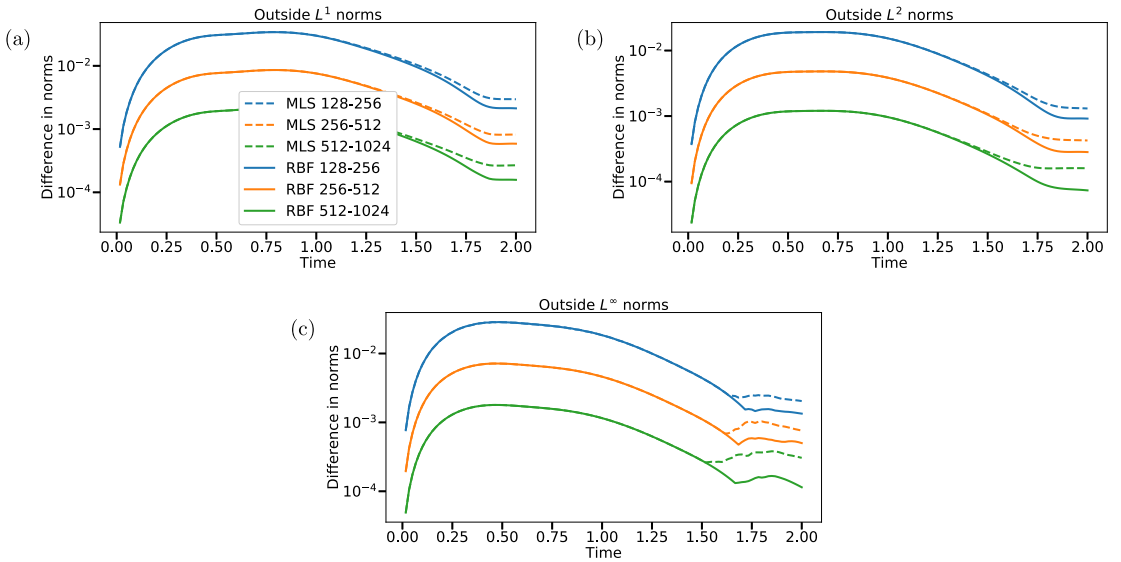


Fig. 9. Norms of solution differences for the concentration field outside the domain for the L^1 (a), L^2 (b), and L^∞ (c) norms. The dashed lines correspond to MLS reconstruction while the solid lines correspond to a RBF reconstruction.

in which q is the exact solution and p is the order of accuracy. We approximate C by comparing differences between two grids of refinement h and $2h$ through the following equation

$$C \approx \frac{\|q_h - q_{2h}\|}{h^p (1 - 2^p)}. \quad (40)$$

For both MLS and RBF reconstructions, we have $p = 2$. Table 1 shows our estimates of the C coefficient for both reconstructions. In all cases, the RBF reconstruction is more accurate.

4.4. Choice of time step size

Previous sections use time step sizes based on a CFL number of 0.5. We note that each substep consists of an unconditionally stable method. While the affect on stability of the splitting error is unclear, we briefly explore the choice of time

Table 1

The leading order error coefficient C for both MLS and RBF reconstructions is estimated using equation (40). In all cases, the RBF reconstruction yields superior accuracy.

	q_o			q_v		
	L^1 norm	L^2 norm	L^∞ norm	L^1 norm	L^2 norm	L^∞ norm
C_{MLS}	41.01	26.71	30.37	5.85	3.49	6.68
C_{RBF}	16.20	11.12	12.88	3.44	1.60	2.50

Table 2

The leading order error coefficient C as estimated by equation (40) as a function of the CFL number C_{CFL} . We see a decrease in error as the CFL number is increased.

C_{CFL}	q_o					
	0.5	1.0	1.5	2.0	2.5	3.0
L^1 norm	16.20	12.44	10.08	8.18	7.04	6.98
L^2 norm	11.12	8.95	7.47	6.38	5.69	5.25
L^∞ norm	12.88	11.81	10.55	9.43	8.46	7.61
C_{CFL}	q_v					
	0.5	1.0	1.5	2.0	2.5	3.0
L^1 norm	3.44	3.01	2.86	2.64	2.40	2.18
L^2 norm	1.60	1.47	1.38	1.24	1.08	0.95
L^∞ norm	2.50	2.65	2.49	2.17	1.81	1.60

