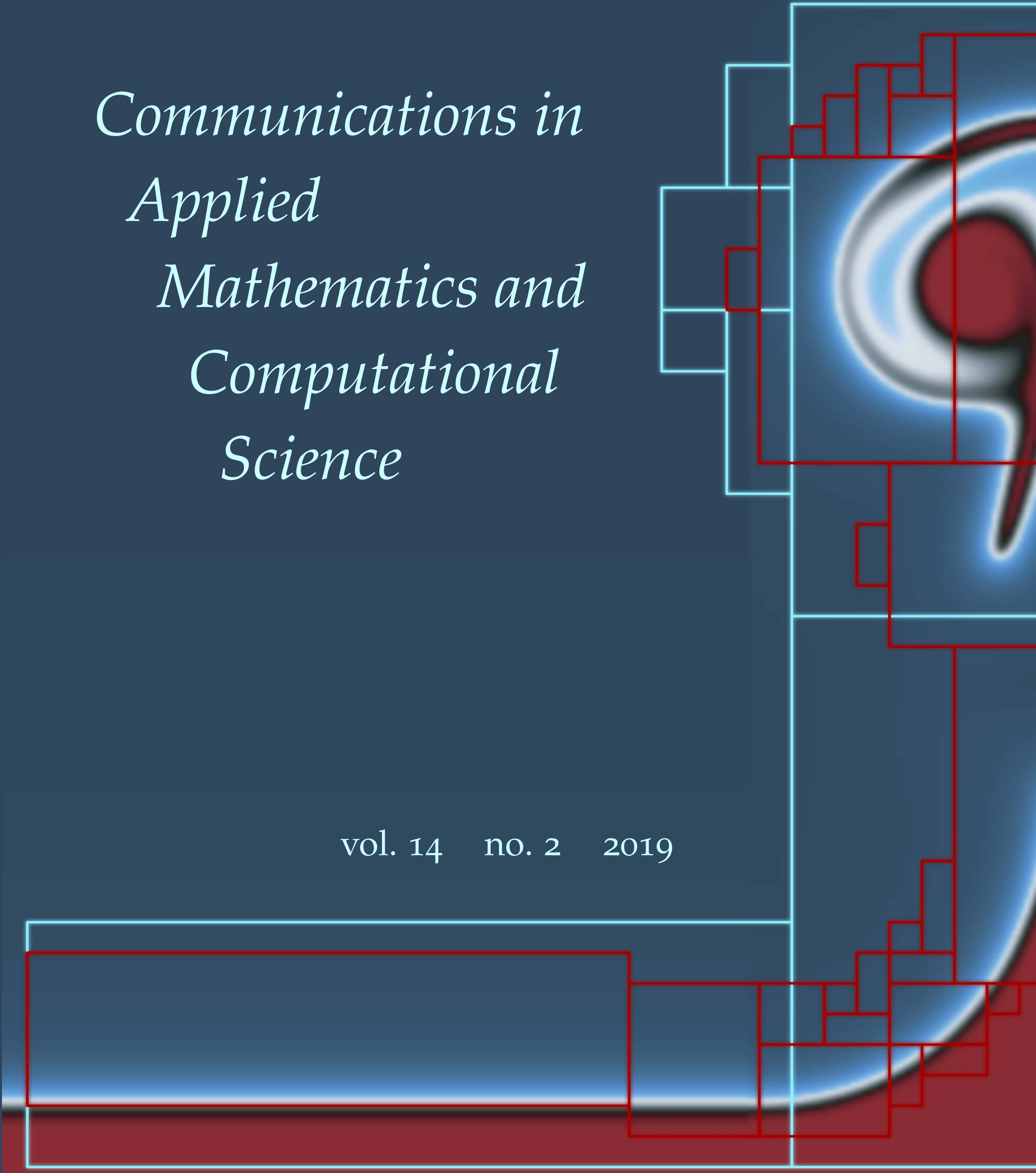
Communications in Applied Mathematics and Computational Science



vol. 14 no. 2 2019

msp

Communications in Applied Mathematics and Computational Science

msp.org/camcos

See inside back cover or msp.org/camcos for submission instructions.

The subscription price for 2019 is US $105/year for the electronic version, and $155/year (+$15, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscriber address should be sent to MSP.

Communications in Applied Mathematics and Computational Science (ISSN 2157-5452 electronic, 1559-3940 printed) at Mathematical Sciences Publishers, 798 Evans Hall #3840, c/o University of California, Berkeley, CA 94720-3840, is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

CAMCoS peer review and production are managed by EditFLOW® from MSP.

PUBLISHED BY

mathematical sciences publishers

nonprofit scientific publishing

http://msp.org/

COMM. APP. MATH. AND COMP. SCI.
Vol. 14, No. 2, 2019
dx.doi.org/10.2140/camcos.2019.14.121

SIMPLE SECOND-ORDER FINITE DIFFERENCES FOR ELLIPTIC PDES WITH DISCONTINUOUS COEFFICIENTS AND INTERFACES

CHUNG-NAN TZOU AND SAMUEL N. STECHMANN

In multiphase fluid flow, fluid-structure interaction, and other applications, partial differential equations (PDEs) often arise with discontinuous coefficients and singular sources (e.g., Dirac delta functions). These complexities arise due to changes in material properties at an immersed interface or embedded boundary, which may have an irregular shape. Consequently, the solution and its gradient can be discontinuous, and numerical methods can be difficult to design. Here a new method is presented and analyzed, using a simple formulation of one-dimensional finite differences on a Cartesian grid, allowing for a relatively easy setup for one-, two-, or three-dimensional problems. The derivation is relatively simple and mainly involves centered finite difference formulas, with less reliance on the Taylor series expansions of typical immersed interface method derivations. The method preserves a sharp interface with discontinuous solutions, obtained from a small number of iterations (approximately five) of solving a symmetric linear system with updates to the right-hand side. Second-order accuracy is rigorously proven in one spatial dimension and demonstrated through numerical examples in two and three spatial dimensions. The method is tested here on the variable-coefficient Poisson equation, and it could be extended for use on time-dependent problems of heat transfer, fluid dynamics, or other applications.

1. Introduction

In many applications, partial differential equations (PDEs) arise with discontinuous coefficients and singular sources (e.g., Dirac delta functions). These complexities often arise due to changes in material properties at an interface or immersed boundary, which may have an irregular shape; see Figure 1. For example, the immersed boundary may be a rigid or flexible structure, such as a heart valve [10], or the immersed interface may separate two fluids as in gas bubbles or liquid droplets [34]. Our own interest was motivated by recently derived equations for atmospheric dynamics, in

MSC2010: 65M06, 76T99, 35J05.
Keywords: sharp interface, immersed boundary method, immersed interface method, ghost fluid method, jump conditions, phase changes.

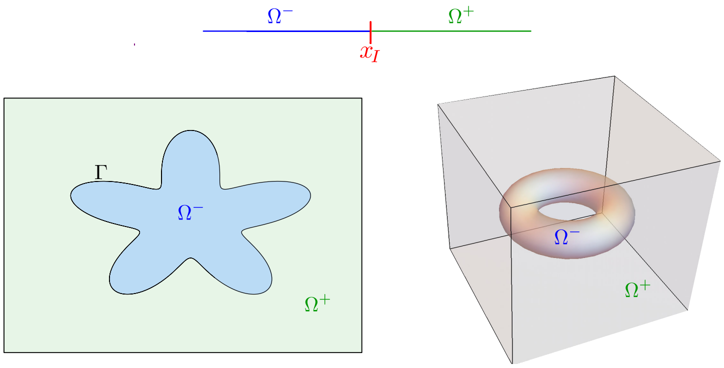


Figure 1. Examples of interfaces separating two regions Ω^- and Ω^+ in (top) 1D, (bottom left) 2D, and (bottom right) 3D.

the limit of rapid rotation and strong (moist) stratification, including phase changes of water and phase interfaces between cloudy and noncloudy regions [33].

For PDEs with such complexities, numerical methods can be challenging to design. Elliptic PDEs are a common test case, and they often form an important component of time-dependent systems. Many methods have been proposed using finite element methods [2; 12], finite volume methods [13; 4], and finite difference methods. Each of these approaches can be valuable in different situations, depending on priorities of computational efficiency, ease of implementation, etc. A primary goal of the present paper is simplicity, and finite difference methods, with Cartesian grids, are perhaps the simplest class of methods. Therefore, for comparison, we next describe some finite difference methods in more detail.

The immersed boundary method (IBM) was introduced in the pioneering work of Peskin [29; 30; 31]. The IBM is simple and efficient and has been applied to a variety of problems with three-dimensional fluid flow [10; 14]. In the IBM approach, the effect of the immersed boundary is represented as a forcing function applied to the fluid. Ideally, the forcing should be singular and the solution should have discontinuities. However, the IBM uses a smoothed version of a Dirac delta function, which introduces some smearing near the boundary or interface and causes the solution to be continuous. The method was originally designed with first-order accuracy, and it has been extended to be “formally” second-order accurate [16; 11; 28; 6; 7], although the “formal” second-order accuracy holds only in the case that the forcing is sufficiently smooth, not in the case of a nearly singular forcing.

The immersed interface method (IIM) was developed to produce improvements such as second-order accuracy and a solution with a sharp discontinuity and no smearing at the interface [19; 20]. The method is derived by allowing an extended stencil, beyond the standard stencil for the Laplacian operator, to be used at grid points near the interface; for the extended stencil, the finite-difference weights are then found by the method of undetermined coefficients, with constraints on the coefficients being chosen to achieve the desired local truncation error based on Taylor series. The extended stencil of the IIM must be chosen with care in order to avoid instability [8; 21; 5], since the IIM linear operator is not symmetric. One approach is to carefully construct the IIM operator to satisfy a discrete maximum principle by using constrained quadratic optimization techniques [21; 5].

While the IIM has been implemented in multidimensional fluid flow problems, the formulation is complicated by the need for derivations of many spatial and temporal jump conditions, and also derivatives of jump conditions [22; 18; 38; 37]. Many other versions of the IIM with different derivations have been developed [36; 3; 32; 17], and some are discussed further in Section 5 below. In the present paper, one distinguishing feature is that the present derivation involves the relatively simple use of centered finite difference formulas, without the need for derivatives of jump conditions, and with less reliance on the Taylor series expansions of typical IIM derivations. Such simplifications to the derivations should contribute to enhanced ease of use on three-dimensional problems.

The ghost fluid method (GFM) is another method that produces a solution with a sharp discontinuity and no smearing at the interface [23]. While it is only first-order accurate, the GFM is simple to formulate and implement, and it is efficient for problems with three-dimensional multiphase fluid flow [15; 35]. Another advantageous property is that the GFM finite difference operator is symmetric, which allows the use of conjugate gradient algorithms and guarantees robustness of the method.

In the present paper, the goal is to design a method with the advantageous properties of the GFM — sharp interface, easy to formulate and implement, efficient for use on three-dimensional problems, and utilization of a symmetric matrix — while also achieving the possibility of second-order accuracy. The simple formulation here (Section 2) uses elementary finite differences along one-dimensional coordinates, and the resulting linear system can be written with the same symmetric matrix as the GFM but with corrections to the right-hand side that yield second-order accuracy. The right-hand-side corrections are determined iteratively, which is the main new computational expense beyond the GFM. Note that, while this interesting algorithmic connection exists with the GFM, the derivations of the GFM and the present method are quite different; the present method is derived using finite differences (with explicit estimates of local truncation error from finite difference formulas), whereas the GFM and its error and convergence are based on a weak formulation of the problem [24]. Example solutions with the present method are shown for one-dimensional (1D), two-dimensional (2D), and three-dimensional (3D) problems (Section 3). A small, fixed number of iterations (≈ 5) is shown to be sufficient for achieving a second-order accurate solution (Section 4), which suggests the present methods may be efficient enough for use on complex three-dimensional fluid flow. Conclusions and further comparisons with the formulations of other methods [36; 3; 32; 17; 26; 27] are discussed in Sections 5 and 6.

Given that many previous methods have been proposed for this problem over many years, it is worthwhile to emphasize one of the main distinguishing features of the present method: a simple derivation and setup. The derivation here is mainly achieved using centered finite difference formulas, so it is relatively easy to formulate and set up the method, even in 3D. At the same time, the method does utilize a small number of iterations, so it may have a greater computational expense than some other methods (unless one could propose a more sophisticated and faster iterative procedure, a direction which we have not yet pursued exhaustively). In summary, in terms of practical use, the simple derivation and formulation should be useful for applications where one is less concerned with achieving the least possible expense of the computation itself and more concerned with minimizing the time and effort needed to initially design and code the method.

2. Numerical methods

In this section, the numerical methods are derived for 1D, 2D, and 3D equations in Sections 2.1, 2.2, and 2.3, respectively. A rigorous proof of second-order convergence is presented in Section 2.1.2 for the 1D case.

2.1. ***One dimension.*** Consider a one (spatial) dimensional domain Ω divided into subdomains Ω^+ and Ω^- by an interface Γ. The variable coefficient Poisson equation on each subdomain reads

$$(\beta u_x)_x = f(x) \quad \text{for } x \in \Omega \setminus \Gamma, \tag{1}$$

where $\beta = \beta(x)$ and $f(x)$ can be discontinuous across interface points $x_I \in \Gamma$. The jump conditions across the interface are given as

$$\begin{aligned} [u] = u^+ - u^- &= a(x) \quad \text{for } x \in \Gamma, \\ [\beta u_x] = \beta^+ u_x^+ - \beta^- u_x^- &= b(x) \quad \text{for } x \in \Gamma. \end{aligned} \tag{2}$$

We focus here on the case of two subdomains and one interface point, as it is straightforward to extend the methods for cases with more subdomains and interface points. Here $u^\pm = \lim_{x \to x_I^\pm} u(x)$ and $\beta^\pm = \lim_{x \to x_I^\pm} \beta(x)$.

As an alternative formulation of the problem, one could incorporate the jump conditions (2) into the differential equation itself by adding singular sources to the right-hand side of the equation. In such a formulation, the differential equation would take the form $(\beta u_x)_x = f(x) + b_I \delta(x - x_I) + a_I \bar{\beta} \delta'(x - x_I)$, where $\bar{\beta} = (\beta^+ + \beta^-)/2$ and $a_I = a(x_I)$ and $b_I = b(x_I)$, and where this differential equation is valid over the entire domain Ω. On the other hand, the differential equation in (1) is

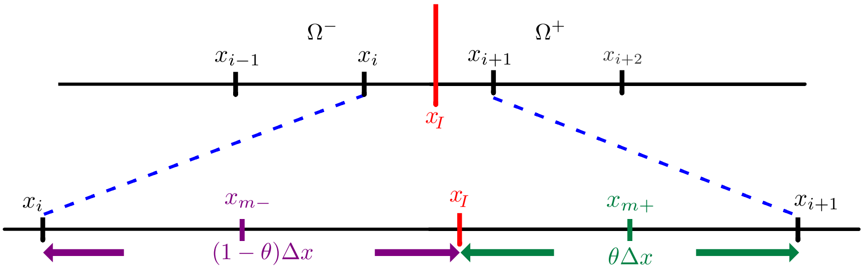


Figure 2. Cartesian grid points and an interfacial point in between.

valid only within each of the separate regions Ω^+ and Ω^-, and the jump conditions in (2) are needed to connect the solutions in Ω^+ and Ω^- and complete the problem specification. It will be convenient here to use the separate formulation in (1)–(2) throughout the paper.

2.1.1. *Finite differences.* A second-order finite-difference method can be derived on a Cartesian grid, with a symmetric operator, in the following way.

First, if the interface $\Gamma = \{x_I\}$ does not intersect with the grid edges connecting the three points x_{i-1}, x_i, and x_{i+1}, then we call x_i a standard Cartesian point. For all the standard Cartesian points we follow the standard second-order discretization for (1):

$$\frac{\beta_{i+1/2}((u_{i+1}-u_i)/\Delta x)-\beta_{i-1/2}((u_i-u_{i-1})/\Delta x)}{\Delta x}=f_i+O(\Delta x^2). \tag{3}$$

Next, consider nonstandard Cartesian points, such as x_i and x_{i+1} with an interfacial point $x_I \in \Gamma$ in between and with $x_i \in \Omega^-$ and $x_{i+1} \in \Omega^+$, as shown in Figure 2. Since the number of nonstandard points is assumed to be small, it should be possible to have an overall second-order accurate method that locally uses a first-order discretization at nonstandard points. Therefore, we use a first-order discretization of $(\beta u_x)_x$,

$$(\beta u_x)_x(x_i)=\frac{\beta(x_{m-})u_x(x_{m-})-\beta(x_{i-1/2})u_x(x_{i-1/2})}{x_{m-}-x_{i-1/2}}+O(\Delta x), \tag{4}$$

followed by second-order discretizations of the u_x terms, which lead to

$$\frac{\beta_{m-}(u_{I-}-u_i)/((1-\theta)\Delta x)-\beta_{i-1/2}(u_i-u_{i-1})/(\Delta x)}{((2-\theta)/2)\Delta x}=f_i+O(\Delta x), \tag{5}$$

where $\theta=(x_{i+1}-x_I)/\Delta x$. Note that the midpoints $x_{m-}=(x_i+x_I)/2$ and $x_{m+}=(x_I+x_{i+1})/2$, illustrated in Figure 2, are useful here to allow second-order discretizations of u_x. Here $\beta_{m-}=\beta(x_{m-})$ and $\beta_{m+}=\beta(x_{m+})$.

The final step is to replace in (5) the appearance of the interface value u_{I-} with Cartesian values and adjustments consisting of known quantities. To do this, we obtain additional equations by discretizing (1) at x_{I-} and x_{I+}, the left and right limits of x_I, using a method similar to the one above:

$$\frac{\beta_{m+}(u_{i+1}-u_{I+})/(\theta\Delta x)-\beta_{I+}u_x(x_{I+})}{\theta\Delta x/2}=f_{I+}+O(\Delta x)\quad \text{at } x_{I+}, \tag{6}$$

$$\frac{\beta_{I-}u_x(x_{I-})-\beta_{m-}(u_{I-}-u_i)/((1-\theta)\Delta x)}{(1-\theta)\Delta x/2}=f_{I-}+O(\Delta x)\quad \text{at } x_{I-}, \tag{7}$$

where $\beta_{I\pm}=\beta(x_{I\pm})$. The non-Cartesian unknowns $u_x(x_{I\pm})$ and $u_{I\pm}$ above can now be replaced by Cartesian unknowns by the following two steps. First, the weighted sum $(\theta\Delta x/2)\cdot(6)+((1-\theta)\Delta x/2)\cdot(7)$ is a combination that produces the jump $[\beta u_x]$:

$$\begin{aligned}\beta_{m+}\left(\frac{u_{i+1}-u_{I+}}{\theta\Delta x}\right)-\beta_{m-}\left(\frac{u_{I-}-u_i}{(1-\theta)\Delta x}\right)-[\beta u_x]&\\ =(\theta\cdot f_{I+}+(1-\theta)\cdot f_{I-})\frac{\Delta x}{2}+O(\Delta x^2).&\end{aligned} \tag{8}$$

Second, by using the jump conditions (2), we see that (8) can be rewritten as our desired formula for replacing u_{I-} by Cartesian u values:

$$\begin{aligned}u_{I-}=\frac{\hat\beta(1-\theta)}{\beta_{m-}}u_{i+1}+\frac{\hat\beta\theta}{\beta_{m+}}u_i&\\ -\frac{\hat\beta\theta(1-\theta)\Delta x^2}{\beta_{m+}\beta_{m-}}\left(\frac{\beta_{m+}a_I}{\theta\Delta x^2}+\frac{b_I}{\Delta x}+\tfrac{1}{2}(\theta\cdot f_{I+}+(1-\theta)\cdot f_{I-})\right),&\end{aligned} \tag{9}$$

where

$$\hat\beta=\frac{\beta_{m+}\beta_{m-}}{(1-\theta)\cdot\beta_{m+}+\theta\cdot\beta_{m-}}. \tag{10}$$

Lastly, substituting (9) into (5) yields a first-order discretization of the differential equation at x_i, in terms of only Cartesian values of u:

$$\begin{aligned}\frac{1}{\Delta x^2}(\beta_{i-1/2}\cdot u_{i-1}-(\beta_{i-1/2}+\hat\beta)u_i+\hat\beta\cdot u_{i+1})=f_i\cdot\left(\frac{2-\theta}{2}\right)&\\ +\frac{\hat\beta\theta}{\beta_{m+}}\left(\frac{\beta_{m+}}{\theta}\frac{a_I}{\Delta x^2}+\frac{b_I}{\Delta x}+\tfrac{1}{2}(\theta\cdot f_{I+}+(1-\theta)\cdot f_{I-})\right)+O(\Delta x).&\end{aligned} \tag{11}$$

For the neighboring nonstandard point at x_{i+1}, one can derive a similar finite difference formula:

$$\begin{aligned}\frac{1}{\Delta x^2}(\hat\beta\cdot u_i-(\hat\beta+\beta_{i+3/2})u_{i+1}+\beta_{i+3/2}\cdot u_{i+2})=f_{i+1}\cdot\left(\frac{1+\theta}{2}\right)&\\ +\frac{\hat\beta(1-\theta)}{\beta_{m-}}\left(-\frac{\beta_{m-}}{(1-\theta)}\frac{a_I}{\Delta x^2}+\frac{b_I}{\Delta x}+\tfrac{1}{2}(\theta\cdot f_{I+}+(1-\theta)\cdot f_{I-})\right)+O(\Delta x).&\end{aligned} \tag{12}$$

Comparing (11) and (12), it is clear that the difference operator acting on u is symmetric. The linear system can be solved using many standard efficient methods.

Note that this method in (11)–(12) looks similar to the GFM, which is first-order accurate [23; 24], but (11)–(12) include important differences that render this method second-order accurate. For instance, the right-hand-side terms in (11)–(12) have coefficients that are different from the GFM and that arise here as part of a systematic finite-differences derivation. Also, the values of β at the midpoints x_{m-} and x_{m+} were needed for the present method, whereas β values at the interface and Cartesian grid points and Cartesian midpoints are utilized in the GFM [23; 24].

In comparison to the IIM [19], notice that the present method has a symmetric operator, whereas the IIM operator is nonsymmetric. Also, the derivation of the IIM requires taking derivatives of jump conditions, whereas the present method is derived by simply applying finite difference formulas to the differential equation. It would be interesting to try to make a more firm connection between the present method and the IIM, which might also help tie together the GFM and IIM; however, we have not found any simple and clear connection beyond the comparisons described above.

To summarize, the basic idea in deriving (11)–(12) was to (i) start with midpoint-based finite differences using both Cartesian points and interface points, and then (ii) use the jump conditions to eliminate the interface values $u_{I\pm}$ from the system.

2.1.2. *Proof of second-order convergence.*

Theorem 1. *The numerical solution in Section 2.1.1 converges to the exact solution in the L^2 norm with second-order accuracy*: $\|\boldsymbol{U} - \boldsymbol{U}_{\text{ex}}\|_2 = O(\Delta x^2)$.

Proof. The setup of the proof is as follows. The numerical method in (3), (11), and (12) can be written in matrix-vector form as $A\boldsymbol{U} = \boldsymbol{F}$, and the exact solution satisfies $A\boldsymbol{U}_{\text{ex}} = \boldsymbol{F} + \boldsymbol{\tau}$, where $\boldsymbol{\tau}$ is the local truncation error. The error $\boldsymbol{e} = \boldsymbol{U} - \boldsymbol{U}_{\text{ex}}$ then satisfies $A\boldsymbol{e} = -\boldsymbol{\tau}$, and solving for $\boldsymbol{e}$ gives $\boldsymbol{e} = -A^{-1}\boldsymbol{\tau}$. The L^2 norm of the error then satisfies

$$\|\boldsymbol{e}\|_2 = \|A^{-1}\boldsymbol{\tau}\|_2 \leq \|A^{-1}\|_2 \|\boldsymbol{\tau}\|_2, \tag{13}$$

where the remaining task is to analyze $\|A^{-1}\|_2$ and $\|\boldsymbol{\tau}\|_2$ for small Δx.

Consistency was established in Section 2.1.1. Specifically, the local truncation error can be written as

$$\boldsymbol{\tau} = \boldsymbol{\tau}_s + \boldsymbol{\tau}_{\text{ns}} \quad \text{with } \|\boldsymbol{\tau}_s\|_2 = O(\Delta x^2) \text{ and } \|\boldsymbol{\tau}_{\text{ns}}\|_2 = O(\Delta x^2), \tag{14}$$

where we have split $\boldsymbol{\tau}$ so that the elements of $\boldsymbol{\tau}_s$ are nonzero only at standard points and the elements of $\boldsymbol{\tau}_{\text{ns}}$ are nonzero only at nonstandard points. The $O(\Delta x^2)$ scaling in (14) is then true because each element of $\boldsymbol{\tau}_s$ is $O(\Delta x^2)$, based on the finite difference formulas at the standard points; and each element of $\boldsymbol{\tau}_{\text{ns}}$ is $O(\Delta x)$, but the fraction of nonstandard points is $O(\Delta x)$, so $\|\boldsymbol{\tau}_{\text{ns}}\|_2 = O(\Delta x^2)$.

Stability is established by the bound

$$\|A^{-1}\|_2 \leq \frac{|\Omega|^2}{\beta_m}, \tag{15}$$

where $|\Omega|$ is the total length of the domain and $\beta_m = \min_{x \in \Omega} \beta(x)$ is a constant that is independent of Δx, and it is assumed that $\beta(x) > 0$ for all x. The proof of this bound is well-known [25] and is based on summation by parts and discrete Poincaré–Friedrichs inequality.

The proof of the theorem is completed by combining the consistency and stability results in (14) and (15) to show that (13) is $O(\Delta x^2)$. □

Note that, in the L^∞ norm, the local truncation error can only be bounded as $\|\boldsymbol{\tau}_{\mathrm{ns}}\|_\infty = O(\Delta x)$, which would complicate the present proof technique for second-order convergence if attempted with the L^∞ norm instead of the L^2 norm. Also, note that we have no such proof in two- or three-dimensional space, although proofs for 2D and 3D have been presented for similar methods [1], and numerical examples below demonstrate second-order convergence, in both the L^2 and L^∞ norms.

2.2. ***Two dimensions.*** Now consider the two-dimensional Poisson equation

$$(\beta u_x)_x + (\beta u_y)_y = f(x, y) \quad \text{for } \Omega \setminus \Gamma, \tag{16}$$

where $\Omega = \Omega^+ \cup \Omega^- \cup \Gamma$ and Γ is the interface between the sets Ω^+ and Ω^-. With $\boldsymbol{n} = (n^1(x, y), n^2(x, y))$ as the unit normal along Γ, the interface jump conditions are given as

$$\begin{aligned} [u] = u^+ - u^- \qquad &= a(\boldsymbol{x}) \quad \text{for } \boldsymbol{x} \in \Gamma, \\ [\beta u_n] = \beta^+ u_n^+ - \beta^- u_n^- &= b(\boldsymbol{x}) \quad \text{for } \boldsymbol{x} \in \Gamma, \end{aligned} \tag{17}$$

where $u_n = \boldsymbol{n} \cdot \nabla u$ is the derivative of u in the direction of the normal vector.

2.2.1. *Finite differences.* The goal of this section is to extend the ideas of the 1D case of Section 2.1 to the 2D case of (16)–(17) and arrive at a second-order finite-difference method. Similar to the 1D case, we call a Cartesian point (x_i, y_j) a standard point if this point and its nearest neighbors all lie within Ω^+ or all lie within Ω^-. For standard points, (16) is discretized with the standard, second-order, five-point finite-difference formula. For nonstandard points, on the other hand, the interface must be taken into account.

For nonstandard points, such as point (x_i, y_j) illustrated in Figure 3, we obtain a first-order discretization by using similar ideas as in the 1D case. Following a derivation similar to (5)–(11), by essentially just replacing f by $f - (\beta u_y)_y$, we

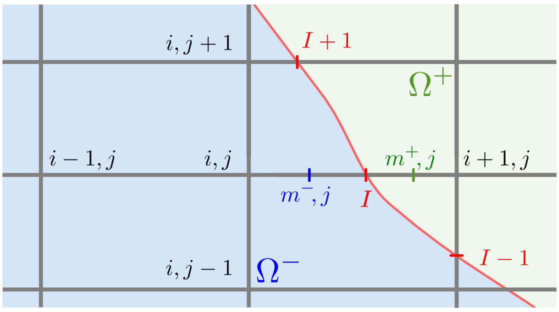


Figure 3. Nonstandard grid point at (x_i, y_j).

arrive at

$$\frac{1}{\Delta x^2}(\beta_{i-1/2,j} \cdot u_{i-1,j} - (\beta_{i-1/2,j} + \hat{\beta})u_{i,j} + \hat{\beta} \cdot u_{i+1,j})$$
$$+ \frac{1}{\Delta y^2}(\beta_{i,j-1/2} \cdot u_{i,j-1} - (\beta_{i,j-1/2} + \beta_{i,j+1/2})u_{i,j} + \beta_{i,j+1/2} \cdot u_{i,j+1})$$
$$= f_{i,j} \cdot \left(\frac{2-\theta}{2}\right) + (\beta u_y)_y(x_i, y_j) \cdot \frac{\theta}{2} + F^x_{\text{cor}} + O(\Delta x), \quad (18)$$

where

$$\hat{\beta} = \frac{\beta(x_{m+}, y_j) \cdot \beta(x_{m-}, y_j)}{(1-\theta) \cdot \beta(x_{m+}, y_j) + \theta \cdot \beta(x_{m-}, y_j)}, \quad (19)$$

and

$$F^x_{\text{cor}} = \frac{\hat{\beta}\theta}{\beta(x_{m+}, y_j)} \left\{ \frac{\beta(x_{m+}, y_j) a(x_I, y_j)}{\theta \Delta x^2} + \frac{[\beta u_x]}{\Delta x} \right.$$
$$\left. + \tfrac{1}{2}(\theta \cdot (f - (\beta u_y)_y)(x_{I+}, y_j) + (1-\theta) \cdot (f - (\beta u_y)_y)(x_{I-}, y_j)) \right\}. \quad (20)$$

This finite-difference formula has a left-hand side with the desirable property of a symmetric operator, as in the 1D case. However, the right-hand side of (18) now depends on the solution u itself, so an iterative method will be described below for finding a solution.

Also, a more general case would allow for other interface crossings, such as a crossing at point (x_i, y_J), with $y_j < y_J < y_{j+1}$, which would generate some slight modifications to the derivation and finite-difference formula. Since the more general case is only slightly different from (18), it is relegated to Appendix A.

To estimate the derivatives on the right-hand side of (18), simple finite differences are used. For the term $(\beta u_y)_y(x_i, y_j)$, standard centered differences can be used with the points (x_i, y_{j-1}), (x_i, y_j), and (x_i, y_{j+1}). For the term $(\beta u_y)_y(x_{I-}, y_j)$ at the interface, from (20), one can approximate it with the nearby Cartesian value

$(\beta u_y)_y(x_i, y_j)$ with an acceptable error of $O(\Delta x)$, and then one can use a standard centered discretization with the points (x_i, y_{j-1}), (x_i, y_j), and (x_i, y_{j+1}). The term $(\beta u_y)_y(x_{I+}, y_j)$ can be handled similarly by using $(\beta u_y)_y$ at the nearby Cartesian point (x_{i+1}, y_j). Lastly, the jump $[\beta u_x]$ from (20) can be written in terms of normal and tangential jumps as

$$\begin{aligned}[\beta u_x] &= [\beta u_n]n^1 - [\beta u_\tau]n^2 \\ &= b_I n^1 - [\beta u_\tau]n^2. \end{aligned} \tag{21}$$

The term $[\beta u_\tau]$ can then be estimated using finite differences with u values from the interface points labeled $I-1$, I, and $I+1$ in Figure 3 (or possibly using another triplet, say $I-2$, I, and $I+1$, if the two interface points $I-1$ and I are located too close together, such as within $O(h^2)$ distance; see Appendix B for details). Note that a second-order finite-difference formula is needed for $[\beta u_\tau]$ in order for the term $[\beta u_x]/\Delta x$ to have an error of $O(\Delta x)$. To determine the u values at the interface points, one can use the formula

$$\begin{aligned} u(x_{I-}, y_j) = {} & \frac{(1-\theta)\hat{\beta}}{\beta(x_{m-}, y_j)} u_{i+1,j} + \frac{\theta\hat{\beta}}{\beta(x_{m+}, y_j)} u_{i,j} - \frac{\hat{\beta}(1-\theta)\theta\Delta x^2}{\beta(x_{m+}, y_j)\beta(x_{m-}, y_j)} \\ & \cdot \left(\frac{\beta(x_{m+}, y_j)a_I}{\theta\Delta x^2} + \frac{[\beta u_x]}{\Delta x} + \frac{\theta}{2} \cdot ((\beta u_x)_x)_{i+1,j} + \frac{(1-\theta)}{2} \cdot ((\beta u_x)_x)_{i,j} \right), \end{aligned} \tag{22}$$

and $u(x_{I_+}, y_j) = u(x_{I_-}, y_j) + a(x_I, y_j)$ by the jump condition (17). This formula arises as part of the derivation of (18) and is similar to the 1D case, and formulas for $u(x_i, y_{J_\pm})$ can be obtained similarly if the crossing is in the y-direction. Note that this formula in 2D does not actually provide the desired result of the interface u value in terms of the Cartesian u values, since the right-hand side depends on interface u values via the $[\beta u_x]$ term. Nevertheless, this formula can be used as part of an iterative procedure to complete the specification of the numerical methods.

2.2.2. *Iterative methods.* In this section, a simple iterative method is proposed here for solving the linear system from Section 2.2.1.

Before describing the standard iterative method of the present paper, consider first a type of Picard iteration:

$$A\boldsymbol{u}^{[k+1]} = \boldsymbol{F}^{[k]}. \tag{23}$$

This is an iterative version of the matrix-vector form of the finite difference method, one row of which is described in (18): A is the symmetric matrix from the left-hand side, $\boldsymbol{u}^{[k+1]}$ is the vector of all Cartesian u values (from iteration $k+1$), and $\boldsymbol{F}^{[k]}$ is the vector from the right-hand-side terms. The basic idea is to iteratively update $\boldsymbol{F}^{[k]}$ on the right-hand side as new, more accurate information about $\boldsymbol{u}^{[k]}$ is obtained. As an initial condition, $\boldsymbol{F}^{[0]}$ is defined as the right-hand side of (18) with all instances of u

ignored (i.e., with $[\beta u_\tau]$, $(\beta u_x)_x$, and $(\beta u_y)_y$ all set to zero, essentially equivalent to setting $u^{[0]} = 0$ as an initial guess), and the first solution $\boldsymbol{u}^{[1]}$ is found by solving $A\boldsymbol{u}^{[1]} = \boldsymbol{F}^{[0]}$. As a result, *the solution* $\boldsymbol{u}^{[1]}$ *at the first iteration is essentially the same as the GFM solution* [23; 24] *and is therefore a first-order accurate solution.* It can be used to estimate the interface u values, which we assemble abstractly into a vector $\boldsymbol{u}_I^{[k]}$ and update iteratively as $\boldsymbol{u}_I^{[k+1]} = B\boldsymbol{u}_I^{[k]} + C\boldsymbol{u}^{[k+1]} + \boldsymbol{G}$, one row of which is described by (22): the $B\boldsymbol{u}_I^{[k]}$ corresponds to the $[\beta u_\tau]$ term, the $C\boldsymbol{u}^{[k+1]}$ corresponds to all terms with Cartesian u values, and the $\boldsymbol{G}$ corresponds to the jump terms involving a_I and b_I. An initial interface value of $\boldsymbol{u}_I^{[0]} = \boldsymbol{\Gamma}$ is used, consistent with the idea of ignoring all instances of u in the initial condition $\boldsymbol{F}^{[0]}$. The second iteration then proceeds by defining $\boldsymbol{F}^{[1]}$ based on the right-hand side of (18) and now using $\boldsymbol{u}^{[1]}$ and $\boldsymbol{u}_I^{[1]}$ to provide a more accurate estimate of the true $\boldsymbol{F}$ value. The solution $\boldsymbol{u}^{[2]}$ at the second iteration is then found from solving the symmetric system $A\boldsymbol{u}^{[2]} = \boldsymbol{F}^{[1]}$. This procedure can be repeated to iteratively estimate the solution of the finite-difference method.

For the stopping criterion for the iterative procedure, the differences $u_d^{[k]} = \|\boldsymbol{u}^{[k+1]} - \boldsymbol{u}^{[k]}\|_\infty$ and $F_d^{[k]} = \|\boldsymbol{F}^{[k+1]} - \boldsymbol{F}^{[k]}\|_\infty$ are monitored. When k is large enough so that $u_d^{[k]} < C_u h^2$, where $h = \Delta x = \Delta y$, one can presumably stop iterating since the iterations are producing only small corrections that are within the desired $O(h^2)$ accuracy of the numerical solution. As our standard stopping criterion, in addition to $u_d^{[k]} < C_u h^2$ we also require $F_d^{[k]} < C_F h$ in order to ensure that the estimated right-hand-side terms are not significantly changing at any location. The constants C_u and C_F will be set equal to 1 here for simplicity, but in the future it would be interesting to tailor the choices of C_u and C_F to the particular problem under consideration; for instance, they could be chosen based on the expected error, which could be estimated based on, e.g., expected local truncation error and/or smoothness of the solution. Also note that, while this standard stopping criterion was chosen with solution accuracy as the main consideration, one could also imagine other stopping criteria that consider computational efficiency or other factors; some other stopping criteria are explored in Section 4.

As the standard iterative method used here, a modification of Picard iteration is actually used. While Picard iteration does work well in many cases, we found that it diverges in some cases. Nevertheless, by making some slight modifications, a robust method can be designed. Our standard iterative method here uses a simple relaxation procedure to extend Picard iteration; it is described in Appendix C, and it is shown below to provide robust results.

2.3. *Three dimensions.*

The three-dimensional Poisson equation is

$$(\beta u_x)_x + (\beta u_y)_y + (\beta u_z)_z = f(x, y, z), \quad \text{for } \Omega \setminus \Gamma, \tag{24}$$

where $\Omega = \Omega^+ \cup \Omega^- \cup \Gamma$ and Γ is a surface that marks the interface between the sets Ω^+ and Ω^-. The interface jump conditions are given as in the 2D case in (17).

The 3D discretization is essentially the same as in the 2D case in Section 2.2. We note one difference that arises: in 3D, the jump $[\beta u_x]$ from (21) takes the form

$$\begin{aligned}[\beta u_x] &= [\beta u_n]c^0 + [\beta u_{\tau_1}]c^1 + [\beta u_{\tau_2}]c^2 \\ &= b_I c^0 + [\beta u_{\tau_1}]c^1 + [\beta u_{\tau_2}]c^2,\end{aligned} \tag{25}$$

where $\hat{\boldsymbol{x}} = c^0\hat{\boldsymbol{n}} + c^1\hat{\boldsymbol{\tau}}_1 + c^2\hat{\boldsymbol{\tau}}_2$ was used to write the unit coordinate vector $\hat{\boldsymbol{x}}$ in terms of the interface normal vector $\hat{\boldsymbol{n}}$ and two unit vectors $\hat{\boldsymbol{\tau}}_1$ and $\hat{\boldsymbol{\tau}}_2$ from the 2D tangent plane of the interface. Here, in 3D, note that tangential derivatives are needed in two independent directions in the 2D tangent plane. The two directions can be conveniently chosen by using the Cartesian coordinate planes. For example, if $(x_I, y_j, z_k) \in \Gamma$, where x_I is not a Cartesian grid point, then the intersection of surface Γ and the plane $z = z_k$ can be used to define one direction in the 2D tangent plane, and the intersection of surface Γ and the plane $y = y_j$ can be used to define the other direction. In this way, computation of the tangential derivatives in 3D can be reduced to essentially the same form as in 2D.

3. Examples

In this section, second-order convergence is demonstrated through numerical examples. In all examples, the same grid spacing is used in each coordinate direction ($\Delta x = \Delta y = \Delta z$), and the number of grid points in each coordinate direction is N, so the total number of grid points is N, N^2, or N^3 for the 1D, 2D, or 3D cases, respectively.

3.1. *One dimension.*

Example 1D-1. Consider a domain $\Omega = [0, 1]$ separated into subdomains $\Omega- = [0, x_I)$ and $\Omega^+ = (x_I, 1]$, where $x_I = 2 - \sqrt{2}$. The solution to the one-dimensional equation $\beta u_{xx} = f$ is $u^- = \exp(-x) - 0.3646x + 0.4$ and $u^+ = \exp(-x)/2 + x^2/2 + 0.5005x$ where $\beta = 100$ in Ω^- and $\beta = 200$ in Ω^+, with $f = 100\exp(-x)$ in Ω^- and $f = 100\exp(-x) + 200$ in Ω^+. The jump conditions connecting the two equations at x_I are $a(x_I) = u^+ - u^- = 0$ and $b(x_I) = 100(2u_x^+ - u_x^-) = 253.72$.

The exact solution and error analysis of this example can be found in Figure 4.

3.2. ***Two dimensions.*** The following 2D and 3D examples are tested on some rectangular domain Ω where Ω will be divided into Ω^+ and Ω^- by an interface Γ. It will sometimes be convenient to describe the interface Γ in terms of a level-set function $\phi(x)$ as $\Gamma = \{\boldsymbol{x} \in \Omega : \phi(\boldsymbol{x}) = 0\}$, where the two sets Ω^+ and Ω^- can be described as $\Omega^+ = \{\boldsymbol{x} \in \Omega : \phi(\boldsymbol{x}) > 0\}$ and $\Omega^- = \{\boldsymbol{x} \in \Omega : \phi(\boldsymbol{x}) < 0\}$. The coefficients

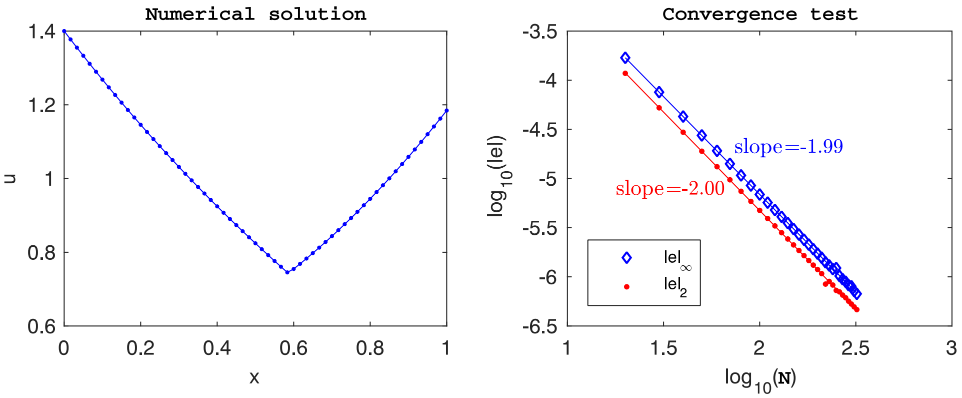


Figure 4. Example 1D-1. Left: numerical solution with number of grid points $N = 61$. Right: error $\|e\|$ as a function of number of grid points N, as a log-log plot including slope of its linear fit.

β are assumed to be smooth in both Ω^+ and Ω^-, but may have a jump across the interface ϕ. The piecewise smooth β in Ω^+ and Ω^- will be denoted by β^+ and β^-, respectively. As a consequence, the solution u may be discontinuous across ϕ, but is $\mathscr{C}^2$ in both Ω^+ and Ω^-, and will similarly be denoted by u^+ and u^-, respectively. The examples in this section will provide tests of the numerical method for several factors that could influence the numerical method's convergence, such as geometry of the interface, spatial variations in the coefficients $\beta^\pm(x, y)$, and contrast β^+/β^- due to jumps in the coefficients.

Example 2D-1 (constant coefficient). In this example, we take β be a piecewise constant function with $\beta^- = 2$ and $\beta^+ = 1$, and the interface is a circle described by the level set function $\phi(x, y) = (x - 0.5)^2 + (y - 0.5)^2 - 0.25^2$. The solution is $u^- = \exp(-x^2 - y^2)$, $u^+ = 0$, with $f^- = 8(x^2 + y^2 - 1)\exp(-x^2 - y^2)$, $f^- = 0$, on the domain $\Omega = [0, 1] \times [0, 1]$. Second-order convergence can be seen in Figure 5, right.

Example 2D-2 (variable coefficient). The next example we take β to be a piecewise smooth function with $\beta^- = x^2 + y^2 + 1$, and $\beta^+ = 1$ with the same domain and level set function as the previous example. The solution is $u^- = \exp(x^2 + y^2)$, and $u^+ = \exp(-x^2 - y^2)$ and source term is $f^- = 4(\beta^-(x^2 + y^2 + 1) + (x^2 + y^2))\exp(x^2 + y^2)$, $f^+ = 4(x^2 + y^2 - 1)\exp(-x^2 - y^2)$. Error analysis is presented in Figure 6, right.

Example 2D-3 (variable coefficient). With the same solution u in Example 2D-2, this example is computed on a domain $\Omega = [-1, 1] \times [-1, 1]$, with $\beta^- = x^2 + y^2 + 1$, $\beta^+ = \sqrt{x^2 + y^2 + 2}$, and $f^- = 4(\beta^-(x^2 + y^2 + 1) + (x^2 + y^2))\exp(x^2 + y^2)$, $f^+ = (4\beta^+(x^2 + y^2 - 1) - 2(x^2 + y^2)/\sqrt{x^2 + y^2 + 2})\exp(-x^2 + y^2)$. The interface

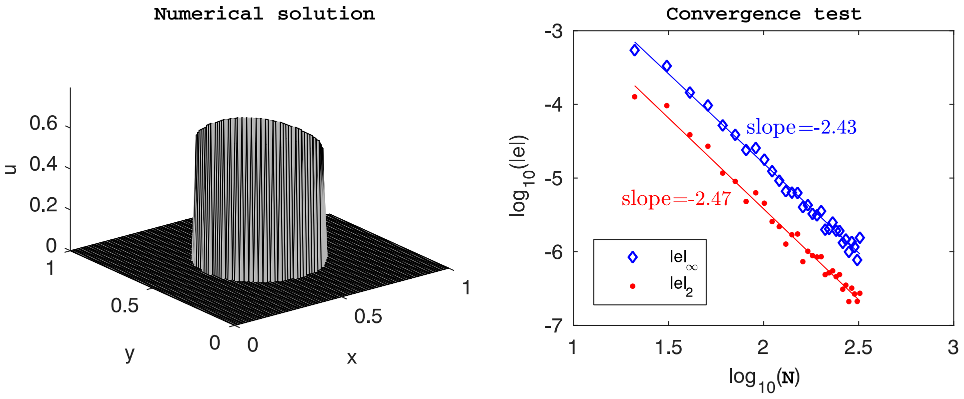


Figure 5. Example 2D-1: constant coefficient. Left: numerical solution, $N = 81$. Right: error $\|e\|$ as a function of number of grid points in each coordinate direction, N, as a log-log plot including slope of its linear fit.

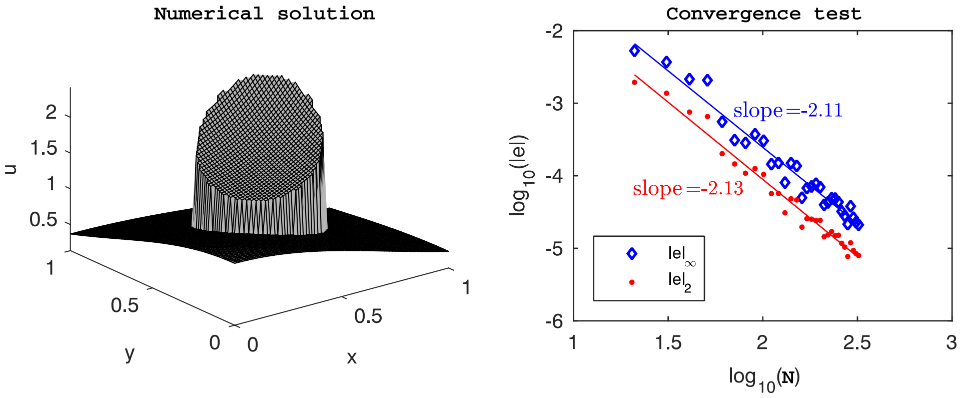


Figure 6. Example 2D-2: variable coefficient. Left: numerical solution, $N = 81$. Right: error $\|e\|$ as a function of number of grid points in each coordinate direction, N, as a log-log plot including slope of its linear fit.

is parametrized by

$$\begin{cases} x(t) = 0.02\sqrt{5} + (0.5 + 0.2\sin(5t))\cos t, \\ y(t) = 0.02\sqrt{5} + (0.5 + 0.2\sin(5t))\sin t, \end{cases} \tag{26}$$

with $t \in [0, 2\pi]$. Second-order convergence is demonstrated in Figure 7, right.

For the spatial variations of the error, we describe the two cases of Figures 6 and 7. In these cases the error appears to typically take its maximum value near the interface. In the special case of the outlier of $N = 251$ from Figure 7, right, the error furthermore takes its maximum value at a single localized spike near one point close to the interface. These features are possibly related to the curvature of the interface or other factors.

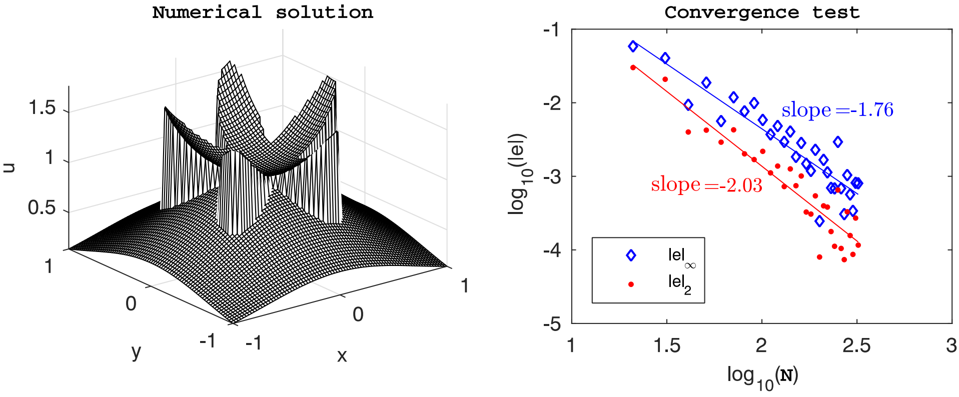


Figure 7. Example 2D-3: variable coefficient. Left: numerical solution, $N = 81$. Right: error $\|e\|$ as a function of number of grid points in each coordinate direction, N, as a log-log plot including slope of its linear fit.

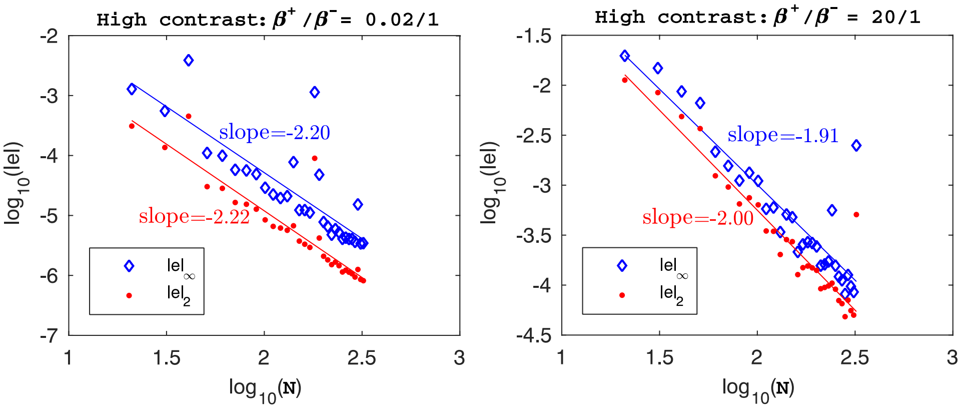


Figure 8. Error plots for high-contrast case, Example 2D-4. Left: $\beta^+/\beta^- = 0.02/1$. Right: $\beta^+/\beta^- = 20/1$.

Example 2D-4 (high-contrast coefficient cases). A series of tests were conducted on the large coefficient ratios, either $\beta^+/\beta^- \ll 1$ or $1 \ll \beta^+/\beta^-$. Here we test with $u^- = \exp(x^2 + y^2)$, and $u^+ = \exp(-x^2 - y^2)$ with a circular interface as in Example 2D-1, and $(\beta^+, \beta^-) = (0.02, 1)$ and $(20, 1)$. Second-order convergence can still be obtained (see Figure 8).

3.3. *Three dimensions.*

Example 3D-1 (variable coefficient with spherical interface). Domain $\Omega = [0, 1] \times [0, 1] \times [0, 1]$ is divided into Ω^+ and Ω^- by a sphere centered at $(0.5, 0.5, 0.5)$ with radius 0.25. The variable coefficients β in (24) are $\beta^- = 10 + \sin(xy + z)$ and $\beta^+ = 10 + \cos(x + yz)$, with solution $u^- = \exp(x^2 + y^2 + z^2)$ and $u^+ = 0$

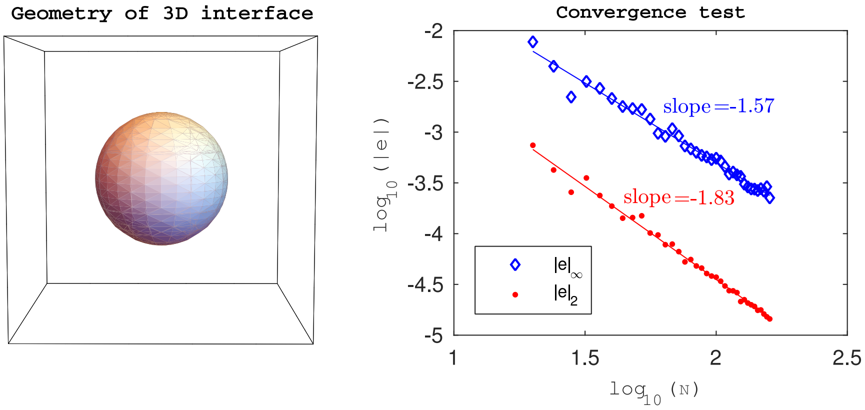


Figure 9. Example 3D-1: variable coefficient with spherical interface. Left: geometry of the interface. Right: error $\|e\|$ as a function of number of grid points in each coordinate direction, N, as a log-log plot including slope of its linear fit.

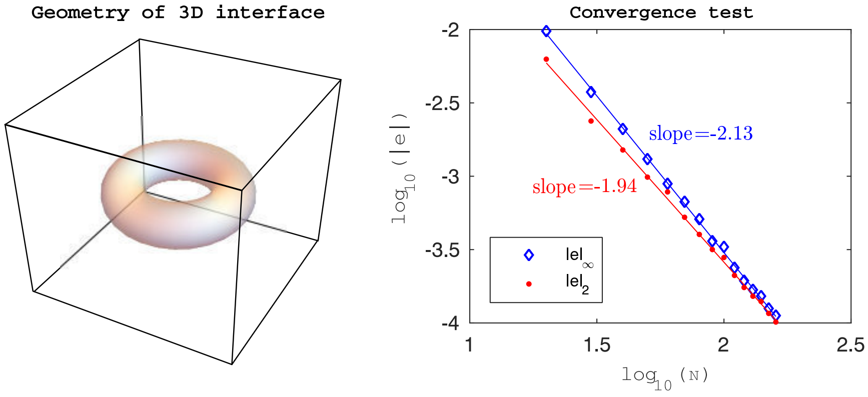


Figure 10. Example 3D-2: variable coefficient with torus interface. Left: numerical solution. Right: error $\|e\|$ as a function of number of grid points in each coordinate direction, N, as a log-log plot including slope of its linear fit.

and $f^- = (4\beta^-(x^2+y^2+z^2+3/2)+(4xy+2z)\cos(xy+z))\exp(x^2+y^2+z^2)$, $f^+ = 0$. See Figure 9 for the geometry of the spherical interface and second-order convergence in L^2.

Example 3D-2 (variable coefficient with torus interface). For the same β, u, and f in the previous example, we test this iterative method on $\Omega = [-1, 1] \times [-1, 1] \times [-1, 1]$ with a toroid interface described by the level set function $\phi(x, y, z) = (x^2+y^2+z^2+R^2-r^2)^2 - 4R^2(x^2+y^2)$, where $R = 0.501+\sqrt{2}/10$, $r = 0.251$. The geometry of the interface and second-order convergence in L^∞ and L^2 are in Figure 10.

4. Greater efficiency via alternative stopping criteria

4.1. ***Iteration counts for standard stopping criterion.*** In most of the cases shown above, the number of iterations required to reach the stopping criterion is small, which makes this iterative method efficient, as demonstrated in Figure 11. More specifically, approximately 10–20 iterations are used in 2D cases, and approximately 5–10 iterations in the 3D cases. For high-contrast cases (Example 2D-4), the number of iterations becomes larger (approximately 50–150, as seen in Figure 12), although it is smaller (approximately 20–40) for some N, and the bound on the number of iterations is essentially independent of the number of grid points.

These examples demonstrate that the present method may be practical and efficient for time-dependent problems where the elliptic solver is needed at every time step. Also, further reduction of the iteration counts may be possible for time-dependent problems. For instance, in these first explorations, we are using a simple yet crude initial guess for the iterative methods; i.e., we are effectively using $u^{[0]} = 0$ (see Section 2.2.2). For a better initial guess $u^{[0]}$, in a time-dependent problem, the solution from the previous time step could potentially be used, with the possibility of substantially reducing the number of iterations required.

Below we discuss other possibilities of further reducing the iterations counts through alternative stopping criteria — e.g., by using a small, fixed number of iterations in Section 4.2, and propose some other feasible stopping criteria in Section 4.3.

4.2. ***Greater efficiency via a small, fixed number of iterations.*** In most cases, the accuracy improves tremendously after only a few iterations; in other words, the latter iterations make only small modifications to the solution in order to satisfy the stopping criterion. Therefore, in practice, we may speed up this numerical method by using a fixed number of iterations without losing too much accuracy. Figure 13

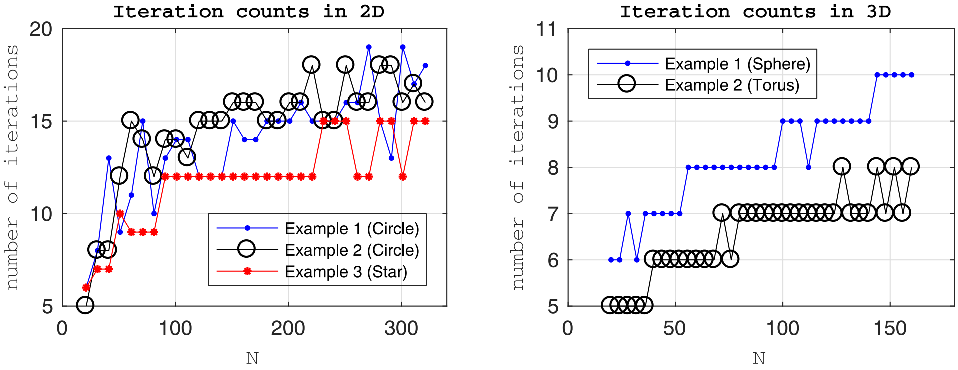


Figure 11. Number of iterations for (left) 2D examples and (right) 3D examples.

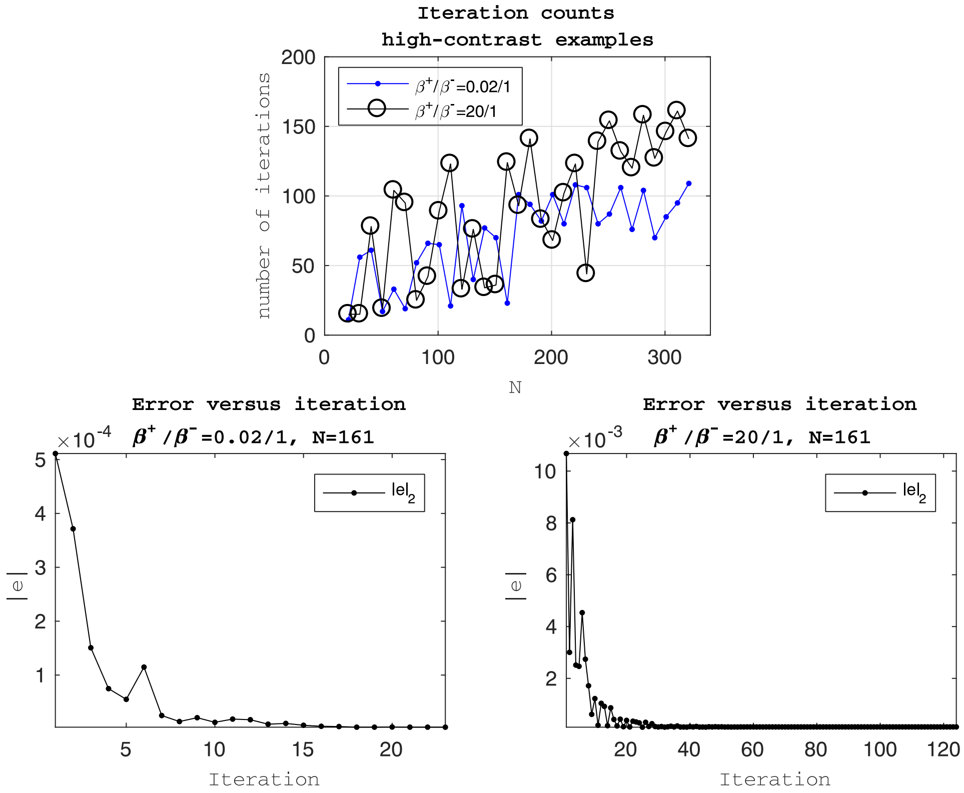


Figure 12. Errors and iterations for the high-contrast cases from Example 2D-4. Top: number of iterations as a function of the number of grid points in each coordinate direction, N. Bottom: L^2 error as a function of iterations, for $N = 161$.

shows results of both 2D and 3D examples with only a small number of iterations (five), which still show second-order accuracy.

4.3. ***Other stopping criteria.*** Several other stopping criteria were also tested, beyond the standard criterion from Section 2.2.2, by using different combinations of criteria for the smallness of the differences $u_d^{[k]} = \|\boldsymbol{u}^{[k+1]} - \boldsymbol{u}^{[k]}\|_\infty$ and/or $F_d^{[k]} = \|\boldsymbol{F}^{[k+1]} - \boldsymbol{F}^{[k]}\|_\infty$. A promising criterion may be to stop when $u_d^{[k]} < h^2$, without enforcing any smallness criterion on $F_d^{[k]}$; in some tests, this led to second-order accuracy with fewer iterations, although we have not yet tested this criterion on a wide array of cases.

Also, it would be interesting in the future to tailor the stopping criterion to the particular problem at hand. For instance, the criterion $u_d^{[k]} < h^2$ could be slightly generalized to $u_d^{[k]} < C_u h^2$, where C_u is a parameter that could be chosen based on, e.g., the expected accuracy or smoothness of the solution, given the parameters of the problem such as $\beta^{\pm}(x, y, z)$, $f(x, y, z)$, and the geometry of the interface.

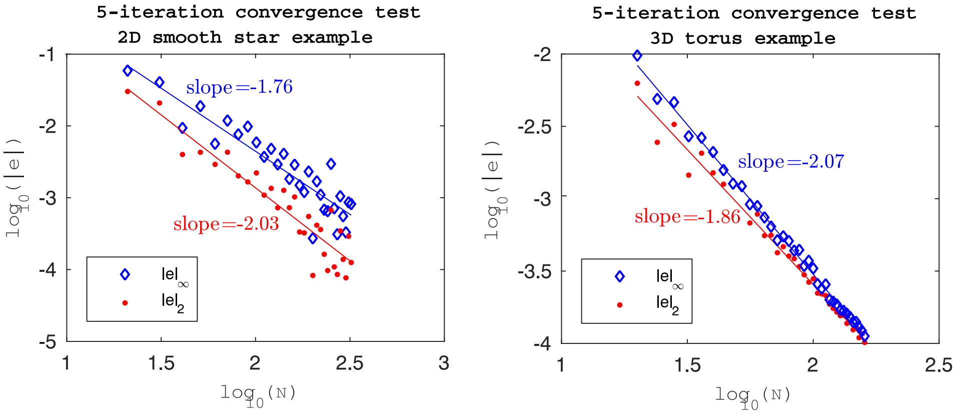


Figure 13. Error as a function of number of grid points in each coordinate direction, N, using a fixed number of iterations (five) for more efficient computations. Left: smooth star example in Section 3.2. Right: torus example in Section 3.3.

5. Comparisons with formulations of other methods

In this section we compare the present formulation with the formulations of other methods [36; 3; 32; 17; 26; 27], to add to the comparisons with the GFM [23; 24] and IIM [19; 20; 8; 21; 5] that were described above in Section 2.1.1. An IIM viewpoint of the present method is also described in the 1D case at the end of the section.

In [36], another approach had been taken to obtain a symmetric operator; the derivation used Taylor series expansions and derivatives of jump conditions, which can be somewhat complex compared to the simple derivations of the present paper that mainly involve centered finite difference formulas. Note that the present method and the method of [36] are, in fact, distinct. As one difference, in the 1D versions of the two methods, the method of [36] requires not only the symmetric operator but also some augmentation to account for jumps, whereas the method of the present paper has an operator in 1D that is symmetric on its own. Also, the method [36] utilizes a discretization of the standard Laplacian operator, whereas the present method maintains the symmetry of the elliptic operator that includes β.

In [3], an interesting approach was proposed which, like the present method, involves a symmetric operator and an iterative method to determine an adjusted forcing. The derivation is somewhat complex in that it is a version of the IIM and therefore uses Taylor series and derivatives of jump conditions. The derivation is presented in 2D, but no 3D results are presented. Also, their iterative procedure does not produce a first-order accurate solution at the first iteration, and therefore, it is likely to require a very large number of iterations (as possibly indicated by their very small relaxation parameter). The number of iterations, however, are not reported, and the iterative methods and stopping criterion are not described in detail. In contrast, in the present paper, the first iteration is essentially the GFM, and the simple finite-difference formulation allows for efficient setup and computation even in 3D.