step size in this section. We again use the oscillatory Couette flow example with radial basis function reconstructions as in the previous section; however, we now vary the CFL number. Recall we defined the CFL number as $C_{\text{CFL}} = \frac{\Delta x}{|\mathbf{u}| \Delta t}$ in which $|\mathbf{u}|$ is the maximum velocity over both space and time so that we used a fixed time step size throughout the entire simulation. As before, we estimate the C coefficient (see equation (40)), and the values are reported in Table 2. We observe a decrease in the error as the CFL number increases. This is not surprising because the error term for the semi-Lagrangian method contains a term that scales like $\mathcal{O}(\frac{1}{\Delta t})$, so that the error can decrease for larger timesteps [12], although this reduction in error will eventually break down.

4.5. Interaction of two concentrations in oscillatory Couette flow

We now consider the case of two fluid phase chemicals advecting and diffusing on two different domains, but interacting through a common boundary. Specifically, we consider oscillatory Couette flow of two chemicals, one of which is contained within a vesicle that passively advects with the flow and is initially in the shape of a disk. The other chemical exists outside the vesicle, but between the disks defining the domain of interest. Specifically, we solve the equations

$$\frac{\partial q_v}{\partial t} + \mathbf{u} \cdot \nabla q_v = D \Delta q_v, \quad \mathbf{x} \in \Omega_v \quad (41a)$$

$$-D \frac{\partial q_v}{\partial \mathbf{n}} = \kappa (q_v - q_o), \quad \mathbf{x} \in \Gamma_v \quad (41b)$$

$$\frac{\partial q_o}{\partial t} + \mathbf{u} \cdot \nabla q_o = D \Delta q_o, \quad \mathbf{x} \in \Omega_o \quad (41c)$$

$$-D \frac{\partial q_o}{\partial \mathbf{n}} = \kappa (q_o - q_v), \quad \mathbf{x} \in \Gamma_v \quad (41d)$$

$$-D \frac{\partial q_o}{\partial \mathbf{n}} = 0, \quad \mathbf{x} \in \Gamma_o \setminus \Gamma_v, \quad (41e)$$

in which $\mathbf{u}$ is defined in the previous section and Ω_v and Ω_o are the domains for the concentration inside and outside the vesicle, respectively. The initial concentration for q_v is given by equation (35), and for q_o is initially uniformly zero.

We modify the time discretization of the diffusion step to use a modified trapezoidal rule for the boundary conditions. We first solve equation (15) for initial approximations $\tilde{q}_o$ and $\tilde{q}_v$ using explicit approximations for the boundary conditions

$$-D \frac{\partial \tilde{q}_o}{\partial \mathbf{n}} = \kappa (\tilde{q}_o - q_v^n), \quad \text{and} \quad (42)$$

$$-D \frac{\partial \tilde{q}_v}{\partial \mathbf{n}} = \kappa (\tilde{q}_v - q_o^n). \quad (43)$$

We then solve equation (15) for q_o^{n+1} and q_v^{n+1} again using the intermediate result when evaluating the boundary conditions