In [17], following [32], another interesting approach is used to obtain a symmetric operator with corrections to the right-hand side. The method is implemented in 2D, but no 3D results are presented. Also, the method is presented for the standard Laplacian operator, not for the case of discontinuous and/or spatially varying coefficient $\beta(\boldsymbol{x})$.

Another interesting method called the correction-function method has been developed by building on the GFM and computing a corrected forcing function to achieve higher-order accuracy [26; 27]. In this method, the corrected forcing function is not derived explicitly; instead, the corrected forcing function is shown to satisfy a certain new PDE, and the new PDE is solved numerically to determine the corrected forcing function. The method has been demonstrated to achieve second-order and even fourth-order accuracy, although it has not yet been implemented for 3D problems and it has only been developed for cases with constant coefficients and piecewise-constant coefficients. It is similar to the method of the present paper in that both methods seek to compute corrections to the GFM; the present paper's method perhaps offers a simpler formulation (involving only one-dimensional finite differences) and simpler implementation for 3D problems.

Note that the present method appears to be well-behaved for any values of subcell location θ, even values of θ that are close to 0 or 1; in contrast, it has been noted in some applications of the GFM [9] that poor behavior could potentially result if θ is close to 0 or 1. One might expect poor behavior in these cases, since the finite difference formulas in, e.g., (6)–(7) have factors of $1/\theta$ and $1/(1-\theta)$; however, these formulas are part of the derivation only, not part of the present numerical algorithm. Here, no special treatment of the case $\theta \approx 0$ or $\theta \approx 1$ was used in the numerical results, and in examining the possible influence of θ on the numerical error, we found no systematic relationship. (For computing tangential derivatives βu_τ, which are not part of the GFM but are used in the present method in higher dimensions, some values of θ are given special treatment for computing u_τ; see Section 2.2.1.) A theoretical explanation can be seen from the finite difference formulas of the method; in particular, notice that (11) and (12) depend on θ in a smooth, nonsingular way. In fact, a more precise error analysis shows that the error term $O(\Delta x)$ in (11) is $O(\theta \Delta x)$ and in (12) is $O((1-\theta)\Delta x)$, which theoretically suggests that $\theta \approx 0$ or $\theta \approx 1$ should lead to smaller truncation error, and also indicates that the proof of the convergence theorem in Section 2.1.2 is valid for any value with $0 < \theta < 1$. Hence, the method should work fine with $\theta \approx 0$ or $\theta \approx 1$ except for possible values of precisely $\theta = 0$ or $\theta = 1$.

6. Conclusions

In this article, a simple numerical scheme is proposed to obtain second-order accuracy in solving the Poisson equation with sharp interfaces. One important contribution is a simple derivation that mainly involves centered finite difference formulas, with less reliance on the Taylor series expansions and derivatives of jump conditions used in typical immersed interface method derivations. The derivation here preserves the symmetry of the differential operator, and the method is formulated on a Cartesian grid. The accuracy of the method is proved rigorously in 1D and verified numerically in 2D and 3D. The three-dimensional problems are relatively easy to set up due to the method's simple derivation.

An iterative procedure was used for solving 2D or 3D problems, and the desired second-order accuracy can be obtained with only a small, fixed number of iterations (typically five), which makes this method efficient, even in 3D. In the future it would be interesting to investigate other algorithmic choices; for instance, perhaps an iterative method could be designed that requires an even smaller number (e.g., two or three) of iterations, or perhaps the method could be successful if the iterated correction terms were instead written as part of the left-hand-side linear operator, in which case the symmetry of the operator is lost but the nonsymmetric system could possibly be solved without the need for the outer iterations introduced in the present paper. Also, here we did not make a great effort to optimize the algorithms for cases with high-contrast coefficients, which require higher iteration counts, but such an effort would be interesting to pursue in the future.

The proposed method may be applied to solving time-dependent problems that require the solution of an elliptic PDE at each time step — for example, the heat equation with interfaces or multiphase flow problems [10; 34; 33]. In such applications, the present method could be used with any characterization of the interface (level set, Lagrangian markers, etc.), and the interface could have a location and shape that evolves in time. Also, for the iterative algorithms described here, some computational savings may be possible for time-dependent problems, since the solution from the previous time step could provide a good initial guess for the iterative method at the current time step.

Appendix A. 2D discretization with two interface crossings

In this appendix, it is shown how to formulate the finite difference method in a case that is more general than in Section 2.2.1.

Suppose the interface crosses the stencil of point (x_i, y_j) in two places, as shown in Figure 14. The crossing between (x_i, y_j) and (x_{i+1}, y_j) is as in Section 2.2.1, and now a new, second crossing is present between (x_i, y_j) and (x_i, y_{j+1}). Accordingly,

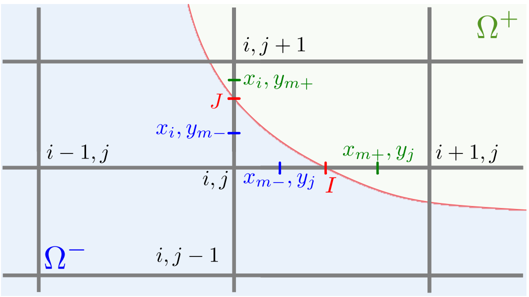


Figure 14. Nonstandard point (x_i, y_j) with interface crossing the stencil in both x and y directions.

define $\zeta = (y_{j+1} - y_J)/\Delta y$, where $(x_i, y_J) \in \Gamma$, and assume $(x_i, y_j) \in \Omega^-$ and $(x_i, y_{j+1}) \in \Omega^+$.

To obtain a finite difference method with a symmetric operator in this case, start by writing the 1D formula from (11) as

$$S^x u = (\beta u_x)_x \cdot (2-\theta)/2 + F^x_{\text{cor}} + O(\Delta x), \tag{27}$$

where S^x is the symmetric finite difference operator and F^x_{cor} is the correction term. A similar formula can be derived for a symmetric finite difference operator in the y direction:

$$S^y u = (\beta u_y)_y \cdot (2-\zeta)/2 + F^y_{\text{cor}} + O(\Delta y). \tag{28}$$

Summing up the two leads to

$$S^x u + S^y u = f - (\beta u_x)_x \cdot \theta/2 - (\beta u_y)_y \cdot \zeta/2 + F^x_{\text{cor}} + F^y_{\text{cor}} + O(\Delta x) + O(\Delta y), \tag{29}$$

which is the desired formula. Also note that the derivation in 3D follows the same simple principles by including the addition of a third component for $S^z u$.

Written out in detail, (29) takes the form

$$\begin{aligned}
&\frac{1}{(\Delta x)^2}\big(\beta(x_{i-1/2}, y_j)\cdot u_{i-1,j} - (\beta(x_{i-1/2}, y_j) + \hat{\beta})u_{i,j} + \hat{\beta}\cdot u_{i+1,j}\big) \\
&\quad + \frac{1}{(\Delta y)^2}\big(\beta(x_i, y_{j-1/2})\cdot u_{i,j-1} - (\beta(x_i, y_{j-1/2}) + \tilde{\beta})u_{i,j} + \tilde{\beta}\cdot u_{i,j+1}\big) \\
&\quad\; = f_{i,j} - (\beta u_x)_x(x_i, y_j)\cdot\frac{\theta}{2} - (\beta u_y)_y(x_i, y_j)\cdot\frac{\zeta}{2} + F^x_{\text{cor}} + F^y_{\text{cor}} + O(\Delta x),
\end{aligned} \tag{30}$$

where $\hat{\beta}$ is the same as (19) and

$$\tilde{\beta} = \frac{\beta(x_i, y_{m+})\cdot\beta(x_i, y_{m-})}{(1-\zeta)\cdot\beta(x_i, y_{m+}) + \zeta\cdot\beta(x_i, y_{m-})}, \tag{31}$$

with midpoints $y_{m+} = (y_J + y_{j+1})/2$ and $y_{m-} = (y_j + y_J)/2$, and

$$\begin{aligned} F^x_{\text{cor}} = \frac{\hat{\beta}\theta}{\beta(x_{m+}, y_j)\Delta x} \bigg\{ & \frac{\beta(x_{m+}, y_j)a(x_I, y_j)}{\theta \Delta x} + [\beta u_x] \\ & + \big(\theta \cdot (f - (\beta u_y)_y)(x_{I+}, y_j) + (1-\theta)\cdot(f - (\beta u_y)_y)(x_{I-}, y_j)\big)\frac{\Delta x}{2} \bigg\}, \end{aligned} \tag{32}$$

$$\begin{aligned} F^y_{\text{cor}} = \frac{\tilde{\beta}\zeta}{\beta(x_i, y_{m+})\Delta y} \bigg\{ & \frac{\beta(x_i, y_{m+})a(x_i, y_J)}{\zeta \Delta y} + [\beta u_y] \\ & + \big(\zeta \cdot (f - (\beta u_x)_x)(x_i, y_{J+}) + (1-\zeta)\cdot(f - (\beta u_x)_x)(x_i, y_{J-})\big)\frac{\Delta y}{2} \bigg\}, \end{aligned} \tag{33}$$

where $[\beta u_x] = [\beta u_n]n^1 - [\beta u_\tau]n^2$ and $[\beta u_y] = [\beta u_\tau]n^1 + [\beta u_n]n^2$.

Several variations could also used. For instance, on the right-hand side of (30), one may replace $(\beta u_x)_x$ by $f - (\beta u_y)_y$, or one may replace $(\beta u_y)_y$ by $f - (\beta u_x)_x$. Similar replacements could be made in (32) and (33). For our numerical tests, we used the $(\beta u_y)_y$-based version: $S^x u + S^y u = f \cdot (2-\theta)/2 + (\beta u_y)_y \cdot (\theta - \zeta)/2 + F^x_{\text{cor}} + F^y_{\text{cor}}$.

Appendix B. Computing tangential derivatives

Here we compute the tangential derivatives up to second order using the values of u at the interfacial points and two of the neighboring interfacial points (see Figure 15 for reference). By parametrizing our boundary of domain in either $y = y(x)$ or $x = x(y)$, whichever is a properly defined function, we first compute $dy/dx(x_I)$ or $dx/dy(y_I)$ and dw/dx or dw/dy correspondingly, where $w = u(x, y(x))$ or $w = u(x(y), y)$.

Suppose $y = y(x)$ is a properly defined function as in Figure 15; the unit tangent vector of the interface $\tau = (\tau_1, \tau_2)$ at (x_I, y_I) can be written as

$$(\tau_1, \tau_2) = \begin{cases} (1, dy/dx)/\sqrt{1 + (dy/dx)^2} & \text{if } \tau_1 > 0, \\ -(1, dy/dx)/\sqrt{1 + (dy/dx)^2} & \text{if } \tau_1 < 0. \end{cases} \tag{34}$$

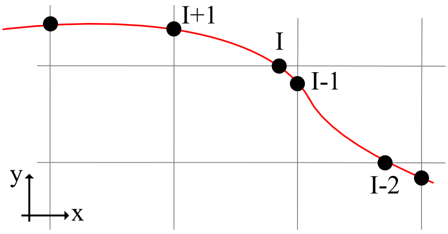


Figure 15. Three neighboring interface points are used to compute u_τ centered at point I.

Combining (34) with $u_\tau = u_x\tau_1 + u_y\tau_2$,

$$u_\tau = \frac{(u_x + u_y y_x)}{\sqrt{1+(dy/dx)^2}} \cdot \operatorname{sign}\tau_1 = \frac{w_x \cdot \operatorname{sign}\tau_1}{\sqrt{1+(dy/dx)^2}} = w_x \cdot \tau_1. \tag{35}$$

Suppose w_y and x_y are both accurate to second order; we have

$$\begin{aligned} u_\tau &= \frac{(w_x + O(h^2))}{\sqrt{1+(dy/dx + O(h^2))^2}} \sim \frac{(w_x + O(h^2))}{\sqrt{1+(dy/dx)^2 + O(h^2)}} \\ &= \frac{(w_x + O(h^2))}{\sqrt{1+(dy/dx)^2}\sqrt{1+O(h^2)}} \sim \frac{(w_x + O(h^2))}{\sqrt{1+(dy/dx)^2}}\big(1 - \tfrac{1}{2}O(h^2)\big) \\ &= \frac{w_x}{\sqrt{1+(dy/dx)^2}} + O(h^2). \end{aligned} \tag{36}$$

Both the derivatives with parametrization should be accurate up to second order. Using the interfacial and neighboring two interfacial points, we can compute y_x to second order by

$$ay(x_{I+1}) + by(x_I) + cy(x_{I-1}) = y_x(x_I) + O(h^2), \tag{37}$$

where

$$D = \Delta x_l \Delta x_r(\Delta x_r - \Delta x_l), \quad a = -\Delta x_l^2/D, \quad c = \Delta x_r^2/D, \quad b = -(a+c), \tag{38}$$

and $\Delta x_l = x_{I-1} - x_I$ and $\Delta x_r = x_{I+1} - x_I$. Note that the denominator D can be small when neighboring interface points are very close, which leads to numerical issues. This issue can be avoided by, for example, using (x_{I+1}, y_{I+1}), (x_I, y_I), and (x_{I-2}, y_{I-2}) instead of (x_{I+1}, y_{I+1}), (x_I, y_I), and (x_{I-1}, y_{I-1}) in Figure 15.

Similar set up for either w_x or w_y follows the above:

$$aw(x_{I+1}) + bw(x_I) + cw(x_{I-1}) = w_x(x_I) + O(h^2), \tag{39}$$

and the coefficients a, b, and c are exactly the same as above. Second-order bu_τ hence follows from the equation $bu_\tau = \pm w_x/y_x$ or $bu_\tau = \pm w_y/x_y$, depending on $|y_x| < 1$ or $|x_y| < 1$, and the sign adjustment comes from the sign of τ_1 and τ_2, respectively.

Appendix C. Relaxation

As discussed in Section 2.2.2, Picard iteration works well in many cases, but we found that it sometimes diverges. For this reason, as our standard iterative scheme, we instead use a simple relaxation scheme to bypass this difficulty and guarantee that the iterative scheme stops. The idea behind the relaxation scheme is to update the forcing term as

$$\boldsymbol{F}^{[k]} = \alpha_k \boldsymbol{F}^{[T_k]} + (1-\alpha_k)\boldsymbol{F}^{[k-1]}, \tag{40}$$

which is a mixture between the previous forcing $\boldsymbol{F}^{[k-1]}$ and the temporary forcing $\boldsymbol{F}^{[T_k]}$ that would have been used if a Picard update would have been followed. The parameter α_k is chosen to guarantee that $\boldsymbol{u}^{[k+1]}$ is not too far away from $\boldsymbol{u}^{[k]}$.

One cycle of the relaxation scheme goes as follows. Suppose $\boldsymbol{u}^{[k]}$ was computed by solving $A\boldsymbol{u}^{[k]} = \boldsymbol{F}^{[k-1]}$, and we now want to compute the next iteration. With $\boldsymbol{u}^{[k]}$, compute the temporary right-hand-side $\boldsymbol{F}^{[T_k]}$ by following the Picard update procedure from Section 2.2.2. A temporary solution $\boldsymbol{u}^{[T_{k+1}]}$ is then obtained by solving $A\boldsymbol{u}^{[T_{k+1}]} = \boldsymbol{F}^{[T_k]}$. Now the parameter α_k is determined to guarantee that $\boldsymbol{u}^{[k+1]}$ is not too far away from $\boldsymbol{u}^{[k]}$; to this end, define the ratio $r_k = \|\boldsymbol{u}^{[T_{k+1}]} - \boldsymbol{u}^{[k]}\|/\|\boldsymbol{u}^{[k]} - \boldsymbol{u}^{[k-1]}\|$. If this ratio is small ($r_k < 1$), then there is no need for relaxation and we set $\alpha_k = 1$. If this ratio is large ($r_k \geq 1$), then we set $\alpha_k = \rho/r_k$, where ρ is a preselected factor between 0 and 1. In practice, we pick $\|\cdot\| = \|\cdot\|_\infty$ and ρ to be between 0.9 and 0.99. With this relaxation scheme for the forcing $\boldsymbol{F}^{[k]}$, the solution is likewise updated as $\boldsymbol{u}^{[k+1]} = \alpha_k \boldsymbol{u}^{[T_{k+1}]} + (1-\alpha_k)\boldsymbol{u}^{[k]}$, as a mixture of the previous solution estimate $\boldsymbol{u}^{[k]}$ and the temporary solution estimate $\boldsymbol{u}^{[T_{k+1}]}$ that would have been used if a Picard update would have been followed.

The differences $u_d^{[k]} = \|\boldsymbol{u}^{[k+1]} - \boldsymbol{u}^{[k]}\|_\infty$ and $F_d^{[k]} = \|\boldsymbol{F}^{[k+1]} - \boldsymbol{F}^{[k]}\|_\infty$ are guaranteed to be decreasing as k increases if this relaxation procedure is followed. Specifically, the relaxation procedure leads to either $u_d^{[k]} = r_k u_d^{[k-1]}$ (if $r_k < 1$) or $u_d^{[k]} = \rho u_d^{[k-1]}$ (if $r_k \geq 1$). Therefore, $u_d^{[k]}$ is decreasing in k and hence the stopping criterion will be met in a finite number of iterations. Note that this stopping criterion, based on $\|\boldsymbol{u}^{[k+1]} - \boldsymbol{u}^{[k]}\|$, does not guarantee that the relaxation procedure's iterate $\boldsymbol{u}^{[k+1]}$ is actually close to the exact solution; nevertheless, one would expect that it should be at least a better estimate than the first iterate $\boldsymbol{u}^{[1]}$, which is the first-order accurate GFM solution; and in practice we find from the examples in Section 3 that the iterations terminate at a second-order accurate solution.

Acknowledgments

The research of Stechmann is partially supported by a Sloan Research Fellowship and National Science Foundation grant AGS-1443325. The authors thank J. Thomas Beale, Aleksandar Donev, Randall J. LeVeque, and anonymous reviewers for helpful comments.

References

[1] J. T. Beale and A. T. Layton, *On the accuracy of finite difference methods for elliptic problems with interfaces*, Commun. Appl. Math. Comput. Sci. **1** (2006), 91–119. MR Zbl

[2] J. Bedrossian, J. H. von Brecht, S. Zhu, E. Sifakis, and J. M. Teran, *A second order virtual node method for elliptic problems with interfaces and irregular domains*, J. Comput. Phys. **229** (2010), no. 18, 6405–6426. MR Zbl

[3] P. A. Berthelsen, *A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions*, J. Comput. Phys. **197** (2004), no. 1, 364–386. MR Zbl

[4] R. K. Crockett, P. Colella, and D. T. 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), no. 7, 2451–2469. MR Zbl

[5] S. Deng, K. Ito, and Z. Li, *Three-dimensional elliptic solvers for interface problems and applications*, J. Comput. Phys. **184** (2003), no. 1, 215–243. MR Zbl

[6] T. G. Fai, B. E. Griffith, Y. Mori, and C. S. Peskin, *Immersed boundary method for variable viscosity and variable density problems using fast constant-coefficient linear solvers, I: Numerical method and results*, SIAM J. Sci. Comput. **35** (2013), no. 5, B1132–B1161. MR Zbl

[7] ———, *Immersed boundary method for variable viscosity and variable density problems using fast constant-coefficient linear solvers, II: Theory*, SIAM J. Sci. Comput. **36** (2014), no. 3, B589–B621. MR Zbl

[8] A. L. Fogelson and J. P. Keener, *Immersed interface methods for Neumann and related problems in two and three dimensions*, SIAM J. Sci. Comput. **22** (2000), no. 5, 1630–1654. MR Zbl

[9] F. Gibou, R. P. Fedkiw, L.-T. Cheng, and M. Kang, *A second-order-accurate symmetric discretization of the Poisson equation on irregular domains*, J. Comput. Phys. **176** (2002), no. 1, 205–227. MR Zbl

[10] B. E. Griffith, X. Luo, D. M. McQueen, and C. S. Peskin, *Simulating the fluid dynamics of natural and prosthetic heart valves using the immersed boundary method*, Int. J. Appl. Mech. **1** (2009), no. 1, 137–177.

[11] B. E. Griffith and C. S. Peskin, *On the order of accuracy of the immersed boundary method: higher order convergence rates for sufficiently smooth problems*, J. Comput. Phys. **208** (2005), no. 1, 75–105. MR Zbl

[12] J. L. Hellrung, Jr., L. Wang, E. Sifakis, and J. M. Teran, *A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions*, J. Comput. Phys. **231** (2012), no. 4, 2015–2048. MR Zbl

[13] H. Ji, F. Lien, and E. Yee, *An efficient second-order accurate cut-cell method for solving the variable coefficient Poisson equation with jump conditions on irregular domains*, Int. J. Numer. Meth. Fluids **52** (2006), no. 7, 723–748. Zbl

[14] B. Kallemov, A. P. S. Bhalla, B. E. Griffith, and A. Donev, *An immersed boundary method for rigid bodies*, Commun. Appl. Math. Comput. Sci. **11** (2016), no. 1, 79–141. MR Zbl

[15] M. Kang, R. P. Fedkiw, and X.-D. Liu, *A boundary condition capturing method for multiphase incompressible flow*, J. Sci. Comput. **15** (2000), no. 3, 323–360. MR Zbl

[16] M.-C. Lai and C. S. Peskin, *An immersed boundary method with formal second-order accuracy and reduced numerical viscosity*, J. Comput. Phys. **160** (2000), no. 2, 705–719. MR Zbl

[17] M.-C. Lai and H.-C. Tseng, *A simple implementation of the immersed interface methods for Stokes flows with singular forces*, Comput. & Fluids **37** (2008), no. 2, 99–106. MR Zbl

[18] L. Lee and R. J. LeVeque, *An immersed interface method for incompressible Navier–Stokes equations*, SIAM J. Sci. Comput. **25** (2003), no. 3, 832–856. MR Zbl

[19] R. J. LeVeque and Z. Li, *The immersed interface method for elliptic equations with discontinuous coefficients and singular sources*, SIAM J. Numer. Anal. **31** (1994), no. 4, 1019–1044. MR Zbl

[20] Z. Li, *A note on immersed interface method for three-dimensional elliptic equations*, Comput. Math. Appl. **31** (1996), no. 3, 9–17. MR Zbl

[21] Z. Li and K. Ito, *Maximum principle preserving schemes for interface problems with discontinuous coefficients*, SIAM J. Sci. Comput. **23** (2001), no. 1, 339–361. MR Zbl

[22] Z. Li and M.-C. Lai, *The immersed interface method for the Navier–Stokes equations with singular forces*, J. Comput. Phys. **171** (2001), no. 2, 822–842. MR Zbl

[23] X.-D. Liu, R. P. Fedkiw, and M. Kang, *A boundary condition capturing method for Poisson's equation on irregular domains*, J. Comput. Phys. **160** (2000), no. 1, 151–178. MR Zbl

[24] X.-D. Liu and T. C. Sideris, *Convergence of the ghost fluid method for elliptic equations with interfaces*, Math. Comp. **72** (2003), no. 244, 1731–1746. MR Zbl

[25] S. H. Lui, *Numerical analysis of partial differential equations*, Wiley, Hoboken, NJ, 2011. MR Zbl

[26] A. N. Marques, J.-C. Nave, and R. R. Rosales, *A correction function method for Poisson problems with interface jump conditions*, J. Comput. Phys. **230** (2011), no. 20, 7567–7597. MR Zbl

[27] ———, *High order solution of Poisson problems with piecewise constant coefficients and interface jumps*, J. Comput. Phys. **335** (2017), 497–515. MR Zbl

[28] Y. Mori and C. S. Peskin, *Implicit second-order immersed boundary methods with boundary mass*, Comput. Methods Appl. Mech. Engrg. **197** (2008), no. 25-28, 2049–2067. MR Zbl

[29] C. S. Peskin, *Flow patterns around heart valves: a numerical method*, J. Comput. Phys. **10** (1972), no. 2, 252–271. Zbl

[30] ———, *Numerical analysis of blood flow in the heart*, J. Comput. Phys. **25** (1977), no. 3, 220–252. MR Zbl

[31] ———, *The immersed boundary method*, Acta Numer. **11** (2002), 479–517. MR Zbl

[32] D. Russell and Z. J. Wang, *A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow*, J. Comput. Phys. **191** (2003), no. 1, 177–205. MR Zbl

[33] L. M. Smith and S. N. Stechmann, *Precipitating quasigeostrophic equations and potential vorticity inversion with phase changes*, J. Atmos. Sci. **74** (2017), no. 10, 3285–3303.

[34] M. Sussman, E. Fatemi, P. Smereka, and S. Osher, *An improved level set method for incompressible two-phase flows*, Comput. & Fluids **27** (1998), no. 5–6, 663–680. Zbl

[35] M. Sussman, K. M. Smith, M. Y. Hussaini, M. Ohta, and R. Zhi-Wei, *A sharp interface method for incompressible two-phase flows*, J. Comput. Phys. **221** (2007), no. 2, 469–505. MR Zbl

[36] A. Wiegmann and K. P. Bube, *The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions*, SIAM J. Numer. Anal. **37** (2000), no. 3, 827–862. MR Zbl

[37] S. Xu and Z. J. Wang, *An immersed interface method for simulating the interaction of a fluid with moving boundaries*, J. Comput. Phys. **216** (2006), no. 2, 454–493. MR Zbl

[38] ———, *Systematic derivation of jump conditions for the immersed interface method in three-dimensional flow simulation*, SIAM J. Sci. Comput. **27** (2006), no. 6, 1948–1980. MR Zbl

Received August 1, 2018. Revised March 12, 2019.

CHUNG-NAN TZOU: ctzou@wisc.edu
Department of Mathematics, University of Wisconsin–Madison, Madison, WI, United States

SAMUEL N. STECHMANN: stechmann@wisc.edu
Department of Mathematics, University of Wisconsin–Madison, Madison, WI, United States

mathematical sciences publishers msp

COMM. APP. MATH. AND COMP. SCI.
Vol. 14, No. 2, 2019
dx.doi.org/10.2140/camcos.2019.14.149

2D FORCE CONSTRAINTS IN THE METHOD OF REGULARIZED STOKESLETS

ONDREJ MAXIAN AND WANDA STRYCHALSKI

For many biological systems that involve elastic structures immersed in fluid, small length scales mean that inertial effects are also small, and the fluid obeys the Stokes equations. One way to solve the model equations representing such systems is through the Stokeslet, the fundamental solution to the Stokes equations, and its regularized counterpart, which treats the singularity of the velocity at points where force is applied. In two dimensions, an additional complication arises from Stokes' paradox, whereby the velocity from the Stokeslet is unbounded at infinity when the net hydrodynamic force within the domain is nonzero, invalidating any solutions that use the free space Stokeslet. A straightforward computationally inexpensive method is presented for obtaining valid solutions to the Stokes equations for net nonzero forcing. The approach is based on modifying the boundary conditions of the Stokes equations to impose a mean zero velocity condition on a large curve that surrounds the domain of interest. The corresponding Green's function is derived and used as a fundamental solution in the case of net nonzero forcing. The numerical method is applied to models of cellular motility and blebbing, both of which involve tether forces that are not required to integrate to zero.

1. Introduction

Stokes flow refers to the regime of viscous flow where inertial effects are small, and the Navier–Stokes equations simplify to the Stokes equations. For fluid-structure interaction problems in cell biology, such as an elastic red blood cell membrane deforming in capillary flow [28; 29], the small length scales of the cell diameter ($\sim 10\,\mu$m) lead to a small Reynolds number. Other important phenomena in cell biology that involve zero Reynolds number flow are cell motility [20; 38] and microorganism swimming [7; 15; 17].

Because the Stokes equations are linear, boundary integral and boundary element methods can be used to determine the velocity and pressure fields that come from a collection of forces [26; 27]. The velocity field generated from a point force is known as a *Stokeslet*. One problem that arises when using the Stokeslet in practice

MSC2010: 65M80, 74F10, 92C37.
Keywords: fluid-structure interaction, Stokes flow, Stokes' paradox, regularized Stokeslets.

is the singularity at the point where the force arises. For closed interfaces, this singularity is integrable, but careful numerical quadratures are necessary to correctly calculate the velocity and pressure [26; 27]. In [9], Cortez introduced the method of regularized Stokeslets to overcome the singularities in both the pressure and velocity expressions for forces located at scattered points. Instead of the force being applied at a point, the force is applied over a small ball of radius ϵ. The regularized Stokeslet and pressure expressions are then obtained analytically from the particular function used to represent the small ball. The method of regularized Stokeslets can also be used for closed surfaces, bypassing the associated issues with numerical quadrature [26].

It is convenient to model and simulate fluid-structure interaction problems in two dimensional domains where model parameter studies can be conducted in a computationally inexpensive manner. Data visualization is also easier in 2D than in 3D. The free space Stokes equations in 2D are actually ill-posed because the velocity obtained from the free space Stokeslet is *unbounded* at infinity when there is a nonzero net hydrodynamic force acting within the domain of flow. This contradicts the assumption in the derivation of the Stokeslet that the velocity is zero at infinity and renders the problem ill-posed [9; 21; 26]. Numerical simulations of such systems can therefore lead to unphysical spurious velocities. The phenomenon of unbounded velocities in systems with nonzero net force, especially as $\|\boldsymbol{x}\| \to \infty$, is usually referred to as Stokes' paradox [38]. We emphasize that this is a unique feature of Stokes flow in 2D. In 3D, the problem is well-posed; the Stokeslet decays to zero at infinity regardless of the net forcing, so the boundary condition is satisfied and the solution is valid for any collection of forces with bounded magnitude.

One way to ensure that the 2D velocity is valid is to add conditions to the original system of equations. The most straightforward way to do this is to impose additional boundary conditions within the region of interest. The method of regularized Stokeslets was employed in [9] to simulate the flow due to a cylinder moving at an imposed velocity. In [1; 4; 12], the method of images was used to add additional Stokeslets outside of the flow domain that enforce a zero boundary condition near a plane wall. In these approaches, there is an additional constraint on the *velocity* that leads to valid solutions near the immersed objects of interest. However, if no boundary conditions within the flow domain are specified by the model of the physical system (e.g., when modeling flexible fibers in Stokes flow [5; 11; 35]), a different approach must be used.

One such approach is to enforce a constraint on the *force* rather than the velocity, in particular that the net hydrodynamic force over the entire domain be zero. Sometimes, this constraint comes naturally, such as in models of flagellar swimming [40] or fibers immersed in a background flow [5]. However, the only a priori requirement of fluid-structure interaction in Stokes flow is that the hydrodynamic force at a point is exactly balanced by the internal and external forces on the immersed structures [22]. In fact, there are many systems with zero Reynolds number that contain force imbalances, including any system that contains tether forces or objects tied to boundaries. For example, Cortez's model of a moving cylinder [9] had a nonzero net hydrodynamic force within the domain of flow. In this case, one potential solution is to subtract the mean force from the force at each point, which automatically gives a zero-sum total force. Here we show this approach can result in nonphysical, displaced equilibrium states.

Teran and Peskin [33] treated the problem of unbalanced forces within the immersed boundary (IB) method [24; 25] by adding a unique, constant velocity throughout the periodic domain to ensure that the net force is zero for all time. In the formulation from [33], an additional constant velocity is permitted because the equations are simulated on a periodic domain, where the solution is unique up to a constant. In this case, it is required that the net force be zero [3]. The net zero force requirement in a periodic IB method presents some challenges. For example, tether forces *must* be introduced in the domain in order to simulate body forces (as the authors did in [33] when modeling peristaltic pumping). In order for the immersed structures to remain stationary, the tether spring stiffness must be large, which in turn increases the overall stiffness of the numerical scheme and the cost of the IB method formulation as a whole.

We present a method to simulate models in 2D Stokes flow with net nonzero forcing using the method of regularized Stokeslets. We accomplish this by surrounding the domain by a large circle and constraining the *mean velocity* on the circle to be zero. Given this boundary condition, we derive the corresponding Green's function and show that a mean zero velocity at the large circle can be achieved simply by adding a constant velocity to the free space Stokeslet solution throughout a large domain of flow. In this way, we avoid having to solve a linear system on the large circle (as in [38]). This observation results in an algorithm that is very straightforward to implement. After presenting our method in Sections 2 and 3, we show in Section 4 how it can be applied to 2D models of cells immersed in viscous fluid. In the process, we compare our formulation to both the explicit zero velocity condition on the large circle, e.g., from [38], and the force-free formulation obtained from subtracting the mean force at each point.

2. Mathematical framework

The steady Stokes equations in two dimensions are

$$\mu\Delta\boldsymbol{u} - \nabla p = -\boldsymbol{f}, \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{2}$$

where μ is the fluid viscosity, p is the pressure, $\boldsymbol{u}$ is the fluid velocity, and $\boldsymbol{f}$ is the hydrodynamic force, exactly equal to the external applied force that comes from fibers or other structures immersed in the fluid [22]. We begin by summarizing the method of regularized Stokeslets [9] for computing $\boldsymbol{u}$ and p from (1) and (2). Then we present the modification for addressing Stokes' paradox.

2.1. ***Method of regularized Stokeslets.*** In the method of regularized Stokeslets, a force of strength $\boldsymbol{f_0}$ is distributed primarily (but not entirely) over a small ball centered on a point $\boldsymbol{x_0}$, so that

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f_0}\phi_\epsilon(\boldsymbol{x}-\boldsymbol{x_0}). \tag{3}$$

The Stokes equations can be solved with the force in (3) to derive the resulting velocity and pressure from a given "blob" or "cutoff" function ϕ_ϵ. For example, if

$$\phi_\epsilon(\boldsymbol{x}) = \frac{3\epsilon^3}{2\pi(\|\boldsymbol{x}\|^2+\epsilon^2)^{5/2}}, \tag{4}$$

then

$$p^\epsilon(\boldsymbol{x},\boldsymbol{x_0}) = \frac{1}{2\pi}(\boldsymbol{f_0}\cdot(\boldsymbol{x}-\boldsymbol{x_0}))\left(\frac{r_0^2+2\epsilon^2+\epsilon\sqrt{r_0^2+\epsilon^2}}{(\sqrt{r_0^2+\epsilon^2}+\epsilon)(r_0^2+\epsilon^2)^{3/2}}\right) \tag{5}$$

and

$$\begin{aligned}\boldsymbol{u}^\epsilon(\boldsymbol{x},\boldsymbol{x_0}) = -\frac{\boldsymbol{f_0}}{4\pi\mu}&\left(\ln(\sqrt{r_0^2+\epsilon^2}+\epsilon) - \frac{\epsilon(\sqrt{r_0^2+\epsilon^2}+2\epsilon)}{(\sqrt{r_0^2+\epsilon^2}+\epsilon)\sqrt{r_0^2+\epsilon^2}}\right)\\ &+\frac{1}{4\pi\mu}(\boldsymbol{f_0}\cdot(\boldsymbol{x}-\boldsymbol{x_0}))(\boldsymbol{x}-\boldsymbol{x_0})\frac{\sqrt{r_0^2+\epsilon^2}+2\epsilon}{(\sqrt{r_0^2+\epsilon^2}+\epsilon)^2\sqrt{r_0^2+\epsilon^2}}\end{aligned} \tag{6}$$

are the pressure and velocity that result from the force in (3), where $r_0 = \|\boldsymbol{x}-\boldsymbol{x_0}\|$. The derivation of these expressions can be found in [9]. Notice that for $r_0 \gg \epsilon$, the standard Stokeslet expressions [26] are recovered,

$$p(\boldsymbol{x},\boldsymbol{x_0}) = \frac{\boldsymbol{f_0}\cdot(\boldsymbol{x}-\boldsymbol{x_0})}{2\pi r_0^2}, \tag{7}$$

$$\boldsymbol{u}(\boldsymbol{x},\boldsymbol{x_0}) = -\frac{\boldsymbol{f_0}}{4\pi\mu}\ln r_0 + (\boldsymbol{f_0}\cdot(\boldsymbol{x}-\boldsymbol{x_0}))\frac{(\boldsymbol{x}-\boldsymbol{x_0})}{4\pi\mu r_0^2}. \tag{8}$$

The pressure and velocity resulting from a collection of forces $\boldsymbol{f_k}$ spread around a collection of points $\boldsymbol{x_k}$ is simply a superposition of the results from (5) and (6). It is easy to see that if $\sum_k \boldsymbol{f_k} \neq \boldsymbol{0}$, the velocity in (6) or (8) is unbounded as $\|\boldsymbol{x}\| \to \infty$, and the boundary conditions $\boldsymbol{u} \to \boldsymbol{0}$ are not satisfied as $\|\boldsymbol{x}\| \to \infty$.

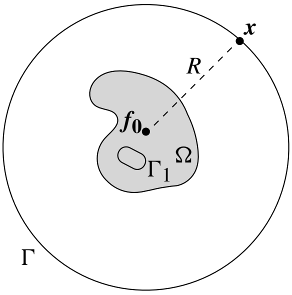


Figure 1. Ω denotes the region bounding immersed interfaces and/or point forces. Γ_1 indicates an immersed interface and $\boldsymbol{f_0}$ denotes a point force at $\boldsymbol{x_0}$ enclosed by Ω. A circle of radius R is denoted by Γ.

2.2. ***Modification for nonzero net force.*** Suppose that all of the forces $\boldsymbol{f_k}$ and immersed interfaces in a model system are located within a domain Ω (see Figure 1). We note that Ω is not necessarily an immersed interface, but rather a sort of "bounding box" in which all of the forces are contained. If there are no other boundary conditions within Ω, such as a specified velocity on a curve Γ_1 contained within the domain, the problem is ill-posed and the free space solution for the velocity in (6) is not valid, even near Ω. To construct a mathematically valid solution inside some space containing Ω, we surround Ω with a large circle, denoted by Γ with radius R (illustrated in Figure 1). One approach from [38] is to enforce a zero velocity boundary condition at every point on the discretized circle. The resulting linear system obtained from (6) is well-conditioned when ϵ is of magnitude less than or equal to the discrete point spacing on the large circle. The system can be solved for the forces required to obtain a zero velocity on the large circle. These forces can then be used in (6) to compute the velocity at locations enclosed by the smaller domain Ω. We note that this requires a linear system to be solved at each time value when simulating a model of a dynamic process.

We take a slightly different approach. Instead of requiring $\boldsymbol{u}|_\Gamma = \boldsymbol{0}$ at every point, we solve Stokes equations with a slightly weaker boundary condition, that the average value of $\boldsymbol{u}$ on Γ be zero: $\langle \boldsymbol{u}\rangle|_\Gamma = \boldsymbol{0}$. Notice that as $R \to \infty$, the regularized Stokeslet solution in (6) converges to the free space Stokeslet in (8) and the velocity on the large circle is approximately *constant* because of the dominance of the radially symmetric first term in (8). If the velocity on the large circle is constant, the two conditions are equivalent. Imposing the mean velocity condition allows us to add an extra velocity $\boldsymbol{u^R}$ throughout the domain so that the average velocity on the large circle is zero.

We begin by deriving the velocity $\boldsymbol{u^R}$ in the case of a single point force followed by the generalization to multiple forces by superposition. Let $\boldsymbol{f_0}$ be the force at a point $\boldsymbol{x_0}$ in Ω. Let $\boldsymbol{x}$ be a point on the large circle Γ (see Figure 1), and let

$r_0 = \|\boldsymbol{x} - \boldsymbol{x_0}\|$. Since $\boldsymbol{x}$ is on a large circle with arbitrarily large radius R, $r_0(\boldsymbol{x}) \gg \epsilon$ for all $\boldsymbol{x} \in \Gamma$, and we can represent the velocity at the large circle using the standard Stokeslet. Thus, the velocity at $\boldsymbol{x}$ due to the force $\boldsymbol{f_0}$ applied at $\boldsymbol{x_0}$ is given by (8).

Using s as the arclength parameter and treating $r_0 = R$ as constant, the average value of $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{x_0})$ is

$$\begin{aligned}\langle \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{x_0})\rangle &= \frac{1}{2\pi R}\int_\Gamma \left(-\frac{\boldsymbol{f_0}}{4\pi\mu}\ln r_0 + (\boldsymbol{f_0}\cdot(\boldsymbol{x}-\boldsymbol{x_0}))\frac{\boldsymbol{x}-\boldsymbol{x_0}}{4\pi\mu r_0^2}\right) ds \\ &= \frac{1}{2\pi R}\left(-\frac{\boldsymbol{f_0}}{4\pi\mu}\ln R\int_\Gamma ds + \frac{1}{4\pi\mu R^2}\int_\Gamma (\boldsymbol{f_0}\cdot(\boldsymbol{x}-\boldsymbol{x_0}))(\boldsymbol{x}-\boldsymbol{x_0})\, ds\right) \\ &= -\frac{\boldsymbol{f_0}}{4\pi\mu}\ln R + \left(\frac{1}{2\pi R}\right)\frac{1}{4\pi\mu R^2}\int_\Gamma (\boldsymbol{f_0}\cdot(\boldsymbol{x}-\boldsymbol{x_0}))(\boldsymbol{x}-\boldsymbol{x_0})\, ds. \qquad (9)\end{aligned}$$

The last equality used the fact that $\int_\Gamma ds = 2\pi R$. Computing the second integral, we begin by changing to an angle parametrization of Γ via $s = R\theta$,

$$\frac{1}{2\pi R}\int_\Gamma (\boldsymbol{f_0}\cdot(\boldsymbol{x}-\boldsymbol{x_0}))(\boldsymbol{x}-\boldsymbol{x_0})\, ds = \frac{1}{2\pi R}\int_0^{2\pi} (\boldsymbol{f_0}\cdot(\boldsymbol{x}-\boldsymbol{x_0}))(\boldsymbol{x}-\boldsymbol{x_0})R\, d\theta. \qquad (10)$$

Because the circle is rotationally invariant, we can place the x axis on the same direction as $\boldsymbol{f_0}$ without loss of generality. Therefore, let $\boldsymbol{f_0} = f\binom{1}{0}$. Furthermore, in the limit $R \to \infty$, $\boldsymbol{x} - \boldsymbol{x_0} = \boldsymbol{x} = \binom{R\cos\theta}{R\sin\theta}$. Then (10) simplifies to

$$\begin{aligned}\frac{1}{2\pi R}\int_0^{2\pi} (\boldsymbol{f_0}\cdot(\boldsymbol{x}-\boldsymbol{x_0}))(\boldsymbol{x}-\boldsymbol{x_0})R\, d\theta &= \frac{1}{2\pi}\int_0^{2\pi} fR\cos\theta \begin{pmatrix} R\cos\theta \\ R\sin\theta \end{pmatrix} d\theta \qquad (11) \\ &= \frac{R^2 f}{2}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{R^2 \boldsymbol{f_0}}{2}. \qquad (12)\end{aligned}$$

Substituting (12) into (9), we have the average velocity over the large circle given the force $\boldsymbol{f_0}$ at $\boldsymbol{x_0}$ as

$$\langle \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{f_0})\rangle = \frac{\boldsymbol{f_0}}{4\pi\mu}\left(\tfrac{1}{2} - \ln R\right). \qquad (13)$$

Our goal is to impose a boundary condition on the large circle. Rather than impose a boundary condition pointwise, we impose a weaker condition on the *mean velocity* on the large circle, namely that the mean velocity is zero. It follows immediately that this can be done by subtracting the constant velocity in (13) throughout the domain of flow. The additional velocity due to a point force $\boldsymbol{f_0}$ is therefore

$$\boldsymbol{u^R}(\boldsymbol{f_0}) = -\frac{\boldsymbol{f_0}}{4\pi\mu}\left(\tfrac{1}{2} - \ln R\right). \qquad (14)$$

In the case of multiple time-dependent forces $\boldsymbol{f_k}(t)$ (for example, forces that come from an interface such as Γ_1 in Figure 1), the constant velocity is simply the superposition of velocities from (14):

$$\boldsymbol{u}^{\boldsymbol{R}}(t) = \sum_{k=1}^{N} \boldsymbol{u}^{\boldsymbol{R}}(\boldsymbol{f}_k(t)) = \sum_{k=1}^{N} -\frac{\boldsymbol{f}_k(t)}{4\pi\mu}\left(\tfrac{1}{2} - \ln R\right). \tag{15}$$

This velocity is added throughout the domain of flow to ensure that $\langle \boldsymbol{u}(t)\rangle|_\Gamma = 0$. In effect, the solution from (6) and (15) together form the *Green's function for Stokes equations with a mean zero boundary condition on the circle of radius R*. We note that the Green's function would change if the domain was surrounded by a large square with edge length $2R$ as opposed to a circle of radius R. However, any geometry, such as a square, can be thought of as bounded by two concentric large circles (for a square centered at 0 with edge length $2R$, the points on the square are between concentric circles of radius R and $R\sqrt{2}$). Because the velocity dependence on R is weak for large R (the derivative of $\boldsymbol{u}^{\boldsymbol{R}}$ scales with $1/R$), altering the geometry of the boundary results in small changes in the velocity on the bounding region.

It is no coincidence that (15) expresses the average value of the velocity due to forces of strength $-\boldsymbol{f}_k$. By adding the constant velocity in (15), we are effectively adding a force on the large circle that *has the effect of adding an equal and opposite force within the region of interest* Ω. Meanwhile, the addition of a constant velocity throughout the domain results in a relative velocity profile that is unchanged from that computed by (6). Our approach contrasts with adding more Stokeslets at arbitrary locations in the domain Ω, which could be problematic because the relative local profile (and subsequent physical conclusions) are dependent on the locations of the additional Stokeslets. However, our approach of imposing forces on the large circle does result in the introduction of a much larger length scale in the problem; the length scale becomes R, the radius of the large circle, instead of the length scale of the immersed objects.

The addition of this constant velocity has no effect on the pressure profile calculated from (5). Because a constant velocity is added, no pressure gradient is generated within the domain. Equivalently, our addition of a constant velocity is a shortcut around explicitly adding forces on the large circle that enforce the zero boundary condition exactly (e.g., [38]). Due to the nature of the pressure solution in (5) (i.e., that it decays as 1/length), the additional forces from the boundary condition on the large circle have no effect on the local pressure profile for large R.

We also note that the choice of blob in (4) yields the resulting analytical expressions for the regularized pressure and velocity in (5) and (6), respectively. We present these expressions derived from ϕ_ϵ because we use them in our numerical simulations. The derivation of $\boldsymbol{u}^{\boldsymbol{R}}$ is independent of the regularized Stokeslet because it is derived from the true free space Stokeslet. The large circle Γ is assumed to be far enough from the domain of interest that the two are equivalent.

The method we present here is therefore compatible with any regularization kernel, including compactly supported immersed boundary kernels [2]. However, IB kernels are generally used over a periodic fluid grid, not over free space, and so the method of [33] is more appropriate in that context.

3. Discretization

In general, we begin with a collection of N points $\boldsymbol{x_k}$ with forces $\boldsymbol{f_k}$ in some domain Ω. At each time step, we compute the velocity of each point $\boldsymbol{x_i}$ as

$$\boldsymbol{u}(\boldsymbol{x_i}) = \boldsymbol{u^R} + \sum_{k=1}^{N} \boldsymbol{u}^{\epsilon}(\boldsymbol{x_i}, \boldsymbol{x_k}), \tag{16}$$

where $\boldsymbol{u^R}$ is given by (15) and $\boldsymbol{u}^{\epsilon}(\boldsymbol{x_i}, \boldsymbol{x_k})$ is given by (6). Because $\boldsymbol{u^R}$ is constant throughout the domain, the calculation in (16) is $\mathbb{O}(2N + (2N)^2)$ operations. The first $\mathbb{O}(2N)$ operations arise from the computation of the constant additional velocity in (15). The second $\mathbb{O}((2N)^2)$ operations come from computing the regularized velocities in (6) for all of the points.

We compare this operation count to alternative formulations. Suppose that the large circle was discretized with M points and the forces on the large circle were solved for explicitly, as in [38]. This calculation is a $2M \times 2M$ linear solve and requires $\mathbb{O}((2M)^2)$ operations using GMRES or $\mathbb{O}((2M)^3)$ operations if done directly. In addition, the calculation of the added velocities at each point in the domain from the forces on the large circle requires another $\mathbb{O}(2NM)$ flops, and it is unclear how to choose the number and location of the M points. Alternatively, the addition of more Stokeslets in a method similar to the method of images [1; 4; 12] would require $\mathbb{O}(2SN)$ operations to compute the added velocity, where S is the number of added Stokeslets. Our added velocity is computed in $2N$ flops, making it much more efficient than any of these alternatives.

3.1. *Choosing the radius.*

Central to our method is the assumption that the velocity computed from (8) on the large circle Γ is relatively constant. The validity of this assumption dictates a lower bound on R. In order to test the variation of the velocity on the large circle Γ, we first impose a force of $\boldsymbol{f_0} = \binom{1}{0}$ at the origin. Next, we use (8) to compute and measure the velocity from the Stokeslet (i.e., the velocity without the addition of $\boldsymbol{u^R}$) on the large circle.

Specifically, we discretize the circle Γ with $N = 100$ points and quantify velocity variability by defining

$$\sigma_u(R) = \max_i \left| \frac{u_x^i - \bar{u}_x}{\bar{u}_x} \right|. \tag{17}$$

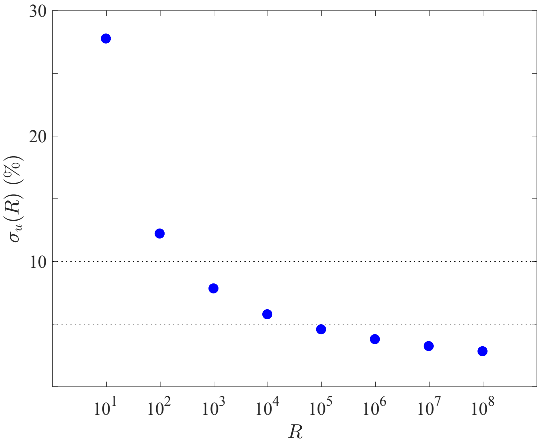


Figure 2. Variation of the velocity on the large circle Γ for different values of R. Dotted lines indicate 10% and 5% variation, $\sigma_u(R)$.

Here the index i runs from 1 to $N = 100$, u_x^i refers to the velocity in the x direction (the direction of the force) at point i, and $\bar{u}_x$ refers to the arithmetic mean of u_x taken over Γ.

Figure 2 shows the values of $\sigma_u(R)$ (as a percentage) for different values of R. We observe a variation in the velocity of less than 10% for $R \geq 10^3$ and a variation of less than 5% for $R \geq 10^5$.

With this in mind, we specify the lower limit on R, $R \geq 10^3$. We can find an upper limit on R based on the Reynolds number. Let v and L be the relevant velocity and length scales. Our systems have values of viscosity μ and density ρ of the same magnitude as water, so that $\mathrm{Re} = \rho v L/\mu = 10^6 v L$ is the relevant Reynolds number. Our model systems are from applications in cell biology, so the relevant velocity scale is in $\mu\mathrm{m/s}$. Therefore, we take $v = 10^{-6}$ m/s and $\mathrm{Re} = L$, where L is the relevant length scale. In order for Stokes flow to be valid, we need ($\mathrm{Re} \ll 1$), which we define to be $\mathrm{Re} \leq 0.1$. Then the relevant length scale cannot exceed $0.1\,\mathrm{m} = 10^5\ \mu\mathrm{m}$. Because we have confined the domain and effectively introduced forces on the large circle, the radius of the large circle is now the largest relevant length scale, and we have determined an upper bound on R, $R \leq 10^5\ \mu$m. We note that this upper bound may change depending on the characteristic time and length scales used to compute the Reynolds number, but it is straightforward to derive it as we have here. Thus, we have determined in general that $10^3 \leq R \leq 10^5$. For the examples in Section 4, we choose $R = 10^3\ \mu$m to ensure the validity of the Stokes equations for these systems. This value of R results in a velocity variation from (17) that is less than 10%. In addition, we did not find any additional stiffness when using R in this range (for larger R, e.g., $R = 10^{16}$, the velocities in (15) would increase, thereby increasing the overall problem stiffness).

4. Examples

The motivation for this work is *tether forces* that arise in the modeling of some biological systems. These are forces that penalize displacement from an initial or resting configuration; points on an immersed object are physically *tethered* to other points in 2D space. We begin by considering a simplified system of tethered particles. This motivating example establishes the need for the additional velocity in (15). We then present a model of a cell motility problem where tether forces are useful for modeling the cell's external environment. We conclude by analyzing a boundary integral model of cellular blebbing with nonzero net forcing that has already been used for modeling bleb initiation and amoeboid cell motility [14; 19; 20]. In all cases, all of the objects are flexible so that no boundary conditions are provided from the physics of each model system.

4.1. ***A motivating example.*** We are interested in modeling the motion of a cell through a viscoelastic structure called the extracellular matrix (ECM). For example, our model of the ECM represents a lattice of collagen fibrils immersed in a viscous fluid. Elasticity of the ECM can be modeled in 2D by a lattice of points that are tied to specified reference points by springs. In order to demonstrate why our methodology is key for modeling this process, we introduce a set of $N = 32$ points distributed on a circle of radius r, shown in Figure 3, top. The points (solid blue dots) are centered at $(10, 0)$ and are tethered to two sets of fixed points (hollow black diamonds) centered at $(80, 0)$ and $(-80, 0)$. For simplicity, we set

$$\boldsymbol{F}_i(t) = -k_{\text{teth}}(\boldsymbol{X}_i(t) - \boldsymbol{X}_i^R + \boldsymbol{X}_i(t) - \boldsymbol{X}_i^L), \tag{18}$$

where $\boldsymbol{X}_i^R$ and $\boldsymbol{X}_i^L$ stand for the position of point i on the right and left fixed circles, respectively. The parameters for the system are $\mu = 1\,\text{Pa s}$, $k_{\text{teth}} = 1\,\text{pN}/\mu\text{m}^2$, $\epsilon = 2\pi r/N =$ the point spacing (although the dynamics are independent of the parameters μ, k_{teth}, and ϵ). The initial configuration results in a force imbalance, with a net force to the left (negative horizontal direction) on the set of moving points. Physically, we expect the points to move in the negative horizontal direction and approach their equilibrium position exactly between the two fixed fibers, i.e., centered at $x = 0$. However, this is not always what occurs when using (6) to update the velocities.

Consider (6). The dominant part of the first term is $-\ln r = -\ln\|\boldsymbol{x} - \boldsymbol{x_0}\|$, which results in a velocity that goes in the direction *opposite* the force. The second term in (6) is $\mathbb{O}(1, \epsilon^{-2})$ and contributes to the velocity in the same direction as the force. In order for the dynamics to match our physical intuition, the cumulative contribution of the second term at each point must be greater than that of the first term, so either $\ln r \approx 1$ or $\epsilon \ll 1$. Thus, as r becomes large, we expect unphysical behavior. For an r value as small as $r \approx 20$ (determined empirically), no value of ϵ (larger than

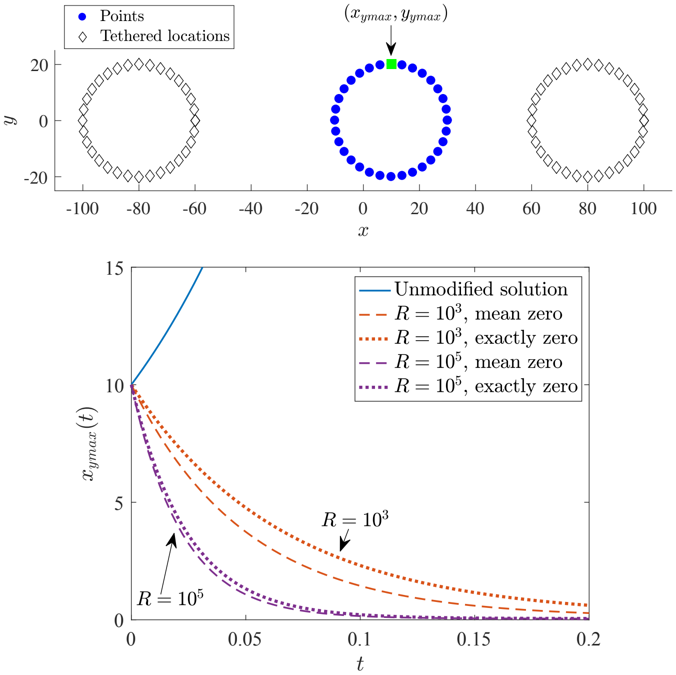


Figure 3. Motivating example of a system of points tethered in place, initially out of equilibrium. Top: initial configuration of the system. The solid blue dots show the points, while the unfilled black diamonds show the corresponding tethering locations. Bottom: x coordinate of the point with largest y coordinate, x_{ymax} (filled green square in top). Without any corrections, the motion is in the positive horizontal ($+x$) direction independent of number of points and ϵ (blue line). This nonphysical behavior can be corrected by adding the velocity $\boldsymbol{u}^{\boldsymbol{R}}$ in (15). We show data for the values $R = 10^3$ (orange) and $R = 10^5$ (purple). Dashed lines give the dynamics for the mean velocity subtraction in (16). Dotted lines show dynamics when the boundary condition $\boldsymbol{u} = \boldsymbol{0}$ is exactly enforced on a large circle of radius R discretized with $N = 100$ points.

machine epsilon) yields physical results. Figure 3, bottom, shows the horizontal (x) position of the point with the largest y coordinate over time (marked with a green square as (x_{ymax}, y_{ymax}) in Figure 3, top), where the velocity is computed by (6). We observe unphysical motion in the positive horizontal direction (blue line), moving the points to the right and resulting in an increased force imbalance as time increases. While this behavior *dominates* for large values of r, it is also present for smaller r and needs to be corrected to give proper, physically correct simulation results.

Our solution with the additional velocity given in (15) gives the expected behavior. For the initial configuration in Figure 3, top, the motion of the points computed with the velocity in (16) shifts the points in the negative horizontal direction (left), allowing them to approach their steady state positions at $x = 0$. Specifically, the horizontal component of the velocity at the origin can be decomposed into $\boldsymbol{u}^{\epsilon} \approx 125$

and $\boldsymbol{u}^{\boldsymbol{R}} \approx -325$ when $R = 10^3$, where the contribution of $\boldsymbol{u}^{\boldsymbol{R}}$ is necessarily greater to obtain the correct physical motion. Figure 3, bottom, shows the horizontal position of the point with the largest y coordinate for values of R that satisfy our derived bounds, $R = 10^3$ (orange lines) and $R = 10^5$ (purple lines). Dashed lines show the solution obtained from (16), and dotted lines show the solution obtained from discretizing the circle of radius R with $N = 100$ points and explicitly enforcing $\boldsymbol{u} = \boldsymbol{0}$ at those points by determining the additional forces $\boldsymbol{f}$ on the large circle via solving a linear system of equations. The linear system of equations in this case has the form $\boldsymbol{U} = \boldsymbol{MF}$, where $\boldsymbol{M}$ is a dense $2N \times 2N$ matrix. To get an exact solution, we solve this directly with LU factorization, although it could also be done with GMRES.

We observe that the mean velocity solution gives faster dynamics than the discretized large circle solution. As R increases, the velocity on the large circle approaches a constant value and the solutions approach the same curve (as shown previously in Figure 2). Further, the maximum difference of $1.0\,\mu$m in the x-coordinate between the two curves for $R = 10^3$ is only 5.2% of the smallest system length scale r, and for $R = 10^5$ this difference decreases to 1.8%.

The choice for the value of R in (15) affects the velocity and the dynamics of the system. However, we note that additional forces solved for by enforcing additional boundary conditions on any geometry C, i.e., $\boldsymbol{u}|_C = \boldsymbol{0}$, would also affect dynamics of the system. For the values of R within the range $10^3 \leq R \leq 10^5$, the *steady-state* behavior of our model and relevant timescales are shown in Figure 3, bottom, to be nearly identical, with the timescales differing by about a factor of 2. If accurate transient results are desired, the value of R can be tuned to give dynamics that fit within the relevant timescales, with the caveat that increasing R leads to a larger length scale.

We note another feature of this example: the initial force on the configuration shown in Figure 3, top, is *uniform across all of the points*. Suppose one wanted to treat the force imbalance in this example by subtracting the mean force from each point, so the total force sums to zero. Because each point has the same force on it, subtracting the mean force gives zero force and zero velocity at every point. We have therefore shown that subtracting the mean force can create artificial equilibrium configurations. This phenomenon occurs not just in this simple example, but also in a more complicated model of cell motility as discussed in Section 4.2. Thus, while subtracting the mean of the forces maintains the relevant system length scales, doing so can introduce errors in the resting position of the system. Which avenue to choose in this trade-off is application dependent.

4.2. ***Model of cell motility.*** Cell motility is an essential process for wound healing, cancer metastasis, and immune responses [30]. In three dimensions, a cell can utilize multiple mechanisms to migrate through the surrounding extracellular matrix

(ECM) [36; 37; 42]. The ECM is a dense network of collagen fibers [13, Figure 1]. Previous studies have used 2D agent based/finite element models [36; 37] to study the effectiveness of bleb-based and protrusion-based mechanisms in different ECM environments. In [42], the authors simulated a variety of mechanisms on a cell with a rigid nucleus via force balance equations. Our goal is to extend this model to a flexible nucleus, where the fluid-structure coupling is treated explicitly via the method of regularized Stokeslets.

We focus here on the movement of the cell via a finger-like protrusive mechanism [18; 42]. In this mechanism, the elastic cell cortex generates random actin-based protrusions. The cortex is the thin layer of the actin cytoskeleton that is attached to the cell membrane. For the purposes of our model, we consider the cortex to represent the combined membrane and cortex. Actin protrusions from the cortex are allowed to bind to ECM fibers. Upon binding, the cortex stiffens, which allows the cell to "pull" on the ECM by generating traction forces on the tip of the protrusion [18; 42]. Here we develop a model of this mechanism in 2D to gain insight into how ECM stiffness affects the ability of the cell to migrate before developing a computationally expensive 3D model.

We consider a 2D cross-section of a cell migrating through an ECM consisting of fibers immersed in fluid. The cell and nucleus are modeled as thin 1D elastic boundaries. The ECM consists of long thin fibers in 3D, and we model the cross section of one fiber as a regularized point force in our 2D model. For the cortex and nucleus, fiber elasticity gives the force density (in pN/μm^2) on a given configuration by

$$\boldsymbol{F}^{\mathrm{el}}_{n/c} = \frac{\partial}{\partial s}(T_{n/c}\boldsymbol{\tau}), \tag{19}$$

where n/c stands for the nucleus or cortex, s is the reference arclength variable, $\boldsymbol{\tau} = \boldsymbol{X}_s/\|\boldsymbol{X}_s\|$ is the unit tangent vector, and

$$T_{n/c} = k_{n/c}(\|\boldsymbol{X}_s\| - 1) \tag{20}$$

is the fiber tension. Here $k_{n/c}$ (pN/μm) represents the stiffness of the nucleus/cortex. At the beginning of our simulations, we choose the cortex to be relatively soft with $k_c = 1\,\mathrm{pN}/\mu\mathrm{m}$ and the nuclear boundary to be much stiffer, $k_n = 50\,\mathrm{pN}/\mu\mathrm{m}$ [16]. We take the diameter of the cortex to be 1 μm and the diameter of the "nucleus" to be 0.9 μm, with the latter taken to be large to model effective elasticity of the cytoplasm.

We also discretize the cortex with $N_c = 80$ points and the nuclear boundary with $N_m = 40$ points. Using this discretization, one can numerically approximate derivatives in (19) and (20) via centered differences to obtain a force density at each point in pN/μm^2, then multiply by the reference point spacing to obtain a force, $\widehat{\boldsymbol{F}}^{\mathrm{el}}_{n/c}$ in pN/μm at each point on the nucleus/cortex.

The ECM can be represented in a cross-sectional sense as an array of points in space with some characteristic length spacing. In this section, we keep the spacing constant and fix it to be on average the same as the diameter of the cell, so that the cell is not sterically hindered from passing through the ECM. Future work will focus on the effect of ECM density in a more rigorous context; our goal here is instead to show the effect of matrix stiffness at constant fiber density.