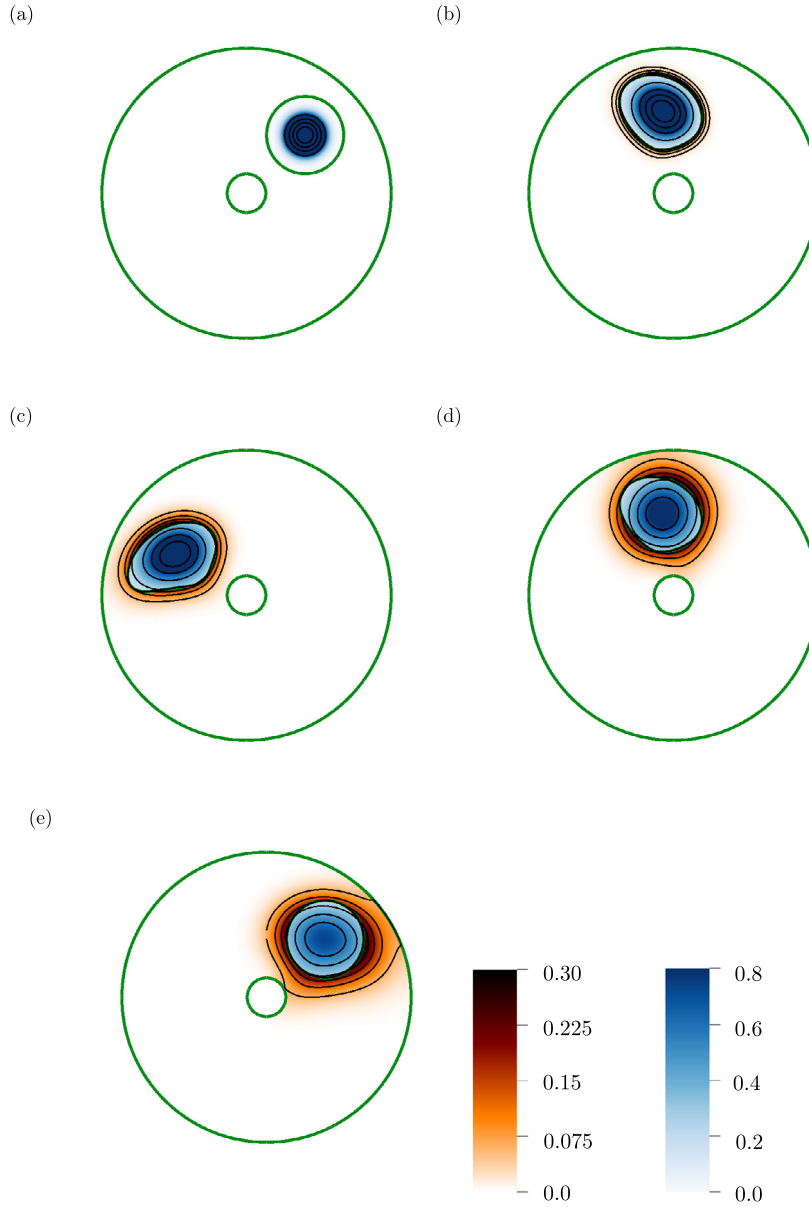


Fig. 10. Concentrations q_o (orange) and q_v (blue) at times 0.0 (a), 0.5 (b), 1.0 (c), 1.5 (d), and 2.0 (e). The zero contour of the level set ϕ_o is shown in green. The diffusion coefficient is fixed at $D = 0.05$ with 1024 grid cells in each direction. We use a fixed time-step corresponding to a maximum CFL number of $C_{\text{CFL}} = 0.5$ to a final time of $T = 2$.

$$-D \frac{\partial q_o^{n+1}}{\partial \mathbf{n}} = \kappa (q_o^{n+1} - \tilde{q}_v), \text{ and} \quad (44)$$

$$-D \frac{\partial q_v^{n+1}}{\partial \mathbf{n}} = \kappa (q_v^{n+1} - \tilde{q}_o), \quad (45)$$

in which $\tilde{q}_v$ and $\tilde{q}_o$ are the results from the intermediate result.

For this example, we perform only RBF reconstructions at the boundaries. We set $\kappa = 1$. The remaining parameters are the same as in the previous section. Concentration values for various points in time are shown in Fig. 10. As q_v reaches the boundary Γ_v , it is converted to q_o at a rate proportional to the difference in concentrations.

As in the previous section, we perform a numerical convergence study for both the interior and exterior fields. Figs. 11 and 12 show the convergence rates as a function of time. The convergence rates as we refine the grid appear to be converging toward second order, although the method has not yet settled down into its asymptotic regime. We also perform a convergence study for the solution extrapolated to the boundary, which is shown in Fig. 11d. We again see approximate

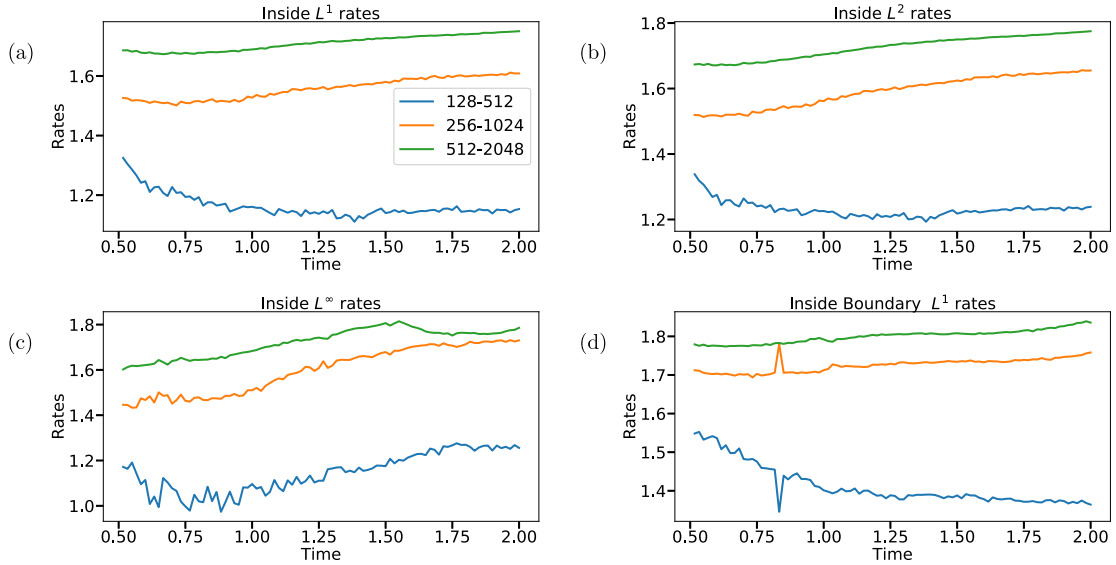


Fig. 11. Numerical convergence study for the interior of the disk in the L^1 (a), L^2 (b), and L^∞ (c) norms. The convergence rates are computed from simulations with $N = 128, 256$, and 512 grid cells (blue), $N = 256, 512$, and 1024 grid cells (orange), and $N = 512, 1024$, and 2048 grid cells (green). Also shown are the convergence rates for the extrapolation of the solution to the boundary (d).

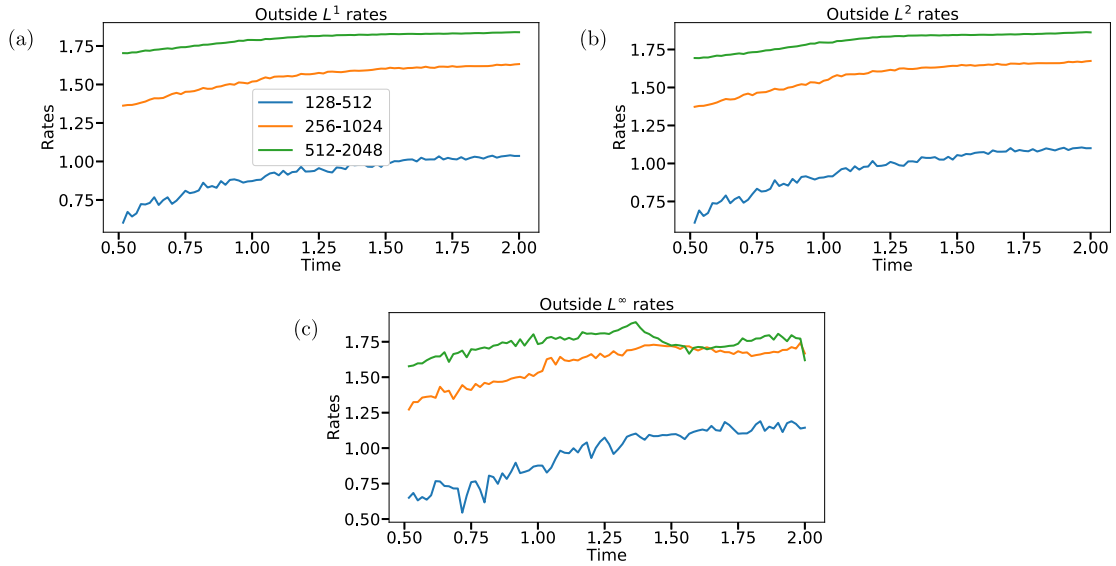


Fig. 12. Numerical convergence study for the exterior of the disk but inside the two cylinders in the L^1 (a), L^2 (b), and L^∞ (c) norms. The convergence study is computed from simulations with $N = 128, 256$, and 512 grid cells (blue), $N = 256, 512$, and 1024 grid cells (orange), and $N = 512, 1024$, and 2048 grid cells (green).

second order rates for this reconstruction. A plot of the norms of the solution differences is shown in Figs. 13 and 14. The differences in the finest grid show a difference of less than one percent at the final time.