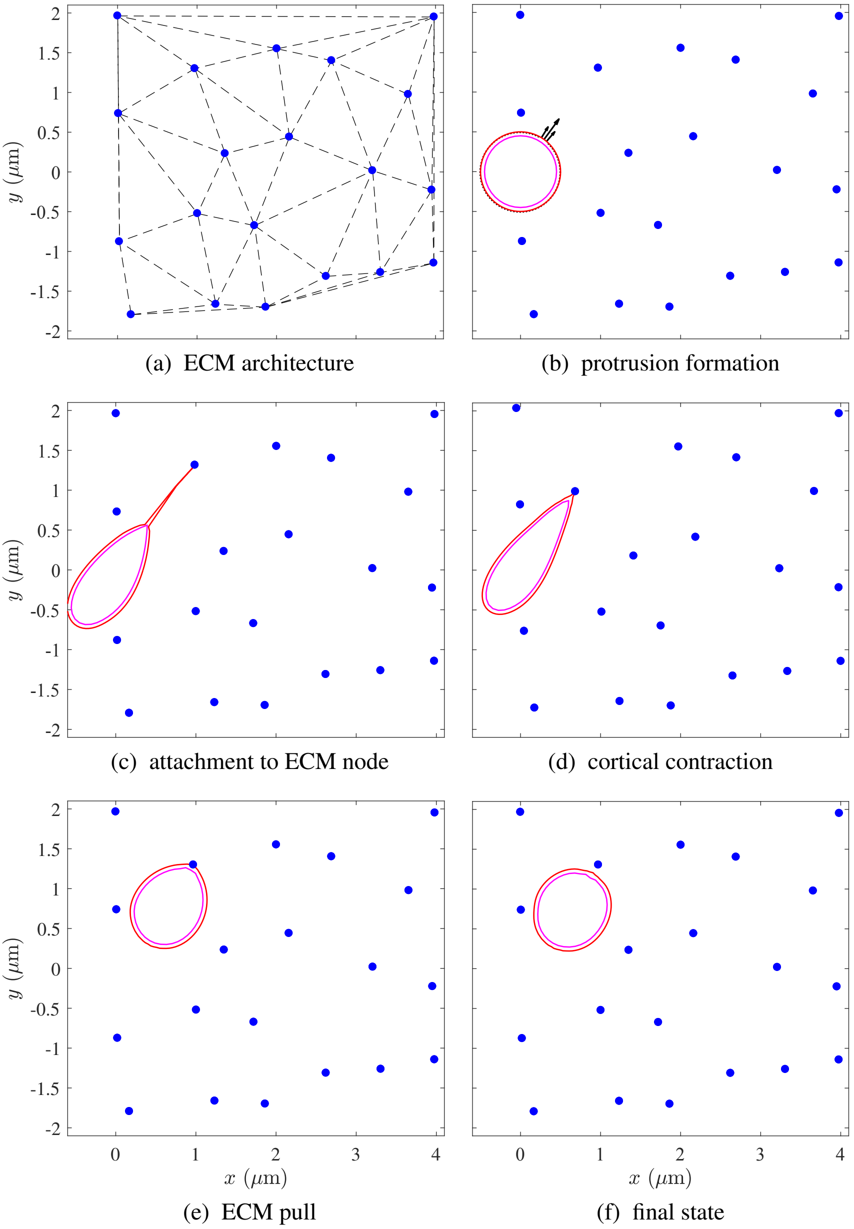
We therefore generate 20 ECM nodes that are approximately spaced by the cell diameter on a 4 × 4 box, shown as blue points in Figure 4a. We triangulate this set of points, with each edge representing a spring that connects two ECM nodes (dashed black lines in Figure 4a). Let k_{teth} (pN/μm^2) denote the stiffness of these springs. Then the time-dependent force (in pN/μm) on each ECM node is given by

$$\widehat{\boldsymbol{F}}^j_{\text{ECM}}(t) = -k_{\text{teth}}\Biggl(\boldsymbol{X}^j(t) - \boldsymbol{Z}^j + \sum_{i \in \mathcal{N}(j)} (\boldsymbol{X}^j(t) - \boldsymbol{X}^i(t))\Biggr). \tag{21}$$

Here $i \in \mathcal{N}(j)$ denotes a point i which is a neighbor of point j, in the sense that the nodes are connected by an edge in the Delaunay triangulation (black dotted lines in Figure 4a). We note also the presence of an anchoring (tether) node, $\boldsymbol{Z}^j$, whose purpose is to make sure the network stays in place dynamically. Without linking the nodes to reference nodes $\boldsymbol{Z}^j$, the force function in (21) would be translation-invariant, and the entire network of nodes would be free to slide away from the cell without penalty. We can compute $\boldsymbol{Z}^j$ for each node by setting the force at $t = 0$ in (21) to zero and solving for $\boldsymbol{Z}^j$. This is desired physically for the cell to migrate relative to the ECM. We have therefore determined the forces on the nucleus, cortex, and ECM that need to be computed at each time point and passed to (16).

We note that the use of points for the ECM fibers necessitates the use of a regularized method, as the velocity due to a point force is technically infinite at that point. We set $\epsilon = 0.075$ μm in the regularized equations, so that each point has an effective radius around it that is much smaller than the radius of the cell. We note that this value for ϵ is also the approximate spacing between the discrete nuclear and cortical points, which is one criterion for choosing ϵ [9; 10].

The overall simulation algorithm is as follows. Draw from a uniform distribution a point j on the front edge of the cortex (representing cell polarization) and suppose that actin polymerizes at that point. We apply a force density of strength $f_0 = 500\,\text{pN}/\mu\text{m}^2$ in the normal direction at point j and a force density of strength $f_0/2$ in the normal direction at points $j-1$ and $j+1$. Importantly, the cell physically cannot generate any net force on the fluid, so we spread the equal and opposite force over the other $N_c - 3$ cortex points. The force distribution on the cortex is shown in Figure 4b. We note that the only effect of f_0 is to set the timescale of migration, and we are concerned with the *relative* timescale across different ECM stiffnesses (so that the choice of f_0 is arbitrary). For this reason we also set $\mu = 1$ Pa s for simplicity.



(a) ECM architecture

(b) protrusion formation

(c) attachment to ECM node

(d) cortical contraction

(e) ECM pull

(f) final state

Figure 4. One cycle of a cell migrating via a finger protrusion mechanism through an ECM matrix of elastic nodes with $k_{\text{teth}} = 50\,\text{pN}/\mu\text{m}^2$. (a) The structure of the ECM, which has 20 nodes (blue points) that are linked together by springs (dashed black lines). (b–f) The dynamic process of cell migration. (b) A protrusion forms on the cell surface. (c) The protrusion binds to a node. (d) The cortical stiffness increases, pulling the node inward. (e) The dynamic balance between elasticity of the cell and ECM elasticity pulls the cell towards the ECM node's resting position. (f) The cell releases the node and is ready to form another protrusion.

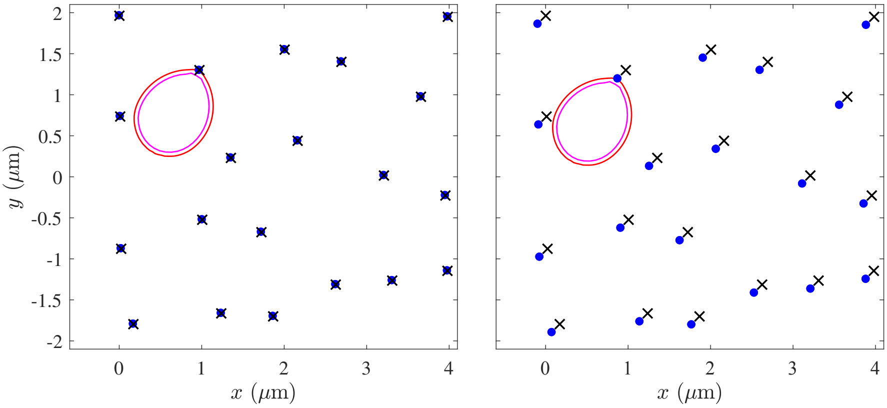


Figure 5. Final states for one cycle of the cell motility problem with $k_{\text{teth}} = 50\,\text{pN}/\mu\text{m}^2$ and (left) $R = 10^3$ and (right) sum of forces being zero via mean subtraction. Subtracting the mean of the force has created an artificial equilibrium state in the right, where each of the nodes has been displaced from its original position (black x's) by some constant amount. This does not occur in the left.

We then allow the cell protrusion to grow by evolving the system in time until the protrusion tip comes into contact with a node. By contact, we mean that the discrete points are a distance 2ϵ or less from each other, so that their "blob" functions are in contact. Once the discrete points come within a distance 2ϵ of each other, the protrusion tip binds to the node (shown in Figure 4c), and the cortex becomes stiffer by a factor of 100 to model the increased traction at the protrusion tips seen in [18]. The increased stiffness causes the cortex to rapidly become rounder. Since the cortex is attached to the node, it then pulls the node inward as shown in Figure 4d. As the node is pulled in, it generates a force in the opposite direction due to elasticity of the ECM (the node's resting configuration is its initial configuration in the ECM lattice). These forces balance dynamically, so that as the cortex becomes more round, the cortical force due to elasticity decreases, which in turn allows the force on the ECM to decrease, thereby pulling the entire cell and node back towards the initial position of the node. In the final state, Figure 4e, the cell is round and the node returns to a point close to its initial position. At this time, the node detaches from the cell by moving a distance 2ϵ away in the normal direction, as shown in Figure 4f, and the process can then repeat. We define this entire process as one cycle.

In this application, the anchor ECM nodes $\boldsymbol{Z}^j$ create a net force in the domain. As we observed in Section 4.1, handling this imbalance by subtracting the mean force at each node can create nonphysical translated equilibrium configurations. Figure 5 shows the final configuration when the system velocity has dropped below ϵ for a migrating cell in an ECM with $k_{\text{teth}} = 50\,\text{pN}/\mu\text{m}^2$ for (left) a system simulated using (16) and (right) a system simulated with zero net forcing via subtracting the mean force at each node. A shift in the entire domain in the negative horizontal and vertical directions is shown in Figure 5, right. Physically, we expect the nodes to return to their initial configuration in Figure 4a (marked with black x's in Figure 5).

The translation of the ECM structure in the case of subtracting the mean forces occurs because the pulling inward of the ECM node (shown in Figure 4d) creates a net force in the positive horizontal and vertical directions. Subtracting the mean force from each node does result in a net applied force of zero, but also results in an equilibrium configuration where the nodes have been shifted in the direction opposite the force imbalance. This situation is analogous to that of Section 4.1, where there was an artificial equilibrium state in the positive horizontal direction (and no relaxation to the equilibrium) resulting from a force imbalance in the negative horizontal direction. Such a shift may not be important for some applications if the *relative* position of the objects is desired. For our application, we are interested in the *absolute distance traveled by the cell*. For this reason, along with the ECM returning to its initial resting configuration for subsequent motility cycles, we conclude that it is better to use (16) to update the velocity rather than subtracting the mean forces.

We now use the model to simulate the distance traveled by the cell for different values of ECM stiffness k_{teth}. We calculate the total Euclidean distance traveled by the nucleus' center of mass as a function of time for the same 20-node ECM, but with varying stiffness $k_{\text{teth}} = 10, 25, 50\,\text{pN}/\mu\text{m}^2$. For comparison, we also simulate a rigid ECM by enforcing a $\boldsymbol{u} = 0$ boundary condition at each of the ECM nodes rather than the mean velocity condition on the large circle. We simulate up to a finite time, which corresponds to the time the cell has finished one cycle of migration (Figure 4e) in the rigid ECM case ($t = 1.36 \approx 1360\Delta t$). The time step is adaptive; generally it is taken to be $\Delta t = 0.001$, but it shrinks to $\Delta t = 2 \times 10^{-4}$ for a small time (0.05 s) beginning when the cortex binds to a node and stiffens to $k_c = 100\,\text{pN}/\mu\text{m}$. In Figure 6, we plot the Euclidean displacement in the direction of the ECM node over time for different values of stiffness k_{teth}. In all cases, the cell initially moves backwards slightly (as seen in Figure 4c) prior to contacting an ECM node. Once the ECM node is contacted and the cortex contracts, the distance traveled increases with time, with larger velocities for stiffer matrices. However, the data from all simulations appear to be approaching the same steady state value of displacement. For stiffer matrices, the node resists deformation by the cell (shown in Figure 4d), and the cell moves toward the node. In the rigid case, the node does not move, and the cell quickly contracts to form a circular configuration around the node. The conclusion of this preliminary study is therefore that *stiffer matrices* allow for faster cell velocities for finger-like protrusion mechanisms. We plan to study this problem in more detail by varying the matrix density and nuclear and cortical stiffness in future work.

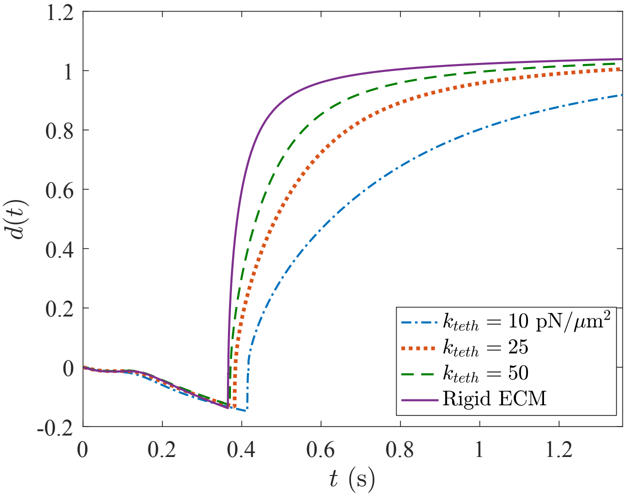


Figure 6. Euclidean distance traveled by the nucleus' center of mass versus time for different values of the ECM stiffness. Stiffer ECMs display faster velocities.

4.3. ***Cellular blebbing.*** Cellular blebs are spherical membrane protrusions that have been observed during cell migration [6]. A bleb forms when the cell cortex, normally attached to the cell membrane by linker proteins, detaches from the membrane. Cells that bleb are pressurized due to actomyosin contractility within the cortex. Once a bleb is initiated, a pressure driven flow drives the intracellular fluid (cytoplasm) that locally expands the membrane.

Bleb initiation has been modeled using different approaches, including solid mechanics [39], the immersed boundary method [31; 32], and boundary integral methods [14; 19; 20]. Results from several of these models have shown a bleb relieves only a small amount of intracellular pressure when the cytoplasm is modeled as a viscous fluid [31; 34]. Results from other models simulated with boundary integral methods show large pressure relief after bleb expansion [14; 20]. Our goal is to identify the source of this contradiction because maintaining high intracellular pressure is essential for cells to migrate using blebs [23].

Here we present a model of bleb expansion based on [20]. We treat both the membrane and cortex as one dimensional closed curves. The membrane and cortex parametrizations are represented by $\boldsymbol{X}^m(s)$ and $\boldsymbol{X}^c(s)$, respectively, where s is the arclength parameter. The most critical part of the model is the adhesion that connects the membrane and cortex. We model adhesion by an elastic spring connecting the membrane to the cortex with stiffness k_{adh}. The force density on the membrane due to adhesion is given by

$$\boldsymbol{F}_{\text{adh}}^{\text{mem/cor}}(s) = -k_{\text{adh}}(\boldsymbol{X}^m(s) - \boldsymbol{X}^c(s)), \tag{22}$$

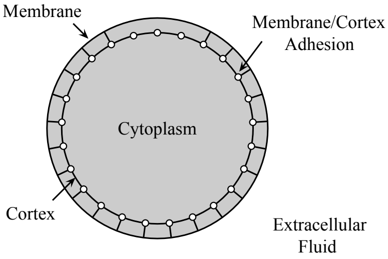


Figure 7. Components of the bleb model. The cytoplasm is modeled as a viscous fluid. A bleb is initiated by removing membrane-cortex adhesive links in a small region at the top of the cell.

with the force density on the cortex, $\boldsymbol{F}_{\text{adh}}^{\text{cor/mem}}(s)$, equal and opposite. Elastic forces on the membrane and cortex are due to surface tension and stretching and are computed by (19) with

$$T = \gamma_m + k_m(\|\boldsymbol{X}_s\| - 1) \tag{23}$$

with constants γ_m and k_m representing membrane surface tension and stiffness, respectively. The corresponding elastic parameters for the cortex are denoted by γ_c and k_c. The membrane satisfies a no-slip boundary condition, and its velocity is computed by (16). The velocity of the cortex is computed via a force balance, similar to [20],

$$\frac{d\boldsymbol{X}^c}{dt} = \frac{1}{\nu_c}(\boldsymbol{F}_{\text{el}}^{\text{cor}} + \boldsymbol{F}_{\text{adh}}^{\text{cor/mem}}), \tag{24}$$

where ν_c is the cortical viscosity. A bleb is initiated by removing the adhesive links in a small region of length approximately 5 μm at the top of the cell (see Figure 7).

This particular blebbing model is a physical example where the net force on the membrane fiber is nonzero. When the links between the top of the membrane and cortex are broken, there is a net vertical force on the membrane because part of the adhesive forces acting in the negative vertical direction are no longer present. Even though the cortex feels the equal and opposite forces, it moves independently of the fluid according to (24). The net hydrodynamic force is therefore equal to the net force on the membrane, and is nonzero. Without including the constant velocity from (15), the membrane would escape from the domain because of a spurious downward velocity resulting from the force imbalance in the positive vertical direction. The addition of $\boldsymbol{u}^R$ results in a well-posed problem so that we may solve for the membrane position in the blebbing model using the method of regularized Stokeslets.

Although previous studies have used the method of regularized Stokeslets to simulate cellular blebbing and migration in 2D [19; 20], the authors did not specify how they addressed the force imbalance in their models. We found our approach

symbol	quantity	value	source
r_{mem}	cell radius	$10\,\mu$m	[32; 34]
r_{cortex}	cortex radius	9.9 or 9.85 μm	[14]
γ_m	membrane surface tension	$40\,\text{pN}/\mu\text{m}$	[32]
k_m	membrane stiffness	$80\,\text{pN}/\mu\text{m}$	
γ_c	cortex surface tension	$250\,\text{pN}/\mu\text{m}$	[31]
k_c	cortical stiffness	$100\,\text{pN}/\mu\text{m}$	[31]
$k_{\text{adh}}^{\text{mem/cortex}}$	membrane/cortex adhesion stiffness coefficient	$247\,\text{pN}/\mu\text{m}^3$	
μ	cytosolic viscosity	5 Pa s	[31]
ν_c	cortical viscosity	$10\,\text{pN s}/\mu\text{m}^3$	[31]

Table 1. Parameters for the blebbing model.

to give nearly identical results to a model where the net zero force constraint is enforced by subtracting the mean of the calculated forces at each Stokeslet point.

We simulate the cellular blebbing process with the parameters in Table 1. The two different values of the cortex radius are used to test our hypothesis that a force imbalance on the *cortex* is what drives the pressure relief seen by previous authors [14; 19; 20]. For $k_{\text{adh}} = 247\,\text{pN}/\mu\text{m}^3$, the forces on the cortex (in the absence of a bleb) are exactly in balance when $r_{\text{cortex}} = 9.9\,\mu$m. When $r_{\text{cortex}} = 9.85\,\mu$m, there is initially a force imbalance on the cortex independent of bleb initiation.

We first equilibrate the model for ten time steps, then initiate a bleb at $t = 0$ by breaking the adhesion at the 7 (out of $N = 100$) points with largest y coordinate. Figure 8 shows the membrane shape over time for the two different values of the cortex radius, where time units are reported after bleb initiation. The bleb sizes and shapes are exactly the same. Despite this, the pressure dynamics of the two models are quite different. As shown in Figure 9, the pressure drops significantly when $r_{\text{cortex}} = 9.85$ but remains constant in the case $r_{\text{cortex}} = 9.90$. This is because of the force imbalance on the cortex in the former case. When the cortex's initial position is inwards of its resting position, it expands outward dynamically. This decreases the force on the membrane (and on the fluid) due to membrane-cortex adhesion (22), leading to a global pressure decrease inside the cell. Importantly, we observe that at $t \geq 1$ s, the cortex has reached its resting position and the two pressure profiles are the same (and are unchanged substantially with bleb expansion).

Models that include dynamic breaking of membrane-cortex adhesive links exhibit drastic changes in intracellular pressure [14; 20]. In such models, the dynamic breaking of adhesive links over time leads to pressure relief because the membrane force is updated suddenly without accounting for the corresponding force imbalance on the cortex. As the cortex slowly responds, it contracts inward in response to the loss of adhesive force, which promotes more link breakage and pressure changes.

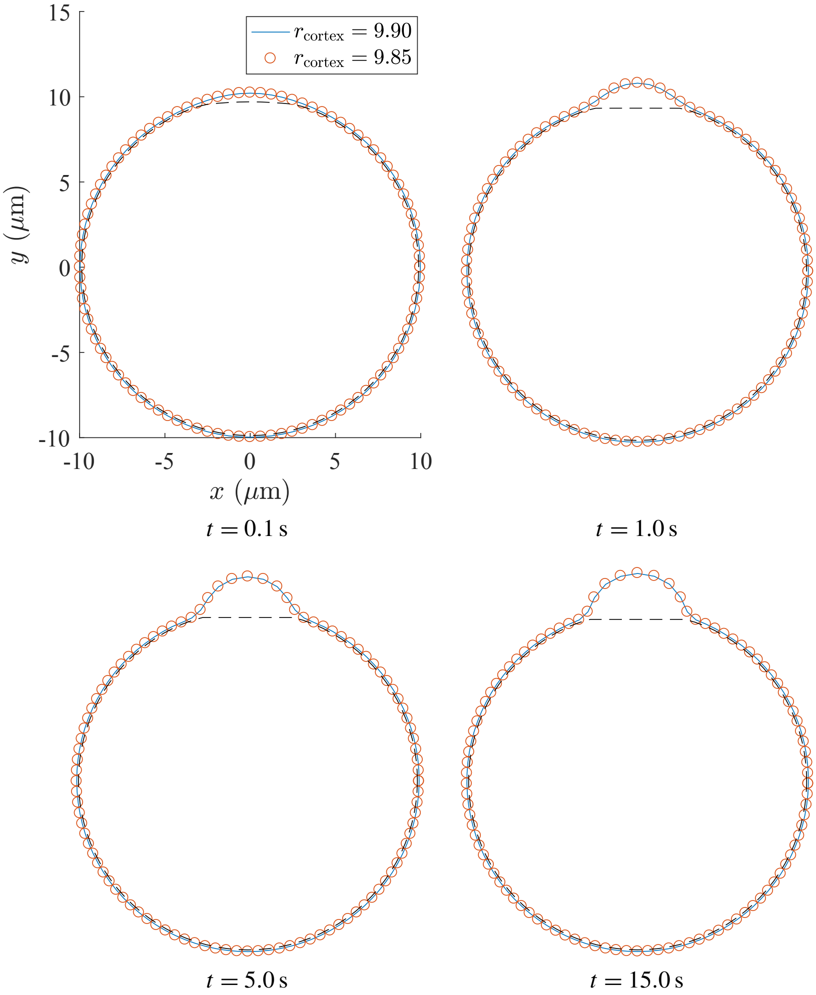


Figure 8. Membrane position for a blebbing cell with initial cortex radius 9.90 μm (blue line) or 9.85 μm (red circles). The position of the cortex is shown as a dashed black line and is in approximately the same position in both simulations. The positions are shown at several time values after bleb initiation.

The cortex itself is therefore never truly in equilibrium, and the force imbalance on it drives pressure changes.

The assumption that forces on the cortex are not equilibrated may be valid during highly dynamic processes such as during cell migration [41]. However, some experiments involve isolating a specific event, such as the expansion of a single bleb in [34]. In this work, experimental data show the cell achieves a quasisteady state behavior after bleb expansion, and the cortex is unlikely to be dynamically relieving pressure.

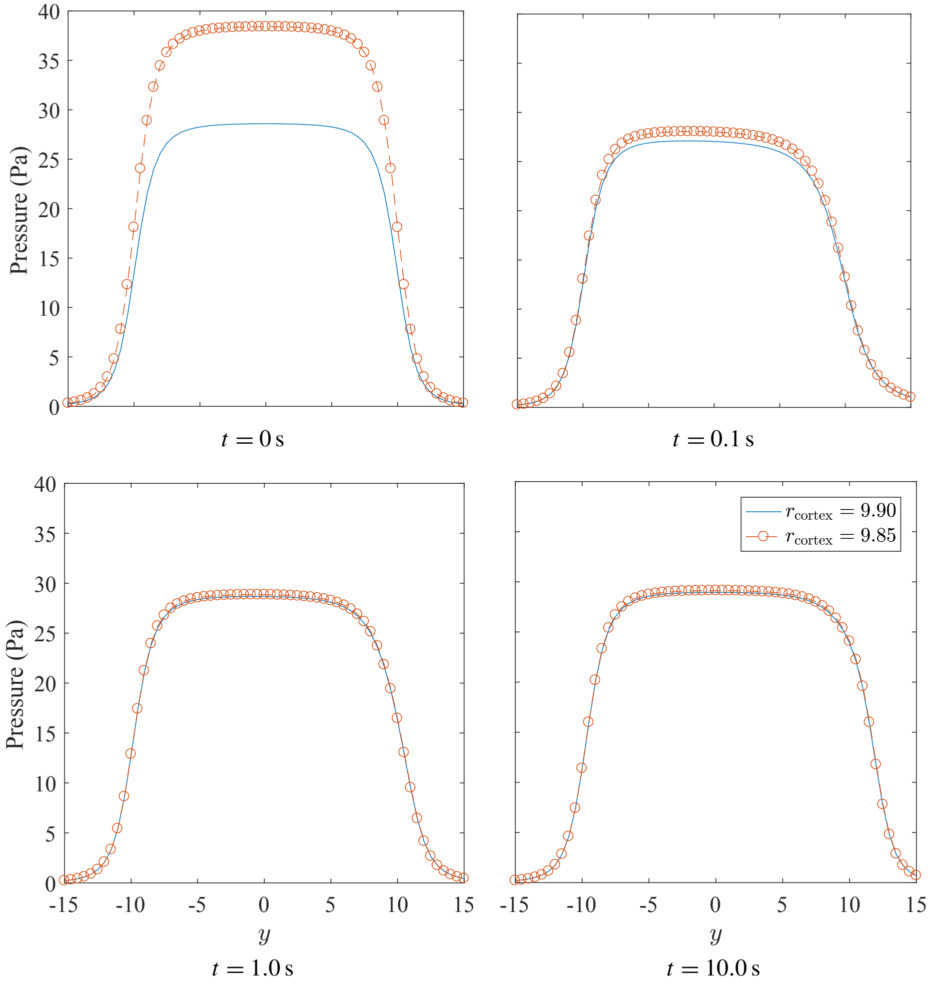


Figure 9. Pressure profile along the line $x = 0$ for the blebbing cell with initial cortex radius 9.90 μm (blue line) or 9.85 μm (red circles). Profiles are shown at (top left) $t = 0$ s, (top right) $t = 0.1$ s, (bottom left) $t = 1.0$ s, and (bottom right) $t = 10.0$ s. Note the large pressure relief when the forces are initially unbalanced on the cortex ($r_{\text{cortex}} = 9.85$).

5. Conclusion

When developing models for systems from biology, physics, and engineering that involve fluid-structure interaction, the simplification from 3D to 2D allows for model prototyping and fast simulations. In our applications, we seek to simulate cell motility and blebbing under a broad range of parameters, so fast simulations that are easy to visualize are critical for understanding model behavior. In zero Reynolds number flow, boundary integral methods are appealing because the velocity (and position) of immersed structures can be easily computed at the locations of interest rather than by interpolation after solving for the velocity on an Eulerian grid as in the IB method [25]. The condition of net zero force for 2D boundary integral methods can be a limiting factor during the development and simulation of mathematical models, especially those that include elastic tether-like forcing.

Here we present a numerical method to treat force constraints in 2D Stokes flow in an infinite domain. When the regularized or standard Stokeslet is used with a net nonzero hydrodynamic force in the flow domain, the velocity is unbounded at infinity (Stokes' paradox). For problems where no specific boundary conditions are imposed from the physics of the model system, the standard free space Stokeslet solution without modification fails because it is a Green's function for a system of equations that is not well-posed. The treatment of this problem must therefore involve solving a new, well-posed system with the appropriate Green's function.

One option is to require the net force to be zero within the domain by subtracting the mean force, thereby making the free space problem well-posed. This approach maintains the system time and length scales, but we show here that it can lead to falsely translated equilibrium states and unphysical dynamics. Alternatively, shifting the boundary conditions to create a well-posed system and locally valid solution is appealing (previously treated via solving a linear system [8; 9; 38]). We show here that by using a confined geometry and enforcing the condition of a mean zero velocity on the boundary, we can easily derive a new Green's function for a well-posed system of equations that gives the physically correct behavior. Introducing the new boundary has the effect of introducing a new length scale and a corresponding change in the Reynolds number, and we use this fact to derive an upper bound on the size of the boundary. We combine this with a lower bound that comes from the variation of the velocity on the boundary to obtain a unique choice of R.

We test this method by applying it to several model systems. In Section 4.1 we use a simple example of tethered points to illustrate how the ill-posedness of the free space Stokes equations can lead to nonphysical behavior of the regularized Stokeslet solution. In both this example and the model of a cell migrating through an ECM in Section 4.2, we show that subtracting the mean force can create translations in the structure configurations, which for our applications are problematic because we seek measurements of the cell displacement. Because of this, we find our solution of solving the Stokes equations with a zero mean flow on the boundary to give the most physically relevant results for our applications. Finally, we apply this technique to show how force imbalances on the permeable cell cortex drive pressure relief (independent of bleb formation) in blebbing cells.

We emphasize that this technique is not a solution to generally address Stokes' paradox. The problem remains that no flow is truly 2D, and so representing 3D flows in two dimensions introduces modeling error. However, there are ways to address models with nonzero net forcing so that insight can be gained from 2D models without having to take on the computational complexity of 3D. Here we describe a method to address Stokes' paradox and show that for our applications, the method gives solutions free of artificial translations. Additionally, the method is straightforward to implement and does not involve solving linear systems.

Future work involves extending the work of Section 4.2 to use our approach to investigate different mechanisms of cell migration (rear contraction in addition to frontal protrusion). We plan to examine the effectiveness of each mechanism for different values of ECM density, ECM stiffness, and cortical tension.

Acknowledgements

The authors thank Ricardo Cortez, Saverio Spagnolie, Charles Peskin, Aleksander Donev, and Yoichiro Mori for helpful discussions. We thank Alex Mogilner for discussions regarding the cell migration model. Ondrej Maxian is supported by the NSF Graduate Research Fellowship DGE-1342536 and the Henry MacCracken fellowship. This work was also supported by a grant from the Simons Foundation (#429808, Wanda Strychalski).

References

[1] J. Ainley, S. Durkin, R. Embid, P. Boindala, and R. Cortez, *The method of images for regularized Stokeslets*, J. Comput. Phys. **227** (2008), no. 9, 4600–4616. MR Zbl

[2] Y. Bao, J. Kaye, and C. S. Peskin, *A Gaussian-like immersed-boundary kernel with three continuous derivatives and improved translational invariance*, J. Comput. Phys. **316** (2016), 139–144. MR Zbl

[3] G. K. Batchelor, *An introduction to fluid dynamics*, Cambridge University, 1999. MR Zbl

[4] J. R. Blake, *A note on the image system for a Stokeslet in a no-slip boundary*, Math. Proc. Combridge **70** (1971), no. 2, 303–310. Zbl

[5] E. L. Bouzarth, A. T. Layton, and Y.-N. Young, *Modeling a semi-flexible filament in cellular Stokes flow using regularized Stokeslets*, Int. J. Numer. Methods Biomed. Eng. **27** (2011), no. 12, 2021–2034. MR Zbl

[6] G. Charras and E. Paluch, *Blebs lead the way: how to migrate without lamellipodia*, Nat. Rev. Mol. Cell Bio. **9** (2008), 730–736.

[7] L. H. Cisneros, R. Cortez, C. Dombrowski, R. E. Goldstein, and J. O. Kessler, *Fluid dynamics of self-propelled microorganisms, from individuals to concentrated populations*, Exp. Fluids **43** (2007), no. 5, 737–753.

[8] C. A. Copos and R. D. Guy, *A porous viscoelastic model for the cell cytoskeleton*, ANZIAM J. **59** (2018), no. 4, 472–498. MR Zbl

[9] R. Cortez, *The method of regularized Stokeslets*, SIAM J. Sci. Comput. **23** (2001), no. 4, 1204–1225. MR Zbl

[10] R. Cortez, L. Fauci, and A. Medovikov, *The method of regularized Stokeslets in three dimensions: analysis, validation, and application to helical swimming*, Phys. Fluids **17** (2005), no. 3, 031504. MR Zbl

[11] R. Cortez and M. Nicholas, *Slender body theory for Stokes flows with regularized forces*, Commun. Appl. Math. Comput. Sci. **7** (2012), no. 1, 33–62. MR Zbl

[12] R. Cortez and D. Varela, *A general system of images for regularized Stokeslets and other elements near a plane wall*, J. Comput. Phys. **285** (2015), 41–54. MR Zbl

[13] S. Even-Ram and K. M. Yamada, *Cell migration in 3D matrix*, Curr. Opin. Cell Biol. **17** (2005), no. 5, 524–532.

[14] C. Fang, T. H. Hui, X. Wei, X. Shao, and Y. Lin, *A combined experimental and theoretical investigation on cellular blebbing*, Sci. Rep. UK **7** (2017), 16666.

[15] L. J. Fauci and R. Dillon, *Biofluidmechanics of reproduction*, Annu. Rev. Fluid Mech. **38** (2006), 371–394. MR

[16] P. Friedl, K. Wolf, and J. Lammerding, *Nuclear mechanics during cell migration*, Curr. Opin. Cell Biol. **23** (2011), no. 1, 55–64.

[17] E. Lauga and T. R. Powers, *The hydrodynamics of swimming microorganisms*, Rep. Progr. Phys. **72** (2009), no. 9, 096601, 36. MR

[18] W. R. Legant, J. S. Miller, B. L. Blakely, D. M. Cohen, G. M. Genin, and C. S. Chen, *Measurement of mechanical tractions exerted by cells in three-dimensional matrices*, Nat. Methods **7** (2010), 969–971.

[19] F. Y. Lim, K.-H. Chiam, and L. Mahadevan, *The size, shape, and dynamics of cellular blebs*, Europhys. Lett. **100** (2012), no. 2, 28004.

[20] F. Y. Lim, Y. L. Koon, and K.-H. Chiam, *A computational model of amoeboid cell migration*, Comput. Method. Biomec. **16** (2013), no. 10, 1085–1095.

[21] G. Morra, *Insights on the physics of Stokes flow*, Pythonic geodynamics: implementations for fast computing, Springer, 2018, pp. 93–104. MR

[22] E. Nazockdast, A. Rahimian, D. Zorin, and M. Shelley, *A fast platform for simulating semi-flexible fiber suspensions applied to cell mechanics*, J. Comput. Phys. **329** (2017), 173–209. MR Zbl

[23] E. K. Paluch and E. Raz, *The role and regulation of blebs in cell migration*, Curr. Opin. Cell Biol. **25** (2013), no. 5, 582–590.