To assess conservation, we perform a convergence study on the total amount of the two concentration fields. Fig. 15 shows the total amounts of both fields as a function of time, while the sum of both amounts remains relatively constant. Despite this method not being conservative, we see a loss of less than one percent over the course of the finest simulation. The total amount of concentration converges at a rate that is between first and second order accuracy.

5. Conclusions

We have presented a numerical method to simulate advection-diffusion problems with Robin boundary conditions on irregular, evolving domains. The method shows second order convergence in the L^1 , L^2 , and L^∞ norms. We have also

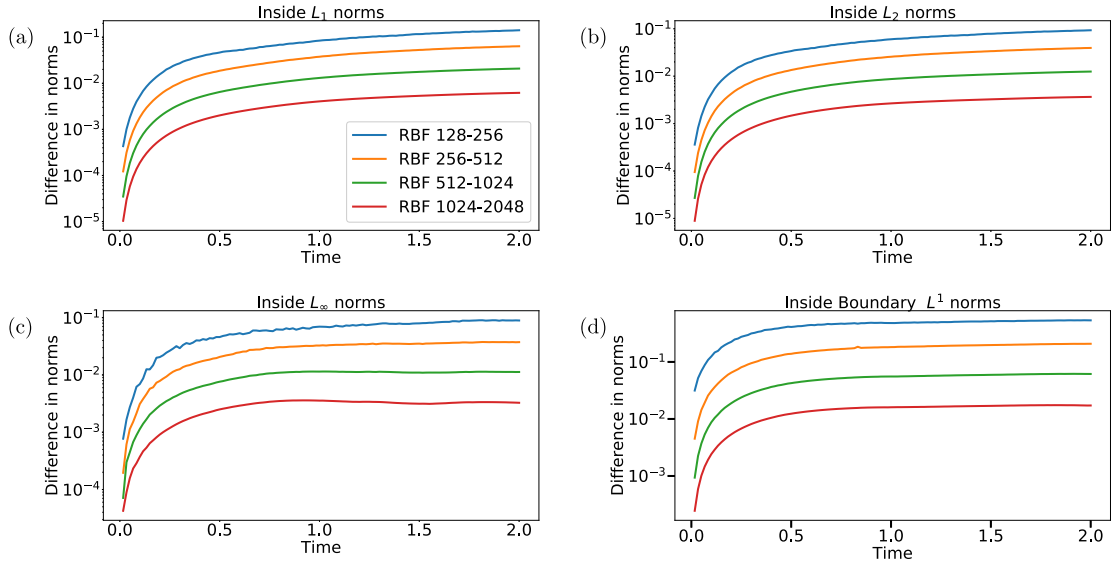


Fig. 13. Norms of solution differences for the concentration field inside the domain for the L^1 (a), L^2 (b), and L^∞ (c) norms. Also shown are the norms of the differences between solutions extrapolated to the boundary (d).

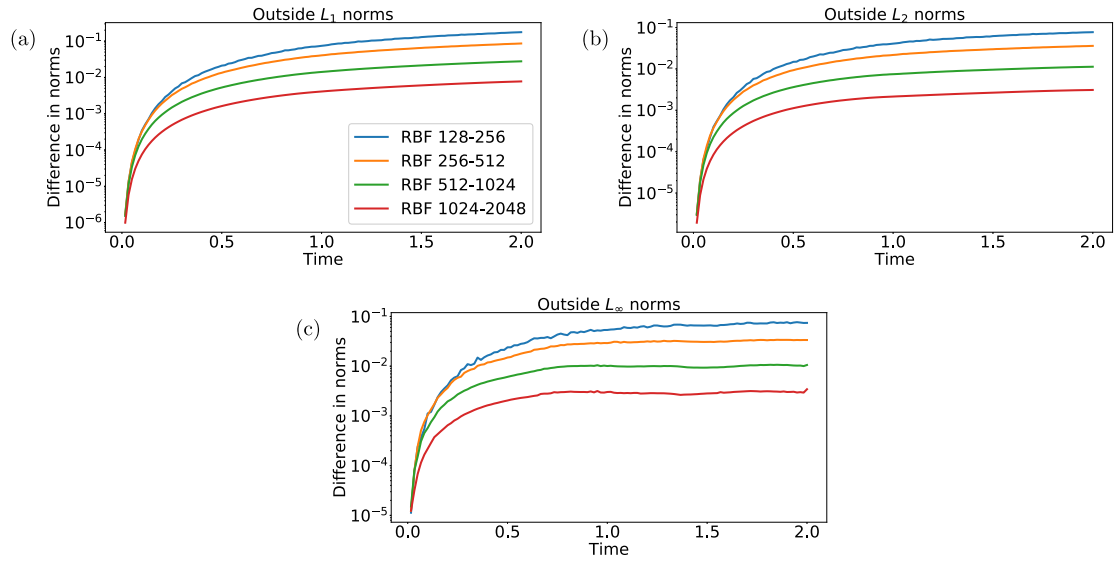


Fig. 14. Norms of solution differences for the concentration field outside the domain for the L^1 (a), L^2 (b), and L^∞ (c) norms.

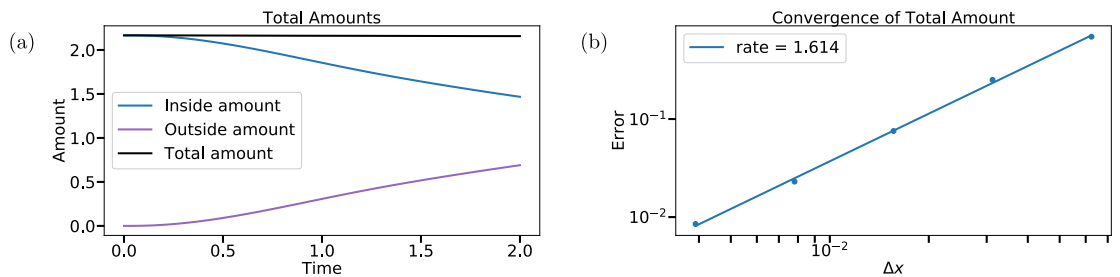


Fig. 15. Total concentration of both q_v and q_o for interactions with a oscillatory Couette flow (a). The total amount converges at a rate that is between first and second order (b).

demonstrated the ability to accurately reconstruct solution values on the boundary, achieving second order accurate results. We have further demonstrated the method to be able to capture transformation of one concentration into another concentration with all interaction mediated through boundary conditions. Although only two dimensional tests are presented, we expect this method to be easily extended to three spatial dimensions. The use of radial basis functions when compared to moving least squares gives optimal convergence rates and more accurate solutions for the grid spacings considered. Z-splines were used in this study for computational efficiency, although any suitable reconstruction procedure should work. This method has many possible applications, including models with chemical transport in evolving bodies, such as esophageal transport, oxygen flow in the lungs, and blood flow.

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

The authors thank Dr. Varun Shankar for helpful discussions on both radial basis functions and semi-Lagrangian schemes. This research was supported through NIH award 5U01HL143336 and NSF awards DMS 1664645, OAC 1450327, and OAC 1931516.