[24] C. S. Peskin, *Flow patterns around heart valves: a numerical method*, J. Comput. Phys. **10** (1972), no. 2, 252–271. Zbl

[25] ———, *The immersed boundary method*, Acta Numer. **11** (2002), 479–517. MR Zbl

[26] C. Pozrikidis, *Boundary integral and singularity methods for linearized viscous flow*, Cambridge University, 1992. MR Zbl

[27] ———, *Interfacial dynamics for Stokes flow*, J. Comput. Phys. **169** (2001), no. 2, 250–301. Zbl

[28] ———, *Numerical simulation of the flow-induced deformation of red blood cells*, Ann. Biomed. Eng. **31** (2003), no. 10, 1194–1205.

[29] ———, *Axisymmetric motion of a file of red blood cells through capillaries*, Phys. Fluids **17** (2005), no. 3, 031503. Zbl

[30] B. Rubinstein, K. Jacobson, and A. Mogilner, *Multiscale two-dimensional modeling of a motile simple-shaped cell*, Multiscale Model. Simul. **3** (2005), no. 2, 413–439. MR Zbl

[31] W. Strychalski and R. D. Guy, *A computational model of bleb formation*, Math. Med. Biol. **30** (2013), no. 2, 115–130. MR Zbl

[32] ———, *Intracellular pressure dynamics in blebbing cells*, Biophys. J. **110** (2016), no. 5, 1168–1179.

[33] J. M. Teran and C. S. Peskin, *Tether force constraints in Stokes flow by the immersed boundary method on a periodic domain*, SIAM J. Sci. Comput. **31** (2009), no. 5, 3404–3416. MR Zbl

[34] J.-Y. Tinevez, U. Schulze, G. Salbreux, J. Roensch, J.-F. Joanny, and E. Paluch, *Role of cortical tension in bleb growth*, P. Natl. Acad. Sci. USA **106** (2009), no. 44, 18581–18586.

[35] A.-K. Tornberg and M. J. Shelley, *Simulating the dynamics and interactions of flexible fibers in Stokes flows*, J. Comput. Phys. **196** (2004), no. 1, 8–40. MR Zbl

[36] M. Tozluoğlu, Y. Mao, P. A. Bates, and E. Sahai, *Cost–benefit analysis of the mechanisms that enable migrating cells to sustain motility upon changes in matrix environments*, J. Roy. Soc. Interface **12** (2015), no. 106, 20141355.

[37] M. Tozluoğlu, A. L. Tournier, R. P. Jenkins, S. Hooper, P. A. Bates, and E. Sahai, *Matrix geometry determines optimal cancer cell migration strategy and modulates response to interventions*, Nat. Cell Biol. **15** (2013), 751–762.

[38] B. Vanderlei, J. J. Feng, and L. Edelstein-Keshet, *A computational model of cell polarization and motility coupling mechanics and biochemistry*, Multiscale Model. Simul. **9** (2011), no. 4, 1420–1443. MR

[39] T. E. Woolley, E. A. Gaffney, J. M. Oliver, R. E. Baker, S. L. Waters, and A. Goriely, *Cellular blebs: pressure-driven, axisymmetric, membrane protrusions*, Biomech. Model. Mechan. **13** (2014), no. 2, 463–476.

[40] J. K. Wróbel, S. Lynch, A. Barrett, L. Fauci, and R. Cortez, *Enhanced flagellar swimming through a compliant viscoelastic network in Stokes flow*, J. Fluid Mech. **792** (2016), 775–797. MR Zbl

[41] A. K. Yip, K.-H. Chiam, and P. Matsudaira, *Traction stress analysis and modeling reveal that amoeboid migration in confined spaces is accompanied by expansive forces and requires the structural integrity of the membrane–cortex interactions*, Integr. Biol. **7** (2015), no. 10, 1196–1211.

[42] J. Zhu and A. Mogilner, *Comparison of cell migration mechanical strategies in three-dimensional matrices: a computational study*, Interface Focus **6** (2016), no. 5, 20160040.

Received November 26, 2018. Revised March 22, 2019.

ONDREJ MAXIAN: om759@cims.nyu.edu
Department of Mathematics, Courant Institute of Mathematical Sciences, New York University, New York, NY, United States

WANDA STRYCHALSKI: wis6@case.edu
Department of Mathematics, Applied Mathematics, and Statistics, Case Western Reserve University, Cleveland, OH, United States

COMM. APP. MATH. AND COMP. SCI.
Vol. 14, No. 2, 2019
dx.doi.org/10.2140/camcos.2019.14.175

POTENTIAL FIELD FORMULATION BASED ON DECOMPOSITION OF THE ELECTRIC FIELD FOR A NONLINEAR INDUCTION HARDENING MODEL

TONG KANG, RAN WANG AND HUAI ZHANG

In this paper we investigate a mathematical model of induction heating including eddy current equations coupled with a nonlinear heat equation. A nonlinear law between the magnetic field and the magnetic induction field in the workpiece is assumed. Meanwhile the electric conductivity is temperature dependent. We present a potential field formulation (the $\boldsymbol{A}$-ϕ method) based on decomposition of the electric field for the electromagnetic part. Using the theory of monotone operator and Rothe's method, we prove the existence of a weak solution to the coupled nonlinear system in the conducting domain. Finally, we solve it by means of the $\boldsymbol{A}$-ϕ finite element method and show some numerical simulation results.

1. Introduction

Electromagnetic induction as a method of heating electrically conducting materials is frequently used in industrial applications such as metal hardening and preheating for forging operations. The basic components of an induction heating system include an induction coil (called inductor), an alternating current power supply, and the workpiece itself. The inductor, which may take different shapes depending on the required heating pattern, is connected to the power supply. The flow of alternating current through the inductor generates an alternating magnetic field which in turn induces eddy currents in the workpiece that dissipate energy and bring about Joule heating. The magnitude of the eddy currents decreases with growing distance from the workpiece surface because of the frequency-dependent skin effect. Thus, induction heating is a suitable heat source for surface heat treatments if the current frequency has been chosen to be large enough. After the current has been

The research is supported by the National Key Research and Development Program of China (2016YFB0200801, 2018YFC0603500), the National Science Foundation of China (41725017, 41590864, 11571352, 11701529), and the Fundamental Research Funds for the Central Universities. Zhang is the corresponding author.

MSC2010: 35Q60, 35Q61.

Keywords: induction hardening, nonlinear eddy current equations, potential field formulation, solvability, Rothe's method.

switched off, the workpiece is quenched by spray-water cooling, which leads to the desired hardening effect.

The investigation of an induction heating system usually relies upon a series of expensive, long, and complicated experiments. Then the mathematical analysis and numerical simulation for induction heating play an important role in the designing process. The mathematical model of induction heating consists of Maxwell's equations coupled with a heat equation. Some papers present various numerical schemes for computation, e.g., [1; 5; 6; 9; 23]. But they omit complicated mathematical or numerical analyses of their models and numerical schemes. Other papers study the well-posedness of the problem and give theoretical results, e.g., [10; 19; 12; 13; 27; 28; 26]. All these works deal with Maxwell's equations with linear constitutive laws. Up to now there are few works considering both a nonlinear relationship between the magnetic field and the magnetic induction field, and dependence of electric conductivity on the temperature. Some authors [22] have studied a nonlinear magnetic field formulation for induction heating in the conducting domain and proved its solvability. In [9], the authors present the vector-scalar potential equations for a nonlinear setting including conducting and nonconducting parts. In this case the total current density is decomposed into summation of the external source and the induced part $-\sigma\,\partial_t \boldsymbol{A}$ caused by the magnetic induced field. This decomposition is also utilized in [12; 13]. The vector and scalar potentials belong to different Hilbert spaces and are solved numerically by using edge and nodal elements, respectively. We note that the mathematical analyses in [9] require the divergence-free property of the vector potential $\boldsymbol{A}$, but it brings about a contradiction with divergence-free $\sigma\,\partial_t \boldsymbol{A}$ since the conductivity σ is a temperature-dependent function. Therefore, we suggest a different potential field decomposition method for this coupled system.

The potential field method (called the $\boldsymbol{A}$-ϕ method) is to transform Maxwell's equations to vector-scalar potential formulations by decomposing the electric field into summation of a vector potential $\boldsymbol{A}$ and the gradient of a scalar potential ϕ, and to solve $\boldsymbol{A}$ and ϕ in the framework of the finite element method [2; 7; 8; 18; 16; 14; 15; 3; 25]. In order to guarantee the vector $\boldsymbol{A}$ is unique, we adopt the penalty function method by prescribing that $\boldsymbol{A}$ is divergence-free in addition to its curl, which is usually referred to as the Coulomb gauge. The $\boldsymbol{A}$-ϕ method can substantially reduce the "spurious solutions". Moreover, since the potential fields belong to $H^1(\Omega)^3$ or $H^1(\Omega)$, we can use nodal elements to solve the full coupled system. As we know, the $\boldsymbol{A}$-ϕ method has been widely used in electrical engineering. Its benefits have been demonstrated by practical applications. For example, it has attractive features including natural coupling to moment and boundary element methods and global energy conservation. Although introducing vector and scalar potentials increases the number of unknowns and equations, this seeming complication is justified by a better way of dealing with possible discontinuities of mediums. It can be applied to the case of any simply/multiply connected domain. We only solve the vector potential $\boldsymbol{A}$ in the nonconducting domain to find the magnetic induction field (or the magnetic field). Therefore, it is meaningful to introduce the $\boldsymbol{A}$-ϕ method to solve induction hardening problems.

The purpose of this paper is to study an induction hardening model with a nonlinear constitutional relation for the magnetic induction field by means of the $\boldsymbol{A}$-ϕ method. The paper is organized as follows. In the section below we give a nonlinear induction hardening model. In Section 3, we present some notations used in this paper and give the $\boldsymbol{A}$-ϕ variational formulation for this nonlinear coupled problem. In Section 4, we design a nonlinear time-discrete decoupled scheme and prove existence and uniqueness of its solution. Then we can solve the model by means of the finite element method. In Section 5, we investigate the coupled equations in the conducting domain. Some stability estimates for the approximate solution are derived. We discuss convergence of subsequences of the approximate solution in appropriate function spaces to a weak solution of the continuous problem. The last section is devoted to presenting numerical experiments by using our proposed $\boldsymbol{A}$-ϕ method.

2. Induction hardening model with a nonlinear law

We shall study a simplified induction hardening model (see Figure 1). Let Ω be a convex and bounded domain with boundary $\partial\Omega$, which consists of a workpiece occupied by ferromagnetic materials and an induction coil. There exits a flow of alternating current through the coil, which generates an alternating magnetic induction field which in turn induces eddy currents in the workpiece and bring about Joule heating. Since air is usually regarded as a thermal insulator and the conductor coil is supposed to have no resistance, we neglect the Joule heat in the coil and only consider the impact on the workpiece from the alternating magnetic induction. Denote the workpiece domain by Ω_c. The nonconducting region is presented by

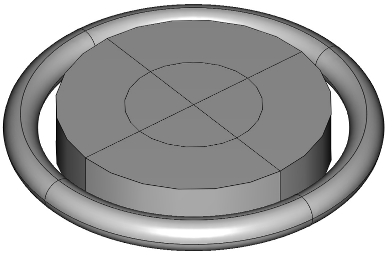


Figure 1. The disc workpiece and the induction coil (inductor).

$\Omega_e := \Omega \setminus \bar{\Omega}_c$. We consider the eddy current equations

$$\begin{cases} \partial_t \boldsymbol{B} + \nabla \times \boldsymbol{E} = 0, \\ \nabla \times \boldsymbol{H} = \sigma \boldsymbol{E} + \boldsymbol{J}_s, \end{cases} \tag{2-1}$$

where $\boldsymbol{E}$ and $\boldsymbol{H}$ stand for the electric and magnetic fields, respectively, $\boldsymbol{B}$ denotes the magnetic flux density, $\boldsymbol{J}_s$ is the source current density in the coil, and $\nabla \cdot \boldsymbol{J}_s = 0$. Then there exists a magnetic induction field $\boldsymbol{B}_s$ in $\mathbb{R}^3$ such that

$$\boldsymbol{J}_s = \nabla \times \frac{1}{\mu_0} \boldsymbol{B}_s.$$

The field $\boldsymbol{B}_s$ can be calculated directly by the Biot–Savart law

$$\boldsymbol{B}_s(\boldsymbol{x}, t) := \frac{\mu_0}{4\pi} \int_\Omega \frac{\boldsymbol{J}_s(\boldsymbol{y}, t) \times (\boldsymbol{x} - \boldsymbol{y})}{|\boldsymbol{x} - \boldsymbol{y}|^3} \, d\boldsymbol{V}, \tag{2-2}$$

where μ_0 is permeability of free space.

We present the nonlinear relation with a nonlinear law between $\boldsymbol{H}$ and $\boldsymbol{B}$ in the form

$$\boldsymbol{H} := \nu \boldsymbol{M}(\boldsymbol{B}) = \begin{cases} \nu m(|\boldsymbol{B}|)\boldsymbol{B} & \text{for a.e. } (\boldsymbol{x}, t) \in \Omega_c \times (0, T), \\ \nu_0 \boldsymbol{B} & \text{for a.e. } (\boldsymbol{x}, t) \in \Omega_e \times (0, T). \end{cases} \tag{2-3}$$

Here we introduce an inverse function $\nu(\boldsymbol{x})$ of magnetic permeability μ, which is strictly positive and bounded (i.e., $0 < \nu_* \le \nu \le \nu^* < \infty$). The function σ represents the electric conductivity and is defined as

$$\sigma := \begin{cases} \sigma(u(\boldsymbol{x}, t)) & \text{for a.e. } (\boldsymbol{x}, t) \in \Omega_c \times (0, T), \\ 0 & \text{for a.e. } (\boldsymbol{x}, t) \in \Omega_e \times (0, T), \end{cases} \tag{2-4}$$

where $u(\boldsymbol{x}, t)$ is a function of temperature in the workpiece. We consider σ to be bounded and strictly positive in $\bar{\Omega}_c$, i.e., there exist positive constants σ_* and σ^* such that

$$0 < \sigma_* \le \sigma(s) \le \sigma^* < \infty, \quad s > 0. \tag{2-5}$$

Let $\boldsymbol{n}$ stand for the outer normal vector associated with $\partial\Omega$ or $\partial\Omega_c$ (regarded as the boundary of Ω or Ω_c). The interface conditions between Ω_c and Ω_e are defined as

$$[\boldsymbol{H} \times \boldsymbol{n}] = \boldsymbol{0}, \quad [\boldsymbol{B} \cdot \boldsymbol{n}] = 0 \quad \text{for a.e. } (\boldsymbol{x}, t) \in \partial\Omega_c \times (0, T), \tag{2-6}$$

where the jump of any function f across $\partial\Omega_c$ is defined as $[f] := f_2|_{\partial\Omega_c} - f_1|_{\partial\Omega_c}$. The boundary and initial conditions are given by

$$\boldsymbol{B} \cdot \boldsymbol{n} = 0 \quad \text{for a.e. } (\boldsymbol{x}, t) \in \partial\Omega \times (0, T) \tag{2-7}$$

and

$$\boldsymbol{B}(0) = \boldsymbol{B}_s(\boldsymbol{x}, 0) := \boldsymbol{B}_s(0) \quad \text{for a.e. } \boldsymbol{x} \in \Omega, \, t = 0 \tag{2-8}$$

with $\nabla \cdot \boldsymbol{B}_s(0) = 0$.

Note that the local Joule heat generated by eddy currents in Ω_c equals

$$\sigma(u)\boldsymbol{E}\cdot\boldsymbol{E}=\sigma(u)|\boldsymbol{E}|^2.$$

Evolution of the temperature function u in Ω_c is given by the solution to the nonlinear heat equation with the initial and boundary conditions

$$\begin{cases}\partial_t\theta(u)-\nabla\cdot(\lambda\nabla u)=\sigma(u)|\boldsymbol{E}|^2 & \text{for a.e. } (\boldsymbol{x},t)\in\Omega_c\times(0,T),\\ \lambda\frac{\partial u}{\partial\boldsymbol{n}}=0 & \text{for a.e. } (\boldsymbol{x},t)\in\partial\Omega_c\times(0,T),\\ u(0)=u_0 & \text{for a.e. } \boldsymbol{x}\in\Omega_c,\ t=0,\end{cases}\tag{2-9}$$

where $\lambda(\boldsymbol{x},t)$ is a positive and bounded function $(0<\lambda_*\le\lambda\le\lambda^*<\infty)$. The real continuous function θ obeys

$$\theta(0)=0,\qquad 0<\theta_*\le\theta'(s),\qquad |\theta(s)|\le C(1+|s|).\tag{2-10}$$

The source term in the heat equation is unbounded and needs to be treated carefully for mathematical analyses of this coupled system, so we introduce the cut-off function [22]

$$\mathcal{R}_r(x):=\begin{cases} r & \text{if } x>r,\\ x & \text{if } |x|\le r,\\ -r & \text{if } x<-r,\end{cases}\tag{2-11}$$

where r is a positive constant. We truncate the right-hand side of (2-9) so it becomes

$$\begin{cases}\partial_t\theta(u)-\nabla\cdot(\lambda\nabla u)=\mathcal{R}_r(\sigma(u)|\boldsymbol{E}|^2) & \text{for a.e. } (\boldsymbol{x},t)\in\Omega_c\times(0,T),\\ \lambda\frac{\partial u}{\partial\boldsymbol{n}}=0 & \text{for a.e. } (\boldsymbol{x},t)\in\partial\Omega_c\times(0,T),\\ u(0)=u_0 & \text{for a.e. } \boldsymbol{x}\in\Omega_c,\ t=0.\end{cases}\tag{2-12}$$

The coupling between the electromagnetic equations and the heat equation is provided through the term $\sigma(u)$ in (2-4) and the Joule heating term in (2-12).

3. A-ϕ formulation for induction hardening

We start this section with definitions of some notations used throughout this paper. Let $L^2(\Omega)$ be the usual Hilbert space of square integrable functions equipped with the inner product and norm

$$(u,v)_\Omega:=\int_\Omega u(\boldsymbol{x})v(\boldsymbol{x})\,d\boldsymbol{x}\quad\text{and}\quad \|u\|_{L^2(\Omega)}:=(u,u)_\Omega^{1/2}.$$

Define $H^m(\Omega):=\{v\in L^2(\Omega):D^\xi v\in L^2(\Omega),\ |\xi|\le m\}$ which is equipped with the norm

$$\|u\|_{H^m(\Omega)}:=\biggl(\sum_{|\xi|\le m}\|D^\xi u\|^2_{L^2(\Omega)}\biggr)^{1/2},$$

where ξ represents a nonnegative triple index. We use boldface notation to represent vector-valued quantities, for example, $\boldsymbol{L}^2(\Omega) := (L^2(\Omega))^3$. The definitions for Ω are similarly defined for Ω_c.

Define the space $\widehat{\boldsymbol{H}}_0^1(\Omega) := \{\boldsymbol{v} \in \boldsymbol{H}^1(\Omega) : \boldsymbol{v} \times \boldsymbol{n}|_{\partial\Omega} = \boldsymbol{0}\}$. We further denote $\boldsymbol{V} := \widehat{\boldsymbol{H}}_0^1(\Omega) \times H^1(\Omega_c)/\mathbb{R}$ equipped with the inner product and norm

$$((\boldsymbol{P}, \varphi), (\boldsymbol{Q}, \psi))_{\boldsymbol{V}} := (\boldsymbol{P}, \boldsymbol{Q})_\Omega + (\nabla \boldsymbol{P}, \nabla \boldsymbol{Q})_\Omega + (\nabla\varphi, \nabla\psi)_{\Omega_c}$$

and

$$\|(\boldsymbol{Q}, \psi)\|_{\boldsymbol{V}} := \left(\|\boldsymbol{Q}\|^2_{\boldsymbol{H}^1(\Omega)} + \|\nabla\psi\|^2_{\boldsymbol{L}^2(\Omega_c)}\right)^{1/2}.$$

Since $\nabla \cdot \boldsymbol{B} = 0$, we can find a magnetic potential $\boldsymbol{A}$ such that $\boldsymbol{B} = \nabla \times \boldsymbol{A}$ and obtain $\boldsymbol{E} = -\partial_t \boldsymbol{A} - \partial_t \nabla\phi$, where ϕ is an arbitrary scalar function. The general physical decomposition of the electric field is $\boldsymbol{E} = -\partial_t \boldsymbol{A} - \nabla\phi$. Here we replace $\nabla\phi$ with $\partial_t\nabla\phi$ in order to keep a symmetric form in mathematical formulation. Meanwhile the penalty function term $-\nabla(\nu\nabla \cdot \boldsymbol{A})$ is added into the dominated $\boldsymbol{A}$-ϕ equation to ensure that $\boldsymbol{A}$ is divergence-free. Thus, the $\boldsymbol{A}$-ϕ formulation reads as

$$\begin{cases} \sigma(u(\boldsymbol{x}, t))\partial_t \boldsymbol{A} + \sigma(u(\boldsymbol{x}, t))\partial_t \nabla\phi & \\ \quad + \nabla \times \nu \boldsymbol{M}(\nabla \times \boldsymbol{A} - \boldsymbol{B}_s) - \nabla(\nu\nabla \cdot \boldsymbol{A}) = \boldsymbol{0} & \text{for a.e. } (\boldsymbol{x}, t) \in \Omega_c \times (0, T), \\ \nabla \cdot (\sigma(u(\boldsymbol{x}, t))\partial_t \boldsymbol{A} + \sigma(u(\boldsymbol{x}, t))\partial_t \nabla\phi) = 0 & \text{for a.e. } (\boldsymbol{x}, t) \in \Omega_c \times (0, T), \\ \nabla \times \nu \boldsymbol{M}(\nabla \times \boldsymbol{A} - \boldsymbol{B}_s) - \nabla(\nu\nabla \cdot \boldsymbol{A}) = \boldsymbol{0} & \text{for a.e. } (\boldsymbol{x}, t) \in \Omega_e \times (0, T), \end{cases} \tag{3-1}$$

with the interface conditions

$$\begin{cases} [A] = \boldsymbol{0}, \ [\nu\nabla \cdot \boldsymbol{A}] = 0 & \text{for a.e. } (\boldsymbol{x}, t) \in \partial\Omega_c \times (0, T), \\ [\nu \boldsymbol{M}(\nabla \times \boldsymbol{A} - \boldsymbol{B}_s) \times \boldsymbol{n}] = \boldsymbol{0} & \text{for a.e. } (\boldsymbol{x}, t) \in \partial\Omega_c \times (0, T), \\ (\sigma\partial_t \boldsymbol{A} + \sigma\partial_t \nabla\phi) \cdot \boldsymbol{n} = 0 & \text{for a.e. } (\boldsymbol{x}, t) \in \partial\Omega_c \times (0, T), \end{cases} \tag{3-2}$$

and the boundary conditions

$$\boldsymbol{A} \times \boldsymbol{n} = \boldsymbol{0}, \quad \nu\nabla \cdot \boldsymbol{A} = 0 \quad \text{for a.e. } (\boldsymbol{x}, t) \in \partial\Omega \times (0, T). \tag{3-3}$$

The initial value $\boldsymbol{A}_0$ is derived from $\boldsymbol{B}_s(0) \in \boldsymbol{L}^2(\Omega)$ by

$$\nabla \times \boldsymbol{A}_0 = \boldsymbol{B}_s(0), \quad \nabla \cdot \boldsymbol{A}_0 = 0 \quad \text{in } \Omega \text{ with } \boldsymbol{A}_0 \times \boldsymbol{n} = \boldsymbol{0} \text{ on } \partial\Omega,$$

and ϕ_0 and $\phi_0{}'$ are defined as zero in Ω_c.

Remark 3.1. Taking the divergence of both sides of the first and the third equations of (3-1) and taking into account the second equation, we obtain

$$\nabla^2(\nu\nabla \cdot \boldsymbol{A}) = 0 \quad \text{in } \Omega.$$

Considering the boundary condition $\nu\nabla \cdot \boldsymbol{A} = 0$, we have

$$\nu\nabla \cdot \boldsymbol{A} = 0 \quad \text{a.e. in } \Omega.$$

It is clear that the magnetic field $\boldsymbol{B}$ and the electric field $\boldsymbol{E}$ derived from the solutions $\boldsymbol{A}$ and ϕ satisfy problem (2-1) and conditions (2-6)–(2-8).

Next, let us give the variational formulation of the $\boldsymbol{A}$-ϕ coupled system. For problem (3-1), we multiply the first and the third equation by any test function $\boldsymbol{Q} \in \widehat{\boldsymbol{H}}_0^1(\Omega)$, integrate over Ω_c and Ω_e, respectively, apply Green's formula and the interface conditions, and sum the two equations to have

$$(\sigma(u)\partial_t\boldsymbol{A}+\sigma(u)\partial_t\nabla\phi,\, \boldsymbol{Q})_{\Omega_c}+(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s),\, \nabla\times\boldsymbol{Q})_\Omega+(\nu\nabla\cdot\boldsymbol{A},\, \nabla\cdot\boldsymbol{Q})_\Omega=0.$$

We also multiply the second equation of (3-1) by any $\psi \in H^1(\Omega_c)/\mathbb{R}$, integrate over Ω_c, and use Green's formula and the boundary condition on $\partial\Omega_c$ to obtain

$$(\sigma(u)\partial_t\boldsymbol{A}+\sigma(u)\partial_t\nabla\phi,\, \nabla\psi)_{\Omega_c}=0.$$

Thus, we can rewrite (3-1)–(3-3) in the variational form

$$(\sigma(u)\partial_t\boldsymbol{A}+\sigma(u)\partial_t\nabla\phi,\, \boldsymbol{Q}+\nabla\psi)_{\Omega_c}+(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s),\, \nabla\times\boldsymbol{Q})_\Omega + (\nu\nabla\cdot\boldsymbol{A},\, \nabla\cdot\boldsymbol{Q})_\Omega=0 \quad (3\text{-}4)$$

and, in a similar way, give the variational formulation of (2-12)

$$(\partial_t\theta(u),\, v)_{\Omega_c}+(\lambda\nabla u,\, \nabla v)_{\Omega_c}=(\mathcal{R}_r(\sigma(u)|\partial_t\boldsymbol{A}+\partial_t\nabla\phi|^2),\, v)_{\Omega_c}, \quad (3\text{-}5)$$

for any $(\boldsymbol{Q},\psi)\in\boldsymbol{V}$ and $v\in H^1(\Omega_c)$.

The vector field $\boldsymbol{M}$ is supposed to be potential, demicontinuous, and strongly monotone. The following results are borrowed from [22; 24]. The potential of $\boldsymbol{M}$ is denoted by $\Phi_{\boldsymbol{M}}$, i.e., grad $\Phi_{\boldsymbol{M}}=\boldsymbol{M}$. Throughout this paper we assume that

$$\begin{aligned}(\boldsymbol{M}(\boldsymbol{x})-\boldsymbol{M}(\boldsymbol{y}))\cdot(\boldsymbol{x}-\boldsymbol{y}) &\ge b_*|\boldsymbol{x}-\boldsymbol{y}|^2, \quad b_*>0, \quad \text{for all } \boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^3,\\ \boldsymbol{M}(\boldsymbol{x})-\boldsymbol{M}(\boldsymbol{y}) &\le C_M|\boldsymbol{x}-\boldsymbol{y}|, \quad C_M>0, \quad \text{for all } \boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^3, \quad (3\text{-}6)\\ \boldsymbol{M}(\boldsymbol{0}) &= \boldsymbol{0}.\end{aligned}$$

Following Theorem 5.1 in [24], we see that $\Phi_{\boldsymbol{M}}$ of the vector field $\boldsymbol{M}$ with (3-6) is strictly convex. Applying Theorem 8.4 in [24], we get

$$\boldsymbol{M}(\boldsymbol{x})\cdot(\boldsymbol{x}-\boldsymbol{y})\ge\Phi_{\boldsymbol{M}}(\boldsymbol{x})-\Phi_{\boldsymbol{M}}(\boldsymbol{y}) \quad \text{for all } \boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^3. \quad (3\text{-}7)$$

We also bound $\Phi_{\boldsymbol{M}}$ from below:

$$\begin{aligned}\Phi_{\boldsymbol{M}}(\boldsymbol{x}) &= \int_0^1 \boldsymbol{M}(\boldsymbol{x}t)\cdot\boldsymbol{x}\,dt = \int_0^1 \boldsymbol{M}(\boldsymbol{x}t)\cdot(\boldsymbol{x}t)t^{-1}\,dt\\ &\ge \int_0^1 b_*|\boldsymbol{x}t|^2t^{-1}\,dt \ge \frac{b_*}{2}|\boldsymbol{x}|^2, \quad (3\text{-}8)\end{aligned}$$

and

$$\Phi_{\boldsymbol{M}}(\boldsymbol{x})\le C\int_0^1(1+|\boldsymbol{x}t|)|\boldsymbol{x}|\,dt\le C\int_0^1|\boldsymbol{x}|^2t\,dt\le C|\boldsymbol{x}|^2. \quad (3\text{-}9)$$

By the chain rule we obtain that

$$\frac{d}{dt}\Phi_{\boldsymbol{M}^{-1}}(\boldsymbol{M}(\boldsymbol{x})) = \boldsymbol{M}^{-1}(\boldsymbol{M}(\boldsymbol{x}))\cdot\frac{d\boldsymbol{M}(\boldsymbol{x})}{dt} = \boldsymbol{x}\cdot\frac{d\boldsymbol{M}(\boldsymbol{x})}{dt}. \tag{3-10}$$

4. Time discretization

For (3-4)–(3-5), we present a nonlinear time-discrete approximation scheme based on the backward Euler scheme. Let n be a positive integer and $\{t_i = i\tau : i = 0, \dots, n\}$ be an equidistant partition of $[0, T]$ with $\tau = T/n$. Set for any function z

$$z_i = z(t_i), \qquad \delta z_i = \frac{z_i - z_{i-1}}{\tau}.$$

Using this notation we can approximate variational formulation (3-4): and (3-5)

$$\begin{aligned}(\sigma(u_{i-1})\delta\boldsymbol{A}_i + \sigma(u_{i-1})\delta\nabla\phi_i, \boldsymbol{Q}+\nabla\psi)_{\Omega_c} + (\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}_i - \boldsymbol{B}_{s,i}), \nabla\times\boldsymbol{Q})_\Omega&\\ + (\nu\nabla\cdot\boldsymbol{A}_i, \nabla\cdot\boldsymbol{Q})_\Omega = 0,&\qquad (4\text{-}1)\\ (\delta\theta(u_i), v)_{\Omega_c} + (\lambda_i\nabla u_i, v)_{\Omega_c} = (\mathcal{R}_r(\sigma(u_{i-1})|\delta\boldsymbol{A}_i + \delta\nabla\phi_i|^2), v)_{\Omega_c},&\qquad (4\text{-}2)\end{aligned}$$

for any $i = 1, \dots, n$, $(\boldsymbol{Q}, \psi) \in \boldsymbol{V}$, and $v \in H^1(\Omega_c)$.

Lemma 4.1 (coercivity). *If Ω is a convex and bounded domain or boundary $\partial\Omega$ is of class $C^{1,1}$, there exists a positive constant C such that*

$$C\big(\|\boldsymbol{Q}+\nabla\psi\|^2_{\boldsymbol{L}^2(\Omega_c)} + \|\nabla\times\boldsymbol{Q}\|^2_{\boldsymbol{L}^2(\Omega)} + \|\nabla\cdot\boldsymbol{Q}\|^2_{L^2(\Omega)}\big) \ge \|(\boldsymbol{Q},\psi)\|^2_{\boldsymbol{V}}$$

for any $(\boldsymbol{Q}, \psi) \in \boldsymbol{V}$.

Proof. Let $\boldsymbol{H}(\mathbf{curl}, \Omega)$ and $\boldsymbol{H}_0(\mathbf{curl}, \Omega) = \{\boldsymbol{v} \in \boldsymbol{H}(\mathbf{curl}, \Omega) : \boldsymbol{v}\times\boldsymbol{n}|_{\partial\Omega} = \boldsymbol{0}\}$ be standard Hilbert spaces. For any $\boldsymbol{Q} \in \widehat{\boldsymbol{H}}^1_0(\Omega) \subset \boldsymbol{H}_0(\mathbf{curl}, \Omega)$, by the embedding theorem in [4], there exists a constant C depending only on Ω such that

$$\|\boldsymbol{Q}\|^2_{\boldsymbol{H}(\mathbf{curl},\Omega)} \le C\big(\|\nabla\times\boldsymbol{Q}\|^2_{\boldsymbol{L}^2(\Omega)} + \|\nabla\cdot\boldsymbol{Q}\|^2_{L^2(\Omega)}\big).$$

Thus, we obtain

$$\begin{aligned}\|\boldsymbol{Q}\|^2_{\boldsymbol{L}^2(\Omega)} + \|\nabla\times\boldsymbol{Q}\|^2_{\boldsymbol{L}^2(\Omega)} + \|\nabla\cdot\boldsymbol{Q}\|^2_{L^2(\Omega)}&\\ + \|\boldsymbol{Q}\|^2_{\boldsymbol{L}^2(\Omega_c)} + \|\nabla\psi\|^2_{\boldsymbol{L}^2(\Omega_c)} - 2\int_{\Omega_c}|\boldsymbol{Q}|\cdot|\nabla\psi|\,d\boldsymbol{x}&\\ \le C\big(\|\nabla\times\boldsymbol{Q}\|^2_{\boldsymbol{L}^2(\Omega)} + \|\nabla\cdot\boldsymbol{Q}\|^2_{L^2(\Omega)} + \|\boldsymbol{Q}+\nabla\psi\|^2_{\boldsymbol{L}^2(\Omega_c)}\big).&\end{aligned}$$

Using inequalities $2ab \le \delta a^2 + b^2/\delta$ with $\delta = \frac{3}{2}$ and

$$\int_\Omega |\nabla\boldsymbol{Q}|^2\,d\boldsymbol{x} \le \|\nabla\times\boldsymbol{Q}\|^2_{\boldsymbol{L}^2(\Omega)} + \|\nabla\cdot\boldsymbol{Q}\|^2_{L^2(\Omega)},$$

we have

$$\|\boldsymbol{Q}\|^2_{\boldsymbol{L}^2(\Omega)} + \|\nabla \boldsymbol{Q}\|^2_{\boldsymbol{L}^2(\Omega)} + \|\nabla\psi\|^2_{\boldsymbol{L}^2(\Omega_c)} \\ \leq C\big(\|\nabla\times \boldsymbol{Q}\|^2_{\boldsymbol{L}^2(\Omega)} + \|\nabla\cdot \boldsymbol{Q}\|^2_{L^2(\Omega)} + \|\boldsymbol{Q}+\nabla\psi\|^2_{\boldsymbol{L}^2(\Omega_c)}\big),$$

which completes the proof. □

With the help of the theory of monotone operators [20; 24] and Lemma 4.1, we can prove existence of a weak solution on each time step.