References

- [1] V. Arias, D. Bochkov, F. Gibou, Poisson equations in irregular domains with Robin boundary conditions - solver with second-order accurate gradients, *J. Comput. Phys.* 365 (2018) 1–6.
- [2] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. Gropp, D. Karpeyev, D. Kaushik, M. Knepley, D. May, L.C. McInnes, R.T. Mills, T. Munson, K. Rupp, P. Sanan, B. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc Users Manual, Tech. Rep. ANL-95/11 - Revision 3.13, Argonne National Laboratory, 2020.
- [3] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. Gropp, D. Karpeyev, D. Kaushik, M. Knepley, D. May, L.C. McInnes, R.T. Mills, T. Munson, K. Rupp, P. Sanan, B. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc web page, <https://www.mcs.anl.gov/petsc>, 2019.
- [4] S. Balay, W.D. Gropp, L.C. McInnes, B.F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A.M. Bruaset, H.P. Langtangen (Eds.), *Modern Software Tools in Scientific Computing*, Birkhäuser Press, 1997, pp. 163–202.
- [5] V. Bayona, Comparison of moving least squares and RBF+poly for interpolation and derivative approximation, *J. Sci. Comput.* 81 (1) (2019) 486–512.
- [6] J. Becerra-Sagredo, C. Málaga, F. Mandujano, Moments preserving and high-resolution semi-Lagrangian advection scheme, *SIAM J. Sci. Comput.* 38 (4) (2016), A2141–A2161.
- [7] P.-A. Billat, E. Roger, S. Faure, F. Lagarce, Models for drug absorption from the small intestine: where are we and where are we going?, *Drug Discov. Today* 22 (5) (2017) 761–775.
- [8] D. Bochkov, F. Gibou, Solving Poisson-type equations with Robin boundary conditions on piecewise smooth interfaces, *J. Comput. Phys.* 376 (2019) 1156–1198.
- [9] F. Bouchon, G.H. Peichl, An immersed interface technique for the numerical solution of the heat equation on a moving domain, in: *Numerical Mathematics and Advanced Applications 2009*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 181–189.
- [10] M. Chai, K. Luo, C. Shao, H. Wang, J. Fan, A finite difference discretization method for heat and mass transfer with Robin boundary conditions on irregular domains, *J. Comput. Phys.* 400 (2020) 108890.
- [11] D.R. Durran, *Semi-Lagrangian Methods*, 1999, pp. 303–333.
- [12] S.J. Fletcher, *Semi-Lagrangian Advection Methods and Their Applications in Geoscience*, Elsevier, 2019.
- [13] N. Flyer, B. Fornberg, V. Bayona, G.A. Barnett, On the role of polynomials in RBF- FD approximations: I. Interpolation and accuracy, *J. Comput. Phys.* 321 (2016) 21–38.
- [14] A.L. Fogelson, K.B. Neeves, Fluid mechanics of blood clot formation, *Annu. Rev. Fluid Mech.* 47 (1) (2015) 377–403.
- [15] E. Gloede, J.A. Cichocki, J.B. Baldino, J.B. Morris, A validated hybrid computational fluid dynamics-physiologically based pharmacokinetic model for respiratory tract vapor absorption in the human and rat and its application to inhalation dosimetry of diacetyl, *Toxicol. Sci.* 123 (1) (2011) 231–246.
- [16] B.E. Griffith, R.D. Hornung, D.M. McQueen, C.S. Peskin, An adaptive, formally second order accurate version of the immersed boundary method, *J. Comput. Phys.* 223 (1) (2007) 10–49.
- [17] B.E. Griffith, N.A. Patankar, Immersed methods for fluid-structure interaction, *Annu. Rev. Fluid Mech.* 52 (1) (2020) 421–448.
- [18] Á. Helgadóttir, Y.T. Ng, C. Min, F. Gibou, Imposing mixed Dirichlet-Neumann-Robin boundary conditions in a level-set framework, *Comput. Fluids* 121 (2015) 68–80.
- [19] R.D. Hornung, S.R. Kohn, Managing application complexity in the SAMRAI object- oriented framework, *Concurr. Comput.* 14 (5) (2002) 347–368.
- [20] IBAMR: an adaptive and distributed-memory parallel implementation of the immersed boundary method. Version 0.8.0, <https://github.com/IBAMR/IBAMR>.
- [21] A. Iske, M. Käser, Conservative semi-Lagrangian advection on adaptive unstructured meshes, *Numer. Methods Partial Differ. Equ.* 20 (3) (2004) 388–411.
- [22] B. Kadoch, D. Kolomenskiy, P. Angot, K. Schneider, A volume penalization method for incompressible flows and scalar advection-diffusion with moving obstacles, *J. Comput. Phys.* 231 (12) (2012) 4365–4383.
- [23] R. Klein, K.R. Bates, N. Nikiforakis, Well-balanced compressible cut-cell simulation of atmospheric flow, *Philos. Trans. - Royal Soc., Math. Phys. Eng. Sci.* 367 (1907) (2009) 4559–4575.
- [24] E.M. Kolahdouz, A.P.S. Bhalla, B.A. Craven, B.E. Griffith, An immersed interface method for discrete surfaces, *J. Comput. Phys.* 400 (2020) 108854.
- [25] K. Leiderman, A.L. Fogelson, Grow with the flow: a spatial-temporal model of platelet deposition and blood coagulation under flow, *Math. Med. Biol.* 28 (1) (2011) 47–84.
- [26] M. Lentine, J.T. Grétarsson, R. Fedkiw, An unconditionally stable fully conservative semi-Lagrangian method, *J. Comput. Phys.* 230 (8) (2011) 2857–2879.
- [27] X. Li, J. Lowengrub, A. Ratz, A. Voigt, Solving pdes in complex geometries, *Commun. Math. Sci.* 7 (1) (2009) 81–107.
- [28] J.-k. Liu, Z.-s. Zheng, Efficient high-order immersed interface methods for heat equations with interfaces, *Appl. Math. Mech.* 35 (9) (2014) 1189–1202.
- [29] S. May, M. Berger, An explicit implicit scheme for cut cells in embedded boundary meshes, *J. Sci. Comput.* 71 (3) (2017) 919–943.