Lemma 4.2. *Let* (2-10) *and* (3-6) *hold true. Moreover, assume that* $(\boldsymbol{A}_0, \phi_0) \in \boldsymbol{V}$, $u_0 \in L^2(\Omega_c)$, *and* $\boldsymbol{B}_s \in H^1((0,T); \boldsymbol{L}^2(\Omega))$. *Then there exist unique* $(\boldsymbol{A}_i, \phi_i) \in \boldsymbol{V}$ *and* $u_i \in H^1(\Omega_c)$ *solving* (4-1) *and* (4-2) *for any* $i = 1, \dots, n$.

Proof. Let us define the operators $\mathcal{L}_{\sigma,i} : \boldsymbol{V} \to \boldsymbol{V}^*$ and $\mathcal{G} : H^1(\Omega_c) \to H^{-1}(\Omega_c) = (H^1(\Omega_c))^*$:

$$\begin{aligned}\langle \mathcal{L}_{\sigma,i}(\boldsymbol{A},\phi),(\boldsymbol{Q},\psi)\rangle &:= \left(\sigma\frac{\boldsymbol{A}+\nabla\phi}{\tau}, \boldsymbol{Q}+\nabla\psi\right)_{\Omega_c} \\ &\qquad + \big(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_{s,i}) - \nu\boldsymbol{M}(-\boldsymbol{B}_{s,i}), \nabla\times\boldsymbol{Q}\big)_{\Omega} \\ &\qquad + (\nu\nabla\cdot\boldsymbol{A}, \nabla\cdot\boldsymbol{Q})_{\Omega}, \\ \langle\mathcal{G}(u), v\rangle &:= \left(\frac{\theta(u)}{\tau}, v\right)_{\Omega_c} + (\lambda\nabla u, \nabla v)_{\Omega_c}.\end{aligned}$$

Lemma 2 in [9] proves that the operator $\mathcal{G}$ is hemicontinuous, strictly monotone, and coercive. Similarly, the properties of the nonlinear operator $\mathcal{L}_{\sigma,i}$ are consequences of Lemma 4.1 and the properties of $\boldsymbol{M}$ which are shown in [9]. We omit the details of the proof here.

To obtain a unique solution $(\boldsymbol{A}_i, \phi_i)$ at a time step t_i, we have to solve the identity

$$\langle\mathcal{L}_{\sigma(u_{i-1}),i}(\boldsymbol{A}_i,\phi_i),(\boldsymbol{Q},\psi)\rangle \\ = \left(\sigma(u_{i-1})\frac{\boldsymbol{A}_{i-1}+\nabla\phi_{i-1}}{\tau}, \boldsymbol{Q}+\nabla\psi\right)_{\Omega_c} - (\nu\boldsymbol{M}(-\boldsymbol{B}_{s,i}), \nabla\times\boldsymbol{Q})_{\Omega}.$$

Since the right-hand side is known and the operator $\mathcal{L}_{\sigma(u_{i-1}),i}$ is hemicontinuous, strictly monotone, and coercive, we use Theorem 18.2 in [24] to prove the existence and uniqueness of the solution. Similarly we use the same theorem to acquire a unique solution $u_i \in H^1(\Omega_c)$ of the setting

$$\langle\mathcal{G}(u_i), v\rangle = \left(\frac{\theta(u_{i-1})}{\tau}, v\right)_{\Omega_c} + (\mathcal{R}_r(\sigma(u_{i-1})|\delta\boldsymbol{A}_i + \delta\nabla\phi_i|^2), v)_{\Omega_c}.$$

This provides us with the solution triple $\{\boldsymbol{A}_i, \phi_i, u_i\}$ at a time step $t = t_i$ for $i = 1, \dots, n$. □

5. Convergence

In this section, we shall discuss convergence of subsequences of the time-discrete approximate solution in appropriate function spaces to a weak solution of the nonlinear coupled system. Since the degenerate vector potential equation in Ω_e lacks a strong estimate of the time derivative, we don't prove convergence for problem (4-1)–(4-2) in both conducting and nonconducting domains. We only investigate the case of the workpiece coupling between nonlinear vector-scalar potential equations and a heat equation.

Coupled problem in conducting domain and stability estimate. Let us consider the following problem in the conducting domain:

$$\begin{cases} \sigma(u(\boldsymbol{x},t))\partial_t \boldsymbol{A} + \sigma(u(\boldsymbol{x},t))\partial_t \nabla\phi & \\ \quad + \nabla\times\nu \boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s) - \nabla(\nu\nabla\cdot\boldsymbol{A}) = \boldsymbol{0} & \text{for a.e. } (\boldsymbol{x},t)\in\Omega_c\times(0,T), \\ \nabla\cdot(\sigma(u(\boldsymbol{x},t))\partial_t\boldsymbol{A}+\sigma(u(\boldsymbol{x},t))\partial_t\nabla\phi) = 0 & \text{for a.e. } (\boldsymbol{x},t)\in\Omega_c\times(0,T), \\ \boldsymbol{A}\times\boldsymbol{n}=\boldsymbol{0},\ \nu\nabla\cdot\boldsymbol{A}=0 & \text{for a.e. } (\boldsymbol{x},t)\in\partial\Omega_c\times(0,T), \\ (\sigma\partial_t\boldsymbol{A}+\sigma\partial_t\nabla\phi)\cdot\boldsymbol{n}=0 & \text{for a.e. } (\boldsymbol{x},t)\in\partial\Omega_c\times(0,T). \end{cases} \tag{5-1}$$

In this section, we reassign $\boldsymbol{V} := \widehat{\boldsymbol{H}}_0^1(\Omega_c)\times H^1(\Omega_c)/\mathbb{R}$ equipped with the norm

$$\|(\boldsymbol{Q},\psi)\|_{\boldsymbol{V}} := \big(\|\boldsymbol{Q}\|^2_{\boldsymbol{H}^1(\Omega_c)} + \|\nabla\psi\|^2_{\boldsymbol{L}^2(\Omega_c)}\big)^{1/2}.$$

We rewrite (5-1) in the variational form

$$\begin{aligned} (\sigma(u)\partial_t\boldsymbol{A}+\sigma(u)\partial_t\nabla\phi,\ \boldsymbol{Q}+\nabla\psi)_{\Omega_c} + (\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s),\ \nabla\times\boldsymbol{Q})_{\Omega} \\ + (\nu\nabla\cdot\boldsymbol{A},\ \nabla\cdot\boldsymbol{Q})_{\Omega} = 0, \end{aligned} \tag{5-2}$$

$$(\partial_t\theta(u),v)_{\Omega_c} + (\lambda\nabla u,\nabla v)_{\Omega_c} = (\mathcal{R}_r(\sigma(u)|\partial_t\boldsymbol{A}+\partial_t\nabla\phi|^2),v)_{\Omega_c}, \tag{5-3}$$

and give the time-discrete formulation of (5-2)–(5-3)

$$\begin{aligned} (\sigma(u_{i-1})\delta\boldsymbol{A}_i+\sigma(u_{i-1})\delta\nabla\phi_i,\ \boldsymbol{Q}+\nabla\psi)_{\Omega_c} + (\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}_i-\boldsymbol{B}_{s,i}),\ \nabla\times\boldsymbol{Q})_{\Omega_c} \\ + (\nu\nabla\cdot\boldsymbol{A}_i,\ \nabla\cdot\boldsymbol{Q})_{\Omega_c} = 0, \end{aligned} \tag{5-4}$$

$$(\delta\theta(u_i),v)_{\Omega_c} + (\lambda_i\nabla u_i,v)_{\Omega_c} = (\mathcal{R}_r(\sigma(u_{i-1})|\delta\boldsymbol{A}_i+\delta\nabla\phi_i|^2),v)_{\Omega_c}, \tag{5-5}$$

for any $i=1,\dots,n$, $(\boldsymbol{Q},\psi)\in\boldsymbol{V}$, and $v\in H^1(\Omega_c)$.

Assume that the domain Ω_c is convex or boundary $\partial\Omega_c$ is of class $C^{1,1}$. Then we apply Lemma 4.1 and further prove that there exist unique $(\boldsymbol{A}_i,\phi_i)\in\boldsymbol{V}$ and $u_i\in H^1(\Omega_c)$ solving (5-4) and (5-5) for any $i=1,\dots,n$. The following lemmas show some basic stability estimates for $\{\boldsymbol{A}_i,\phi_i,u_i\}$.

Lemma 5.1. *Let* (2-10) *and* (3-6) *hold true. Moreover, assume that* $(\boldsymbol{A}_0,\phi_0)\in\boldsymbol{V}$, $u_0\in L^2(\Omega_c)$, $v\in H^1(\Omega_c)$, *and* $\boldsymbol{B}_s\in H^1((0,T);\boldsymbol{L}^2(\Omega_c))$. *There exists a positive constant C such that for* $1\le l\le n$,

$$\text{(i)}\quad \sum_{i=1}^{l}\|\delta\boldsymbol{A}_i+\delta\nabla\phi_i\|^2_{\boldsymbol{L}^2(\Omega_c)}\tau+\max_{1\le i\le l}\|\nabla\times\boldsymbol{A}_i-\boldsymbol{B}_{s,i}\|^2_{\boldsymbol{L}^2(\Omega_c)}+\max_{1\le i\le l}\|\nabla\cdot\boldsymbol{A}_i\|^2_{L^2(\Omega_c)}\le C,$$

$$\text{(ii)}\qquad \max_{1\le i\le l}\|\boldsymbol{A}_i+\nabla\phi_i\|^2_{\boldsymbol{L}^2(\Omega_c)}\le C,$$

$$\text{(iii)}\qquad \nabla\cdot\boldsymbol{A}_l=0\quad \textit{a.e. in } \Omega_c,$$

$$\nabla\times(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}_l-\boldsymbol{B}_{s,l}))\in\boldsymbol{L}^2(\Omega_c),\ \sum_{i=1}^{l}\|\nabla\times(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}_i-\boldsymbol{B}_{s,i}))\|^2_{\boldsymbol{L}^2(\Omega_c)}\tau\le C.$$

Proof. (i) Setting $(\boldsymbol{Q},\psi)=\tau(\delta\boldsymbol{A}_i,\delta\phi_i)$ in (5-4) and summing for $i=1,\dots,l$ yields

$$\begin{aligned}\sum_{i=1}^{l}(\sigma(u_{i-1})(\delta\boldsymbol{A}_i+\delta\nabla\phi_i),\delta\boldsymbol{A}_i+\delta\nabla\phi_i)_{\Omega_c}\tau+\sum_{i=1}^{l}(\nu\nabla\cdot\boldsymbol{A}_i,\nabla\cdot\boldsymbol{A}_i-\nabla\cdot\boldsymbol{A}_{i-1})_{\Omega_c}\\ +\sum_{i=1}^{l}(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}_i-\boldsymbol{B}_{s,i}),(\nabla\times\boldsymbol{A}_i-\boldsymbol{B}_{s,i})-(\nabla\times\boldsymbol{A}_{i-1}-\boldsymbol{B}_{s,i-1}))_{\Omega_c}\\ =-\sum_{i=1}^{l}(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}_i-\boldsymbol{B}_{s,i}),\boldsymbol{B}_{s,i}-\boldsymbol{B}_{s,i-1})_{\Omega_c}.\end{aligned}$$

For the first term we have

$$\sum_{i=1}^{l}(\sigma(u_{i-1})(\delta\boldsymbol{A}_i+\delta\nabla\phi_i),\delta\boldsymbol{A}_i+\delta\nabla\phi_i)_{\Omega_c}\tau\ge\sigma_*\sum_{i=1}^{l}\|\delta\boldsymbol{A}_i+\delta\nabla\phi_i\|^2_{\boldsymbol{L}^2(\Omega_c)}\tau.$$

For the second term we use (3-7)–(3-9) to deduce

$$\begin{aligned}\sum_{i=1}^{l}(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}_i-\boldsymbol{B}_{s,i}),(\nabla\times\boldsymbol{A}_i-\boldsymbol{B}_{s,i})-(\nabla\times\boldsymbol{A}_{i-1}-\boldsymbol{B}_{s,i-1}))_{\Omega_c}\\ \ge\sum_{i=1}^{l}\int_{\Omega_c}\nu(\Phi_{\boldsymbol{M}}(\nabla\times\boldsymbol{A}_i-\boldsymbol{B}_{s,i})-\Phi_{\boldsymbol{M}}(\nabla\times\boldsymbol{A}_{i-1}-\boldsymbol{B}_{s,i-1}))\,d\boldsymbol{x}\\ \ge\int_{\Omega_c}\nu\Phi_{\boldsymbol{M}}(\nabla\times\boldsymbol{A}_l-\boldsymbol{B}_{s,l})\,d\boldsymbol{x}-\int_{\Omega_c}\nu\Phi_{\boldsymbol{M}}(\nabla\times\boldsymbol{A}_0-\boldsymbol{B}_{s,0})\,d\boldsymbol{x}\\ \ge\frac{b_*\nu_*}{2}\|\nabla\times\boldsymbol{A}_l-\boldsymbol{B}_{s,l}\|^2_{\boldsymbol{L}^2(\Omega_c)}\ge C\|\nabla\times\boldsymbol{A}_l-\boldsymbol{B}_{s,l}\|^2_{\boldsymbol{L}^2(\Omega_c)}.\end{aligned}$$

Using $a(a-b)\ge a^2/2-b^2/2$ for any real numbers a,b yields

$$\sum_{i=1}^{l}(\nu\nabla\cdot\boldsymbol{A}_i,\nabla\cdot\boldsymbol{A}_i-\nabla\cdot\boldsymbol{A}_{i-1})_{\Omega_c}\ge\frac{\nu_*}{2}\|\nabla\cdot\boldsymbol{A}_l\|^2_{L^2(\Omega_c)}-\frac{\nu^*}{2}\|\nabla\cdot\boldsymbol{A}_0\|^2_{L^2(\Omega_c)}.$$

Applying Cauchy's and Young's inequalities and using demicontinuity of the vector field $\boldsymbol{M}$, we get

$$-\sum_{i=1}^{l}(\nu \boldsymbol{M}(\nabla \times \boldsymbol{A}_i - \boldsymbol{B}_{s,i}), \boldsymbol{B}_{s,i} - \boldsymbol{B}_{s,i-1})_{\Omega_c} \\ \leq C\sum_{i=1}^{l}\|\nabla \times \boldsymbol{A}_i - \boldsymbol{B}_{s,i}\|^2_{\boldsymbol{L}^2(\Omega_c)}\tau + C\int_0^T \left\| \frac{\partial \boldsymbol{B}_s}{\partial t}\right\|^2_{\boldsymbol{L}^2(\Omega_c)} dt.$$

Thus, we collect the above estimates and apply Grönwall's inequality to conclude the proof of (i).

(ii) From the result (i) and

$$\boldsymbol{A}_l + \nabla\phi_l = \boldsymbol{A}_0 + \nabla\phi_0 + \sum_{i=1}^{l}(\delta\boldsymbol{A}_i + \delta\nabla\phi_i)\tau,$$

we arrive at

$$\|\boldsymbol{A}_l + \nabla\phi_l\|_{\boldsymbol{L}^2(\Omega_c)} \leq C + \sum_{i=1}^{l}\|\delta\boldsymbol{A}_i + \delta\nabla\phi_i\|_{\boldsymbol{L}^2(\Omega_c)}\tau \\ \leq C + C\biggl(\sum_{i=1}^{l}\|\delta\boldsymbol{A}_i + \delta\nabla\phi_i\|^2_{\boldsymbol{L}^2(\Omega_c)}\tau\biggr)^{1/2} \leq C.$$

(iii) The equations of the strong solution of (5-4) are

$$\begin{cases} \sigma(u_{l-1})\delta\boldsymbol{A}_l + \sigma(u_{l-1})\delta\nabla\phi_l & \\ \quad + \nabla \times (\nu\boldsymbol{M}(\nabla \times \boldsymbol{A}_l - \boldsymbol{B}_{s,l})) - \nabla(\nu\nabla\cdot\boldsymbol{A}_l) = \boldsymbol{0} & \text{for a.e. } (\boldsymbol{x},t) \in \Omega_c \times (0,T), \\ \nabla\cdot(\sigma(u_{l-1})\delta\boldsymbol{A}_l + \sigma(u_{l-1})\delta\nabla\phi_l) = 0 & \text{for a.e. } (\boldsymbol{x},t) \in \Omega_c \times (0,T). \end{cases}$$

Taking the divergence of both sides of the first equation and using the second equation yields

$$\nabla^2(\nu\nabla\cdot\boldsymbol{A}_l) = 0 \quad \text{a.e. in } \Omega_c.$$

Considering $\nu\nabla\cdot\boldsymbol{A}_l = 0$ on $\partial\Omega$, we obtain

$$\nu\nabla\cdot\boldsymbol{A}_l = 0 \quad \text{a.e. in } \Omega_c.$$

Further, we have

$$\nabla \times (\nu\boldsymbol{M}(\nabla \times \boldsymbol{A}_l - \boldsymbol{B}_{s,l})) = -(\sigma(u_{l-1})\delta\boldsymbol{A}_l + \sigma(u_{l-1})\delta\nabla\phi_l) \quad \text{a.e. in } \Omega_c,$$

which leads to

$$\nabla \times (\nu\boldsymbol{M}(\nabla \times \boldsymbol{A}_l - \boldsymbol{B}_{s,l})) \in \boldsymbol{L}^2(\Omega_c)$$

and

$$\sum_{i=1}^{l}\|\nabla \times (\nu\boldsymbol{M}(\nabla \times \boldsymbol{A}_i - \boldsymbol{B}_{s,i}))\|^2_{\boldsymbol{L}^2(\Omega_c)}\tau \leq C. \qquad \square$$

Lemma 5.2. *Let the assumptions of Lemma 5.1 be fulfilled. Moreover, assume that $u_0 \in H^1(\Omega_c)$. Then there exists a positive constant C_r, which depends on parameter r of the cut-off function R_r, such that*

$$\text{(i)}\quad \sum_{i=1}^{n} \|\delta u_i\|^2_{L^2(\Omega_c)}\tau + \max_{1\le j\le n} \|\nabla u_j\|^2_{L^2(\Omega_c)} + \sum_{i=1}^{n} \|\nabla u_i - \nabla u_{i-1}\|^2_{L^2(\Omega_c)} \le C_r,$$

$$\text{(ii)}\qquad \max_{1\le j\le n} \|u_j\|^2_{L^2(\Omega_c)} \le C_r,$$

$$\text{(iii)}\qquad \max_{1\le j\le n} \|\delta\theta(u_j)\|^2_{H^{-1}(\Omega_c)} \le C_r.$$

The similar results can be found in [22] and are omitted here.

Convergence of approximate solutions. Next, we use Rothe's method [17] to prove a weak solution $(\boldsymbol{A}, \phi, u)$ to (5-2) and (5-3). In Proposition 5.4 we use monotonicity of the nonlinear vector field $\boldsymbol{M}$ and the Minty–Browder technique [11] to overcome the nonlinearity when passing to the limit.

We construct piecewise linear and piecewise constant in time functions

$$\begin{aligned} \bar{f}_n(0) &= f_n(0) = f_0, \\ \bar{f}_n(t) &= f_i && \text{for } t \in (t_{i-1}, t_i], \\ f_n(t) &= f_{i-1} + (t - t_{i-1})\delta f_i && \text{for } t \in (t_{i-1}, t_i]. \end{aligned}$$

Therefore, we can rewrite (5-4) and (5-5) in a continuous form for the whole time interval $[0, T]$ as

$$\begin{aligned} (\bar{\sigma}_n(t-\tau)(\partial_t \boldsymbol{A}_n + \partial_t \nabla\phi_n),\ \boldsymbol{Q} + \nabla\psi)_{\Omega_c} + (\nu \boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n - \bar{\boldsymbol{B}}_{s,n}), \nabla\times\boldsymbol{Q})_{\Omega_c} \\ + (\nu\nabla\cdot\bar{\boldsymbol{A}}_n, \nabla\cdot\boldsymbol{Q})_{\Omega_c} = 0, \end{aligned} \tag{5-6}$$

$$(\partial_t\theta_n, v)_{\Omega_c} + (\bar{\lambda}_n \nabla\bar{u}_n, v)_{\Omega_c} = (\mathcal{R}_r(\bar{\sigma}_n(t-\tau)|\partial_t \boldsymbol{A}_n + \partial_t\nabla\phi_n|^2), v)_{\Omega_c}, \tag{5-7}$$

for any $(\boldsymbol{Q}, \psi) \in \boldsymbol{V}$ and $v \in H^1(\Omega_c)$. Please note that for any $t \in (t_{i-1}, t_i]$

$$\delta\theta(u_i) = \partial_t\{\theta(u_{i-1}) + (t - t_{i-1})\delta\theta(u_i)\} = \partial_t\theta_n(t).$$

Now we give two convergence propositions. The sequences in the following part are actually subsequences still denoted by the same index n.

Proposition 5.3. *Let* (2-10) *hold true. Assume that $u_0 \in H^1(\Omega_c)$ and $\gamma(s)$ is a global Lipschitz continuous function. Then*:

$$\begin{aligned} \text{(i)}\qquad u_n \to u, \quad \bar{u}_n \to u \quad & \text{in } C([0,T]; L^2(\Omega_c)), \\ \bar{u}_n(t) \rightharpoonup u(t) \quad & \text{in } H^1(\Omega_c) \text{ for all } t \in [0,T], \\ \partial_t u_n \rightharpoonup \partial_t u \quad & \text{in } L^2((0,T); L^2(\Omega_c)). \end{aligned}$$

(ii) $$\bar{\sigma}_n \to \sigma(u), \quad \bar{\sigma}_n(t-\tau) \to \sigma(u) \quad \textit{in } L^2((0,T);\, L^2(\Omega_c)).$$

(iii) $$\bar{\theta}_n - \theta_n \to 0 \quad \textit{in } C([0,T];\, H^{-1}(\Omega_c)).$$

(iv) $$\bar{\theta}_n \to \theta(u) \quad \textit{in } L^2((0,T);\, L^2(\Omega_c)).$$

Using Lemma 1.3.13 in [17] and Lemma 5.2, we can prove Proposition 5.3. The similar proof can be found in [22] and is omitted here.

Proposition 5.4. *Let the assumptions of Proposition 5.3 be fulfilled, and let* (3-6) *be satisfied. Moreover, assume* $(\boldsymbol{A}_0, \phi_0) \in \boldsymbol{V}$, $\nu \in H^1(\Omega_c)$, $\boldsymbol{B}_s \in H^1((0,T);\, \boldsymbol{L}^2(\Omega_c))$, *and* $\partial_t \boldsymbol{B}_s$ *is Lipschitz continuous in time. Then*:

(i)
$$\begin{aligned}
\bar{\boldsymbol{A}}_n \rightharpoonup \boldsymbol{A}, \quad \nabla\bar{\phi}_n \rightharpoonup \nabla\phi \quad & \textit{in } L^2((0,T);\, \boldsymbol{L}^2(\Omega_c)).\\
\bar{\boldsymbol{A}}_n + \nabla\bar{\phi}_n \rightharpoonup \boldsymbol{A} + \nabla\phi \quad & \textit{in } L^2((0,T);\, \boldsymbol{L}^2(\Omega_c)),\\
\nabla \times \bar{\boldsymbol{A}}_n \rightharpoonup \nabla \times \boldsymbol{A} \quad & \textit{in } L^2((0,T);\, \boldsymbol{L}^2(\Omega_c)),\\
\nabla \cdot \boldsymbol{A} = 0 \quad & \textit{a.e. in } \Omega_c.
\end{aligned}$$

(ii)
$$\begin{aligned}
\boldsymbol{A}_n + \nabla\phi_n \to \boldsymbol{A} + \nabla\phi \quad & \textit{in } C([0,T];\, \boldsymbol{L}^2(\Omega_c)),\\
\partial_t \boldsymbol{A}_n + \partial_t \nabla\phi_n \rightharpoonup \partial_t \boldsymbol{A} + \partial_t \nabla\phi \quad & \textit{in } L^2((0,T);\, \boldsymbol{L}^2(\Omega_c)).\\
\bar{\boldsymbol{A}}_n + \nabla\bar{\phi}_n \to \boldsymbol{A} + \nabla\phi \quad & \textit{in } L^2((0,T);\, \boldsymbol{L}^2(\Omega_c)),
\end{aligned}$$

(iii) $$\bar{\boldsymbol{B}}_{s,n} \to \boldsymbol{B}_s \quad \textit{in } L^2((0,T);\, \boldsymbol{L}^2(\Omega_c)).$$

(iv) $$\boldsymbol{M}(\nabla \times \bar{\boldsymbol{A}}_n - \bar{\boldsymbol{B}}_{s,n}) \rightharpoonup \boldsymbol{M}(\nabla \times \boldsymbol{A} - \boldsymbol{B}_s) \quad \textit{in } L^2((0,T);\, \boldsymbol{L}^2(\Omega_c)).$$

(v)
$$\begin{aligned}
\nabla \times \bar{\boldsymbol{A}}_n - \bar{\boldsymbol{B}}_{s,n} \to \nabla \times \boldsymbol{A} - \boldsymbol{B}_s \quad & \textit{in } L^2((0,T);\, \boldsymbol{L}^2(\Omega_c)),\\
\boldsymbol{M}(\nabla \times \bar{\boldsymbol{A}}_n - \bar{\boldsymbol{B}}_{s,n}) \to \boldsymbol{M}(\nabla \times \boldsymbol{A} - \boldsymbol{B}_s) \quad & \textit{in } L^2((0,T);\, \boldsymbol{L}^2(\Omega_c)).
\end{aligned}$$

(vi) $$\partial_t \boldsymbol{A}_n + \partial_t \nabla\phi_n \to \partial_t \boldsymbol{A} + \partial_t \nabla\phi \quad \textit{in } L^2((0,T);\, \boldsymbol{L}^2(\Omega_c)).$$

Proof. (i) Lemmas 4.1 and 5.1 yield

$$\int_0^T \left(\|\bar{\boldsymbol{A}}_n\|^2_{\boldsymbol{H}^1(\Omega_c)} + \|\nabla\bar{\phi}_n\|^2_{L^2(\Omega_c)} \right) dt \le C.$$

Therefore, we conclude that

$$\begin{aligned}
\bar{\boldsymbol{A}}_n \rightharpoonup \boldsymbol{A} \quad & \text{in } L^2((0,T);\, \boldsymbol{H}^1(\Omega_c)),\\
\nabla\bar{\phi}_n \rightharpoonup \nabla\phi \quad & \text{in } L^2((0,T);\, \boldsymbol{L}^2(\Omega_c)),\\
\bar{\boldsymbol{A}}_n + \nabla\bar{\phi}_n \rightharpoonup \boldsymbol{A} + \nabla\phi \quad & \text{in } L^2((0,T);\, \boldsymbol{L}^2(\Omega_c)).
\end{aligned}$$

Take any $\boldsymbol{Q} \in \widehat{\boldsymbol{H}}^1_0(\Omega_c)$. Then

$$\begin{aligned}\lim_{n\to\infty}\int_0^T(\nabla\times\bar{\boldsymbol{A}}_n,\,\boldsymbol{Q})_{\Omega_c}\,dt &= \lim_{n\to\infty}\int_0^T(\bar{\boldsymbol{A}}_n,\nabla\times\boldsymbol{Q})_{\Omega_c}\,dt\\ &= \int_0^T(\boldsymbol{A},\nabla\times\boldsymbol{Q})_{\Omega_c}\,dt = \int_0^T(\nabla\times\boldsymbol{A},\,\boldsymbol{Q})_{\Omega_c}\,dt,\end{aligned}$$

which implies that $\nabla\times\bar{\boldsymbol{A}}_n \rightharpoonup \nabla\times\boldsymbol{A}$ in $L^2((0,T);\boldsymbol{L}^2(\Omega_c))$. Additionally we take any $q\in C_0^\infty(\Omega_c)$ and obtain

$$\begin{aligned}\lim_{n\to\infty}\int_0^T(\nabla\cdot\bar{\boldsymbol{A}}_n,q)_{\Omega_c}\,dt &= -\lim_{n\to\infty}\int_0^T(\bar{\boldsymbol{A}}_n,\nabla q)_{\Omega_c}\,dt\\ &= -\int_0^T(\boldsymbol{A},\nabla q)_{\Omega_c}\,dt = \int_0^T(\nabla\cdot\boldsymbol{A},q)_{\Omega_c}\,dt.\end{aligned}$$

From the density argument $\overline{C_0^\infty(\Omega_c)} = L^2(\Omega_c)$, we have that $\nabla\cdot\bar{\boldsymbol{A}}_n \rightharpoonup \nabla\cdot\boldsymbol{A}$ in $L^2((0,T);\boldsymbol{L}^2(\Omega_c))$. Thus, $\nabla\cdot\boldsymbol{A} = \nabla\cdot\bar{\boldsymbol{A}}_n = 0$ a.e. in Ω.

(ii) From Lemma 5.1 we get that

$$\int_0^T\|\partial_t\boldsymbol{A}_n+\partial_t\nabla\phi_n\|^2_{\boldsymbol{L}^2(\Omega_c)}\,dt\le C,\qquad \max_{t\in[0,T]}\|\bar{\boldsymbol{A}}_n+\nabla\bar{\phi}_n\|_{\boldsymbol{L}^2(\Omega_c)}\le C.$$

Employing Lemma 1.3.13 in [17] we get for a subsequence that

$$\begin{aligned}\boldsymbol{A}_n+\nabla\phi_n &\to \boldsymbol{A}+\nabla\phi && \text{in } C([0,T];\boldsymbol{L}^2(\Omega_c)),\\ \boldsymbol{A}_n+\nabla\phi_n &\rightharpoonup \boldsymbol{A}+\nabla\phi && \text{in } \boldsymbol{L}^2(\Omega_c) \text{ for all } t,\\ \bar{\boldsymbol{A}}_n+\nabla\bar{\phi}_n &\rightharpoonup \boldsymbol{A}+\nabla\phi && \text{in } \boldsymbol{L}^2(\Omega_c) \text{ for all } t,\\ \partial_t\boldsymbol{A}_n+\partial_t\nabla\phi_n &\rightharpoonup \partial_t\boldsymbol{A}+\partial_t\nabla\phi && \text{in } C([0,T];\boldsymbol{L}^2(\Omega_c)).\end{aligned}$$

Since

$$\begin{aligned}&\int_0^T\|(\bar{\boldsymbol{A}}_n+\nabla\bar{\phi}_n)-(\boldsymbol{A}+\nabla\phi)\|^2_{\boldsymbol{L}^2(\Omega_c)}\,dt\\ &\quad\le\int_0^T\|(\boldsymbol{A}_n+\nabla\phi_n)-(\boldsymbol{A}+\nabla\phi)\|^2_{\boldsymbol{L}^2(\Omega_c)}\,dt+\tau^2\int_0^T\|\partial_t\boldsymbol{A}_n+\partial_t\nabla\phi_n\|^2_{\boldsymbol{L}^2(\Omega_c)}\,dt,\end{aligned}$$

which approaches 0 as $n\to\infty$, we conclude that $\bar{\boldsymbol{A}}_n+\nabla\bar{\phi}_n\to\boldsymbol{A}+\nabla\phi$ in $L^2((0,T);\boldsymbol{L}^2(\Omega_c))$.