- [30] C. Min, F. Gibou, Geometric integration over irregular domains with application to level-set methods, *J. Comput. Phys.* 226 (2) (2007) 1432–1443.
- [31] N. Nangia, B.E. Griffith, N.A. Patankar, A.P.S. Bhalla, A robust incompressible Navier-Stokes solver for high density ratio multiphase flows, *J. Comput. Phys.* 390 (2019) 548–594.
- [32] A. Pacheco-Vega, J.R. Pacheco, T. Rodić, A general scheme for the boundary conditions in convective and diffusive heat transfer with immersed boundary methods, *J. Heat Transf.* 129 (11) (2007) 1506–1516.
- [33] K.S. Pang, Modeling of intestinal drug absorption: roles of transporters and metabolic enzymes (for the Gillette review series), *Drug Metab. Dispos.* 31 (12) (2003) 1507–1519.
- [34] J. Papac, F. Gibou, C. Ratsch, Efficient symmetric discretization for the Poisson, heat and Stefan-type problems with Robin boundary conditions, *J. Comput. Phys.* 229 (3) (2010) 875–889.
- [35] C.S. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [36] S. Qadeer, B.E. Griffith, The smooth forcing extension method: a high-order technique for solving elliptic equations on complex domains, *J. Comput. Phys.* 439 (2021) 110390.
- [37] G. Russo, F. Filbet, Semilagrangian schemes applied to moving boundary problems for the BGK model of rarefied gas dynamics, *Kinet. Relat. Models* 2 (1) (2009) 231–250.
- [38] L. Schneiders, D. Hartmann, M. Meinke, W. Schröder, An accurate moving boundary formulation in cut-cell methods, *J. Comput. Phys.* 235 (2013) 786–809.
- [39] J.D. Schroeter, J.S. Kimbell, B. Asgharian, Analysis of particle deposition in the turbinate and olfactory regions using a human nasal computational fluid dynamics model, *J. Aerosol Med.* 19 (3) (2006) 301–313.
- [40] D.B. Stein, R.D. Guy, B. Thomas, Immersed Boundary Smooth Extension (IBSE): a high-order method for solving incompressible flows in arbitrary smooth domains, *J. Comput. Phys.* 335 (2017) 155–178.
- [41] G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* 5 (3) (1968) 506–517.
- [42] W. Strychalski, D. Adalsteinsson, T. Elston, A cut-cell method for simulating spatial models of biochemical reaction networks in arbitrary geometries, *Commun. Appl. Math. Comput. Sci.* 5 (1) (2010) 31–53.
- [43] G. Tian, P.W. Longest, Development of a CFD boundary condition to model transient vapor absorption in the respiratory airways, *J. Biomech. Eng.* 132 (5) (2010).
- [44] J.D. Towers, A source term method for Poisson problems on irregular domains, *J. Comput. Phys.* 361 (2018) 424–441.
- [45] S. Xu, Z.J. Wang, An immersed interface method for simulating the interaction of a fluid with moving boundaries, *J. Comput. Phys.* 216 (2) (2006) 454–493.
- [46] L. Yao, A.L. Fogelson, Simulations of chemical transport and reaction in a suspension of cells I: an augmented forcing point method for the stationary case, *Int. J. Numer. Methods Fluids* 69 (11) (2012) 1736–1752.
- [47] H.C. Yu, H.Y. Chen, K. Thornton, Extended smoothed boundary method for solving partial differential equations with general boundary conditions on complex boundaries, *Model. Simul. Mater. Sci. Eng.* 20 (7) (2012).
- [48] L.X. Yu, E. Lipka, J.R. Crison, G.L. Amidon, Transport approaches to the biopharmaceutical design of oral drug delivery systems: prediction of intestinal absorption, *Adv. Drug Deliv. Rev.* 19 (3) (1996) 359–376.
- [49] N. Zhang, Z.C. Zheng, S. Eckels, Study of heat-transfer on the surface of a circular cylinder in flow using an immersed-boundary method, *Int. J. Heat Fluid Flow* 29 (6) (2008) 1558–1566.