(iii) Thanks to $\boldsymbol{B}_s\in H^1((0,T);\boldsymbol{L}^2(\Omega_c))$, we have

$$\int_0^T\|\bar{\boldsymbol{B}}_{s,n}(t)-\boldsymbol{B}_s(t)\|^2_{\boldsymbol{L}^2(\Omega_c)}\,dt\le C\tau^2\xrightarrow{n\to\infty}0.$$

(iv) The sequence $\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n})$ is bounded in $L^2((0,T);\boldsymbol{L}^2(\Omega_c))$. Thus, there exists $\boldsymbol{p}$ from $L^2((0,T);\boldsymbol{L}^2(\Omega_c))$ such that $\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n})\rightharpoonup\boldsymbol{p}$ in that space (for a subsequence). Now we invoke the Minty–Browder technique. The general idea is based on the monotone character of $\boldsymbol{M}$. Let us investigate the inequality

$$\int_0^T (\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n})-\boldsymbol{M}(\boldsymbol{b}),\,\zeta\nu(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}-\boldsymbol{b}))_{\Omega_c}\,dt\ge 0. \tag{5-8}$$

We split this integral into four:

$$\begin{aligned}
P_1&=\int_0^T(\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\zeta\nu(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}))_{\Omega_c}\,dt,\\
P_2&=\int_0^T(\boldsymbol{M}(\boldsymbol{b}),\,\zeta\nu(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}))_{\Omega_c}\,dt,\\
P_3&=\int_0^T(\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\zeta\nu\boldsymbol{b})_{\Omega_c}\,dt,\\
P_4&=\int_0^T(\boldsymbol{M}(\boldsymbol{b}),\,\zeta\nu\boldsymbol{b})_{\Omega_c}\,dt.
\end{aligned}$$

This inequality holds true for any $\boldsymbol{b}\in L^2((0,T);\boldsymbol{L}^2(\Omega))$ and any nonnegative $\zeta\in\boldsymbol{C}_0^\infty(\Omega_c)$. We want to pass to the limit for $n\to\infty$ in (5-8). We do it for each term in (5-8) separately. We have

$$\begin{aligned}
P_1&=\int_0^T(\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\zeta\nu\nabla\times(\bar{\boldsymbol{A}}_n+\nabla\bar{\phi}_n-\boldsymbol{A}-\nabla\phi))_{\Omega_c}\,dt\\
&\qquad+\int_0^T(\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\zeta\nu\nabla\times\boldsymbol{A})_{\Omega_c}\,dt\\
&\qquad+\int_0^T(\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\zeta\nu(\boldsymbol{B}_s-\bar{\boldsymbol{B}}_{s,n}))_{\Omega_c}\,dt\\
&\qquad-\int_0^T(\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\zeta\nu\boldsymbol{B}_s)_{\Omega_c}\,dt\\
&=\int_0^T(\nabla\times[\zeta\nu\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n})],\,\bar{\boldsymbol{A}}_n+\nabla\bar{\phi}_n-\boldsymbol{A}-\nabla\phi)_{\Omega_c}\,dt\\
&\qquad+\int_0^T(\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\zeta\nu(\boldsymbol{B}_s-\bar{\boldsymbol{B}}_{s,n}))_{\Omega_c}\,dt\\
&\qquad+\int_0^T(\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\zeta\nu(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s))_{\Omega_c}\,dt\\
&=\int_0^T(\zeta\nabla\times[\nu\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n})],\,\bar{\boldsymbol{A}}_n+\nabla\bar{\phi}_n-\boldsymbol{A}-\nabla\phi)_{\Omega_c}\,dt\\
&\qquad+\int_0^T(\nabla\zeta\times[\nu\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n})],\,\bar{\boldsymbol{A}}_n+\nabla\bar{\phi}_n-\boldsymbol{A}-\nabla\phi)_{\Omega_c}\,dt\\
&\qquad+\int_0^T(\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\zeta\nu(\boldsymbol{B}_s-\bar{\boldsymbol{B}}_{s,n}))_{\Omega_c}\,dt\\
&\qquad+\int_0^T(\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\zeta\nu(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s))_{\Omega_c}\,dt.
\end{aligned}$$

Thus, from Lemma 5.1 and (3-6), we see that

$$\lim_{n\to\infty} P_1 = \int_0^T (\boldsymbol{p}, \zeta\nu(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s))_{\Omega_c}\, dt.$$

Passing to the limit for $n \to \infty$ in the remaining terms, we obtain

$$\begin{aligned}\lim_{n\to\infty} P_2 &= \int_0^T (\boldsymbol{M}(\boldsymbol{b}), \zeta\nu\nabla\times\boldsymbol{A})_{\Omega_c}\, dt,\\ \lim_{n\to\infty} P_3 &= \int_0^T (\boldsymbol{p}, \zeta\nu\boldsymbol{b})_{\Omega_c}\, dt,\\ \lim_{n\to\infty} P_4 &= \int_0^T (\boldsymbol{M}(\boldsymbol{b}), \zeta\nu\boldsymbol{b})_{\Omega_c}\, dt.\end{aligned}$$

Returning to (5-8), we see that

$$\begin{aligned}\lim_{n\to\infty}\int_0^T (\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n})-\boldsymbol{M}(\boldsymbol{b}), \zeta\nu(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}-\boldsymbol{b}))_{\Omega_c}\, dt&\\ = \int_0^T (\boldsymbol{p}-\boldsymbol{M}(\boldsymbol{b}), \zeta\nu(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s-\boldsymbol{b}))_{\Omega_c}\, dt&\\ \geq 0.&\end{aligned}$$

Since $\boldsymbol{b}$ has been chosen arbitrarily, we can set $\boldsymbol{b} = \nabla\times\boldsymbol{A}-\boldsymbol{B}_s+\varepsilon\boldsymbol{q}$, where $\varepsilon > 0$ and $\boldsymbol{q} \in L^2((0, T); \boldsymbol{L}^2(\Omega_c))$. Then we have

$$\int_0^T (\boldsymbol{p}-\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s+\varepsilon\boldsymbol{q}), \zeta\nu(-\varepsilon\boldsymbol{q}))_{\Omega_c}\, dt \geq 0.$$

Now passing ε to 0 yields

$$\int_0^T (\boldsymbol{p}-\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s), \zeta\nu(-\boldsymbol{q}))_{\Omega_c}\, dt \geq 0.$$

Similarly, $\boldsymbol{q}$ has been chosen arbitrarily, so we can set it to $\boldsymbol{q} = -\boldsymbol{q}$. Hence, the reverse inequality holds true. That implies

$$\int_0^T (\boldsymbol{p}-\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s), \zeta\nu\boldsymbol{q})_{\Omega_c}\, dt = 0.$$

This is true for any $\boldsymbol{q} \in L^2((0, T); \boldsymbol{L}^2(\Omega_c))$ and nonnegative $\zeta \in C_0^\infty(\Omega_c)$. Therefore, $\boldsymbol{p} = \boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)$ a.e. in $\Omega_c\times(0, T)$, i.e., $\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}) \rightharpoonup \boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)$ in $L^2((0, T); \boldsymbol{L}^2(\Omega_c))$.

(v) We shall show the strong convergence of $\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}) \to \boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)$ in $L^2((0, T); \boldsymbol{L}^2(\Omega_c))$. This can be achieved due to the strong monotonicity of the vector field $\boldsymbol{M}$ and the compensated compactness argument [21, Lemma 3.1].

Let $\zeta \in C_0^\infty(\overline{\Omega}_c)$ be nonnegative. We get in a similar way as in (iv)

$$\begin{aligned}0 &= \lim_{n\to\infty}\int_0^T \big(\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n})-\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s),\\ &\qquad\qquad \zeta\nu(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}-\nabla\times\boldsymbol{A}+\boldsymbol{B}_s)\big)_{\Omega_c}\,dt\\ &\ge \lim_{n\to\infty} c_M\int_0^T(\zeta\nu,|(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n})-(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)|)_{\Omega_c}\,dt\ge 0,\end{aligned}$$

which implies that

$$\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}\to\nabla\times\boldsymbol{A}-\boldsymbol{B}_s\quad\text{in } L^2((0,T);\boldsymbol{L}^2(\Omega_c)).$$

Further using (3-6) yields

$$\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n})\to\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)\quad\text{in } L^2((0,T);\boldsymbol{L}^2(\Omega_c)).$$

(vi) Take any $\eta\in[0,T]$ for which $\nabla\times\bar{\boldsymbol{A}}_n(\eta)\to\nabla\times\boldsymbol{A}(\eta)$ in $\boldsymbol{L}^2(\Omega_c)$. Here the set of such η is dense in $[0,T]$. Let us examine the inequality

$$\begin{aligned}0&\le\sigma_*\int_0^\eta\int_{\Omega_c}|\partial_t\boldsymbol{A}_n+\partial_t\nabla\phi_n-\partial_t\boldsymbol{A}-\partial_t\nabla\phi|^2\,d\boldsymbol{x}\,dt\\ &\le\int_0^\eta\int_{\Omega_c}\bar{\sigma}_n(t-\tau)|\partial_t\boldsymbol{A}_n+\partial_t\nabla\phi_n-\partial_t\boldsymbol{A}-\partial_t\nabla\phi|^2\,d\boldsymbol{x}\,dt\\ &=\int_0^\eta(\bar{\sigma}_n(t-\tau)(\partial_t\boldsymbol{A}_n+\partial_t\nabla\phi_n),\partial_t\boldsymbol{A}_n+\partial_t\nabla\phi_n)_{\Omega_c}\,dt\\ &\qquad+\int_0^\eta(\bar{\sigma}_n(t-\tau)(\partial_t\boldsymbol{A}+\partial_t\nabla\phi),\partial_t\boldsymbol{A}+\partial_t\nabla\phi)_{\Omega_c}\,dt\\ &\qquad-2\int_0^\eta(\bar{\sigma}_n(t-\tau)(\partial_t\boldsymbol{A}_n+\partial_t\nabla\phi_n),\partial_t\boldsymbol{A}+\partial_t\nabla\phi)_{\Omega_c}\,dt.\end{aligned}\tag{5-9}$$

In virtue of Proposition 5.3(ii) and the Lebesgue dominated theorem, we get

$$\begin{aligned}\lim_{n\to\infty}&\int_0^\eta(\bar{\sigma}_n(t-\tau)(\partial_t\boldsymbol{A}+\partial_t\nabla\phi),\partial_t\boldsymbol{A}+\partial_t\nabla\phi)_{\Omega_c}\,dt\\ &=\int_0^\eta(\sigma(u)(\partial_t\boldsymbol{A}+\partial_t\nabla\phi),\partial_t\boldsymbol{A}+\partial_t\nabla\phi)_{\Omega_c}\,dt.\end{aligned}\tag{5-10}$$

According to Propositions 5.3(ii) and 5.4(ii) and the Lebesgue dominated theorem, we deduce

$$\begin{aligned}\lim_{n\to\infty}&\int_0^\eta(\bar{\sigma}_n(t-\tau)(\partial_t\boldsymbol{A}_n+\partial_t\nabla\phi_n),\partial_t\boldsymbol{A}+\partial_t\nabla\phi)_{\Omega_c}\,dt\\ &=\int_0^\eta(\sigma(u)(\partial_t\boldsymbol{A}+\partial_t\nabla\phi),\partial_t\boldsymbol{A}+\partial_t\nabla\phi)_{\Omega_c}\,dt.\end{aligned}\tag{5-11}$$

Using (5-6) and (3-7), we obtain, for $\eta \in (t_{j-1}, t_j]$,

$$
\begin{aligned}
\int_0^{\eta} &(\bar{\sigma}_n(t-\tau)(\partial_t \boldsymbol{A}_n + \partial_t \nabla\phi_n),\, \partial_t \boldsymbol{A}_n + \partial_t \nabla\phi_n)_{\Omega_c}\, dt \\
&= -\sum_{i=1}^{j} (\nu \boldsymbol{M}(\nabla\times\boldsymbol{A}_i - \boldsymbol{B}_{s,i}),\, \nabla\times\boldsymbol{A}_i - \nabla\times\boldsymbol{A}_{i-1})_{\Omega_c} \\
&\qquad + \int_{t_j}^{\eta} (\nu \boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n - \bar{\boldsymbol{B}}_{s,n}),\, \nabla\times\partial_t \boldsymbol{A}_n)_{\Omega_c}\, dt \\
&= -\sum_{i=1}^{j} (\nu \boldsymbol{M}(\nabla\times\boldsymbol{A}_i - \boldsymbol{B}_{s,i}),\, \nabla\times\boldsymbol{A}_i - \boldsymbol{B}_{s,i} - \nabla\times\boldsymbol{A}_{i-1} + \boldsymbol{B}_{s,i-1})_{\Omega_c} \\
&\qquad - \sum_{i=1}^{j} (\nu \boldsymbol{M}(\nabla\times\boldsymbol{A}_i - \boldsymbol{B}_{s,i}),\, \boldsymbol{B}_{s,i} - \boldsymbol{B}_{s,i-1})_{\Omega_c} \\
&\qquad + \int_{t_j}^{\eta} (\nu \boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n - \bar{\boldsymbol{B}}_{s,n}),\, \nabla\times\partial_t \boldsymbol{A}_n)_{\Omega_c}\, dt \\
&\le \sum_{i=1}^{j} \int_{\Omega_c} \{\Phi_{\boldsymbol{M}}(\nabla\times\boldsymbol{A}_i - \boldsymbol{B}_{s,i}) - \Phi_{\boldsymbol{M}}(\nabla\times\boldsymbol{A}_{i-1} - \boldsymbol{B}_{s,i-1})\}\, d\boldsymbol{x} \\
&\qquad - \sum_{i=1}^{j} \Big(\nu \boldsymbol{M}(\nabla\times\boldsymbol{A}_i - \boldsymbol{B}_{s,i}) - \frac{\nu}{\tau}\int_{t_{i-1}}^{t_i} \boldsymbol{M}(\nabla\times\boldsymbol{A} - \boldsymbol{B}_s)\, dt,\, \boldsymbol{B}_{s,i} - \boldsymbol{B}_{s,i-1}\Big)_{\Omega_c} \\
&\qquad + \sum_{i=1}^{j} \Big(\frac{\nu}{\tau}\int_{t_{i-1}}^{t_i} \boldsymbol{M}(\nabla\times\boldsymbol{A} - \boldsymbol{B}_s)\, dt,\, \boldsymbol{B}_{s,i} - \boldsymbol{B}_{s,i-1}\Big)_{\Omega_c} \\
&\qquad + \int_{t_j}^{\eta} (\nu \boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n - \bar{\boldsymbol{B}}_{s,n}),\, \nabla\times\partial_t \boldsymbol{A}_n)_{\Omega_c}\, dt \\
&= \int_{\Omega_c} \nu \Phi_{\boldsymbol{M}}(\nabla\times\boldsymbol{A}_j - \boldsymbol{B}_{s,j})\, d\boldsymbol{x} - \int_{\Omega_c} \nu \Phi_{\boldsymbol{M}}(\nabla\times\boldsymbol{A}_0 - \boldsymbol{B}_{s,0})\, d\boldsymbol{x} \\
&\qquad - \sum_{i=1}^{j} \Big(\nu \boldsymbol{M}(\nabla\times\boldsymbol{A}_i - \boldsymbol{B}_{s,i}) - \frac{\nu}{\tau}\int_{t_{i-1}}^{t_i} \boldsymbol{M}(\nabla\times\boldsymbol{A} - \boldsymbol{B}_s)\, dt,\, \boldsymbol{B}_{s,i} - \boldsymbol{B}_{s,i-1}\Big)_{\Omega_c} \\
&\qquad + \sum_{i=1}^{j} \Big(\frac{\nu}{\tau}\int_{t_{i-1}}^{t_i} \boldsymbol{M}(\nabla\times\boldsymbol{A} - \boldsymbol{B}_s)\, dt,\, \boldsymbol{B}_{s,i} - \boldsymbol{B}_{s,i-1}\Big)_{\Omega_c} \\
&\qquad - \int_0^{t_j} (\nu \boldsymbol{M}(\nabla\times\boldsymbol{A} - \boldsymbol{B}_s),\, \partial_t \boldsymbol{B}_s)_{\Omega_c}\, dt \\
&\qquad + \int_0^{\eta} (\nu \boldsymbol{M}(\nabla\times\boldsymbol{A} - \boldsymbol{B}_s),\, \partial_t \boldsymbol{B}_s)_{\Omega_c}\, dt - \int_{t_j}^{\eta} (\nu \boldsymbol{M}(\nabla\times\boldsymbol{A} - \boldsymbol{B}_s),\, \partial_t \boldsymbol{B}_s)_{\Omega_c}\, dt \\
&\qquad + \int_{t_j}^{\eta} (\nu \boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n - \bar{\boldsymbol{B}}_{s,n}),\, \nabla\times\partial_t \boldsymbol{A}_n)_{\Omega_c}\, dt. \qquad (5\text{-}12)
\end{aligned}
$$

Applying (3-6) and (3-7) yields

$$\begin{aligned}\int_{\Omega_c}\nu\Phi_{\boldsymbol{M}}(\nabla\times\boldsymbol{A}_j-\boldsymbol{B}_{s,j})-\int_{\Omega_c}\nu\Phi_{\boldsymbol{M}}(\nabla\times\boldsymbol{A}(\eta)-\boldsymbol{B}_s(\eta))\,d\boldsymbol{x}&\\ \le\int_{\Omega_c}\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}_j-\boldsymbol{B}_{s,j})(\nabla\times\boldsymbol{A}_j-\boldsymbol{B}_{s,j}-\nabla\times\boldsymbol{A}(\eta)+\boldsymbol{B}_s(\eta))\,d\boldsymbol{x}&\\ \le c_M\nu^*\int_{\Omega_c}|\nabla\times\boldsymbol{A}_j-\boldsymbol{B}_{s,j}|\cdot|\nabla\times\boldsymbol{A}_j-\boldsymbol{B}_{s,j}-\nabla\times\boldsymbol{A}(\eta)+\boldsymbol{B}_s(\eta)|\,d\boldsymbol{x}&\\ \le C\|(\nabla\times\boldsymbol{A}_j-\boldsymbol{B}_{s,j})-(\nabla\times\boldsymbol{A}(\eta)-\boldsymbol{B}_s(\eta))\|_{\boldsymbol{L}^2(\Omega_c)}\xrightarrow{n\to\infty}0.&\qquad(5\text{-}13)\end{aligned}$$

Using (v) and $\boldsymbol{B}_s\in H^1((0,T);\boldsymbol{L}^2(\Omega_c))$, we arrive at

$$\begin{aligned}&\sum_{i=1}^{j}\left(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}_i-\boldsymbol{B}_{s,i})-\frac{\nu}{\tau}\int_{t_{i-1}}^{t_i}\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)\,dt,\,\boldsymbol{B}_{s,i}-\boldsymbol{B}_{s,i-1}\right)_{\Omega_c}\\ &\le c_M\frac{\nu^*}{\sqrt{\tau}}\sum_{i=1}^{j}\int_{t_{i-1}}^{t_i}\|(\nabla\times\boldsymbol{A}_i-\boldsymbol{B}_{s,i})-(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)\|_{\boldsymbol{L}^2(\Omega_c)}\,dt\cdot\|\boldsymbol{B}_{s,i}-\boldsymbol{B}_{s,i-1}\|_{\boldsymbol{L}^2(\Omega_c)}\\ &\le\frac{C}{\sqrt{\tau}}\left(\sum_{i=1}^{j}\tau\int_{t_{i-1}}^{t_i}\|(\nabla\times\boldsymbol{A}_i-\boldsymbol{B}_{s,i})-(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)\|^2_{\boldsymbol{L}^2(\Omega_c)}\,dt\right)^{1/2}\\ &\qquad\qquad\cdot\left(\sum_{i=1}^{j}\|\boldsymbol{B}_{s,i}-\boldsymbol{B}_{s,i-1}\|^2_{\boldsymbol{L}^2(\Omega_c)}\right)^{1/2}\\ &\le C\left(\int_0^T\|(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n})-(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)\|^2_{\boldsymbol{L}^2(\Omega_c)}\,dt\right)^{1/2}\left(\tau\int_0^T\left\|\frac{\partial\boldsymbol{B}_s}{\partial t}\right\|^2_{\boldsymbol{L}^2(\Omega_c)}dt\right)^{1/2}\\ &\xrightarrow{n\to\infty}0.\qquad(5\text{-}14)\end{aligned}$$

Thanks to the Lipschitz continuity of $\partial_t\boldsymbol{B}_s$, we have $\|\partial_t\boldsymbol{B}_s(t_1)-\partial_t\boldsymbol{B}_s(t_2)\|_{\boldsymbol{L}^2(\Omega_c)}\le C|t_1-t_2|$ for any $t_1,t_2\in[0,T]$. Therefore, we deduce that

$$\begin{aligned}\sum_{i=1}^{j}\left(\frac{\nu}{\tau}\int_{t_{i-1}}^{t_i}\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)\,dt,\,\boldsymbol{B}_{s,i}-\boldsymbol{B}_{s,i-1}\right)_{\Omega_c}-\int_0^{t_j}(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s),\,\partial_t\boldsymbol{B}_s)_{\Omega_c}\,dt&\\ =\sum_{i=1}^{j}\int_{\Omega_c}\int_{t_{i-1}}^{t_i}\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)\left(\frac{\boldsymbol{B}_{s,i}-\boldsymbol{B}_{s,i-1}}{\tau}-\partial_t\boldsymbol{B}_s\right)dt\,d\boldsymbol{x}&\\ \le C\nu^*\sum_{i=1}^{j}\left(\int_{t_{i-1}}^{t_i}\|\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)\|^2_{\boldsymbol{L}^2(\Omega_c)}\,dt\right)^{1/2}&\\ \cdot\left(\int_{t_{i-1}}^{t_i}\left\|\frac{\boldsymbol{B}_{s,i}-\boldsymbol{B}_{s,i-1}}{\tau}-\partial_t\boldsymbol{B}_s\right\|^2_{\boldsymbol{L}^2(\Omega_c)}dt\right)^{1/2}&\end{aligned}$$

$$
\begin{aligned}
&\le C\Big(\int_0^{t_j}\|\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)\|^2_{\boldsymbol{L}^2(\Omega_c)}\,dt\Big)^{1/2}\\
&\qquad\qquad\cdot\Big(\sum_{i=1}^{j}\int_{t_{i-1}}^{t_i}\Big\|\frac{\boldsymbol{B}_{s,i}-\boldsymbol{B}_{s,i-1}}{\tau}-\partial_t\boldsymbol{B}_s\Big\|^2_{\boldsymbol{L}^2(\Omega_c)}\,dt\Big)^{1/2}\\
&\le C\Big(\sum_{i=1}^{j}\int_{t_{i-1}}^{t_i}\Big\|\frac{\boldsymbol{B}_{s,i}-\boldsymbol{B}_{s,i-1}}{\tau}-\partial_t\boldsymbol{B}_s\Big\|^2_{\boldsymbol{L}^2(\Omega_c)}\,dt\Big)^{1/2}\\
&\le C\Big(\sum_{i=1}^{j}\int_{t_{i-1}}^{t_i}\Big\|\frac{1}{\tau}\int_{t_{i-1}}^{t_i}(\partial_t\boldsymbol{B}_s(\zeta)-\partial_t\boldsymbol{B}_s(t))\,d\zeta\Big\|^2_{\boldsymbol{L}^2(\Omega_c)}\,dt\Big)^{1/2}\\
&\le C\tau\xrightarrow{n\to\infty}0.
\end{aligned}
\tag{5-15}
$$

Since $\eta\to t_j$ as $n\to\infty$, we obtain

$$
\begin{aligned}
&\int_{t_j}^{\eta}(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s),\,\partial_t\boldsymbol{B}_s)_{\Omega_c}\,dt\\
&\qquad=\int_{t_j}^{\eta}(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)-\nu\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\partial_t\boldsymbol{B}_s)_{\Omega_c}\,dt\\
&\qquad\qquad\qquad+\int_{t_j}^{\eta}(\nu\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\partial_t\boldsymbol{B}_s)_{\Omega_c}\,dt\\
&\qquad\le C\int_0^T\|\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)-\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n})\|_{\boldsymbol{L}^2(\Omega_c)}\|\partial_t\boldsymbol{B}_s\|_{\boldsymbol{L}^2(\Omega_c)}\\
&\qquad\qquad\qquad+C\|\nabla\times\boldsymbol{A}_j-\boldsymbol{B}_{s,j}\|_{\boldsymbol{L}^2(\Omega_c)}\|\boldsymbol{B}_s(\eta)-\boldsymbol{B}_s(t_j)\|_{\boldsymbol{L}^2(\Omega_c)}\\
&\qquad\le C\|\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)-\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n})\|_{L^2((0,T);\boldsymbol{L}^2(\Omega_c))}\\
&\qquad\qquad\qquad+C\|\boldsymbol{B}_s(\eta)-\boldsymbol{B}_s(t_j)\|_{\boldsymbol{L}^2(\Omega_c)}\\
&\qquad\xrightarrow{n\to\infty}0,
\end{aligned}
\tag{5-16}
$$

and

$$
\begin{aligned}
&\int_{t_j}^{\eta}(\nu\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\nabla\times\partial_t\boldsymbol{A}_n)_{\Omega_c}\,dt\\
&\qquad\qquad=(\nu\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\nabla\times\boldsymbol{A}_n(\eta)-\nabla\times\boldsymbol{A}_n(t_j))_{\Omega_c}\\
&\qquad\qquad\le C\|\boldsymbol{M}(\nabla\times\boldsymbol{A}_j-\boldsymbol{B}_{s,j})\|_{\boldsymbol{L}^2(\Omega_c)}\|\nabla\times\boldsymbol{A}_n(\eta)-\nabla\times\boldsymbol{A}_n(t_j)\|_{\boldsymbol{L}^2(\Omega_c)}\\
&\qquad\qquad\le C\|\nabla\times\boldsymbol{A}_j-\boldsymbol{B}_{s,j}\|_{\boldsymbol{L}^2(\Omega_c)}\|\nabla\times\boldsymbol{A}_n(\eta)-\nabla\times\boldsymbol{A}_n(t_j)\|_{\boldsymbol{L}^2(\Omega_c)}\\
&\qquad\qquad\le C\|\nabla\times\boldsymbol{A}_n(\eta)-\nabla\times\boldsymbol{A}_n(t_j)\|_{\boldsymbol{L}^2(\Omega_c)}\xrightarrow{n\to\infty}0.
\end{aligned}
\tag{5-17}
$$

Then, collecting (5-12)–(5-16) yields

$$\begin{aligned}\lim_{n\to\infty}\int_0^\eta(\bar{\sigma}_n(t-\tau)(\partial_t\boldsymbol{A}_n+\partial_t\nabla\phi_n),\,\partial_t\boldsymbol{A}_n+\partial_t\nabla\phi_n)_{\Omega_c}\,dt&\\ \le\int_{\Omega_c}\nu\Phi_{\boldsymbol{M}}(\nabla\times\boldsymbol{A}(\eta)-\boldsymbol{B}_s(\eta))\,d\boldsymbol{x}-\int_{\Omega_c}\nu\Phi_{\boldsymbol{M}}(\nabla\times\boldsymbol{A}_0-\boldsymbol{B}_{s,0})\,d\boldsymbol{x}&\\ +\int_0^\eta(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s),\,\partial_t\boldsymbol{B}_s)_{\Omega_c}\,dt&\\ =\int_0^\eta\int_{\Omega_c}\nu\frac{d}{dt}\Phi_{\boldsymbol{M}}(\boldsymbol{M}^{-1}(\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)))\,d\boldsymbol{x}\,dt&\\ +\int_0^\eta(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s),\,\partial_t\boldsymbol{B}_s)_{\Omega_c}\,dt&\\ \overset{(3\text{-}10)}{=}\int_0^\eta\int_{\Omega_c}\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)\frac{d}{dt}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s)\,d\boldsymbol{x}\,dt&\\ +\int_0^\eta(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s),\,\partial_t\boldsymbol{B}_s)_{\Omega_c}\,dt&\\ \le\int_0^\eta(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s),\,\partial_t\nabla\times\boldsymbol{A})_{\Omega_c}\,dt.&\qquad(5\text{-}18)\end{aligned}$$

Further we use (5-9)–(5-11) and (5-16) to obtain

$$\begin{aligned}\sigma_*\lim_{n\to\infty}\int_0^\eta\int_{\Omega_c}|\partial_t\boldsymbol{A}_n+\partial_t\nabla\phi_n-\partial_t\boldsymbol{A}-\partial_t\nabla\phi|^2\,d\boldsymbol{x}\,dt&\\ \le\int_0^\eta(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}-\boldsymbol{B}_s),\,\partial_t\nabla\times\boldsymbol{A})_{\Omega_c}\,dt-\int_0^\eta(\nu\nabla\cdot\boldsymbol{A},\,\partial_t\nabla\cdot\boldsymbol{A})_{\Omega_c}\,dt&\\ -\int_0^\eta(\sigma(u)(\partial_t\boldsymbol{A}+\partial_t\nabla\phi),\,\partial_t\boldsymbol{A}+\partial_t\nabla\phi)_{\Omega_c}\,dt=0,&\end{aligned}$$

where we use (5-19) given in the proof of the following theorem. Due to the fact that the set of such η is dense in $[0, T]$, we conclude that $\partial_t\boldsymbol{A}_n+\partial_t\nabla\phi_n\to\partial_t\boldsymbol{A}+\partial_t\nabla\phi$ in $L^2((0, T);\boldsymbol{L}^2(\Omega_c))$. □

Now we are in a position to give our main result of this paper.

Theorem 5.5. *Let the assumptions of Propositions 5.3–5.4 be satisfied. There exist a solution* $(\boldsymbol{A},\phi)\in L^2((0,T);\boldsymbol{V})$ *and a solution* $u\in C([0,T];L^2(\Omega_c))\cap L^\infty((0,T);H^1(\Omega_c))$ *with* $\partial_t u\in L^2((0,T);L^2(\Omega_c))$ *such that* $(\boldsymbol{A},\phi)$ *and* u *solve* (5-2) *and* (5-3).

Proof. We first prove that $(\boldsymbol{A},\phi)$ and u solve (5-2). Let us integrate (5-6) in time to obtain

$$\begin{aligned}\int_0^\eta(\bar{\sigma}_n(t-\tau)(\partial_t\boldsymbol{A}_n+\partial_t\nabla\phi_n),\,\boldsymbol{Q}+\nabla\psi)_{\Omega_c}\,dt&\\ +\int_0^\eta(\nu\boldsymbol{M}(\nabla\times\bar{\boldsymbol{A}}_n-\bar{\boldsymbol{B}}_{s,n}),\,\nabla\times\boldsymbol{Q})_{\Omega_c}\,dt+\int_0^\eta(\nu\nabla\cdot\bar{\boldsymbol{A}}_n,\,\nabla\cdot\boldsymbol{Q})_{\Omega_c}\,dt=0.&\end{aligned}$$

Using Propositions 5.3(ii) and 5.4, we pass to the limit for $n \to \infty$ to see

$$\int_0^\eta (\sigma(u)(\partial_t \boldsymbol{A} + \partial_t \nabla\phi),\ \boldsymbol{Q} + \nabla\psi)_{\Omega_c}\,dt + \int_0^\eta (\nu \boldsymbol{M}(\nabla \times \boldsymbol{A} - \boldsymbol{B}_s),\ \nabla \times \boldsymbol{Q})_{\Omega_c}\,dt \\ + \int_0^\eta (\nu \nabla \cdot \boldsymbol{A},\ \nabla \cdot \boldsymbol{Q})_{\Omega_c}\,dt = 0. \quad (5\text{-}19)$$

We differentiate in time to conclude that $(\boldsymbol{A}, \phi, u)$ solve (5-2).

To show that $(\boldsymbol{A}, \phi, u)$ solve (5-3), we integrate (5-7) in time:

$$(\bar{\theta}_n(t), v)_{\Omega_c} - (\theta_n(0), v)_{\Omega_c} + \int_0^t (\bar{\lambda}_n \nabla \bar{u}_n, v)_{\Omega_c}\,ds \\ = \int_0^t (\mathcal{R}_r(\bar{\sigma}_n(s-\tau)|\partial_t \boldsymbol{A}_n + \partial_t \nabla\phi_n|^2), v)_{\Omega_c}\,ds.$$

According to Proposition 5.3(iii) we see that

$$\lim_{n\to\infty} (\theta_n(t) - \bar{\theta}_n(t), v)_{\Omega_c} = 0 \quad \text{for every } t \in [0, T].$$

Due to Proposition 5.4(vi) and the fact that the function $\mathcal{R}_r$ is continuous and bounded, we can apply Lebesgue's dominated convergence theorem to pass to the limit on the right-hand side and obtain

$$\lim_{n\to\infty} \int_0^t (\mathcal{R}_r(\bar{\sigma}_n(s-\tau)|\partial_t \boldsymbol{A}_n + \partial_t \nabla\phi_n|^2), v)_{\Omega_c}\,ds \\ = \int_0^t (\mathcal{R}_r(\sigma(u)|\partial_t \boldsymbol{A} + \partial_t \nabla\phi|^2), v)_{\Omega_c}\,ds.$$

Let us combine these results and pass to the limit for $n \to \infty$ in the variational equation above. Thus, we have

$$(\theta(u(t)), v)_{\Omega_c} - (\theta(u(0)), v)_{\Omega_c} + \int_0^t (\lambda \nabla u, v)_{\Omega_c}\,ds \\ = \int_0^t (\mathcal{R}_r(\gamma(u)|\partial_t \boldsymbol{A} + \partial_t \nabla\phi|^2), v)_{\Omega_c}\,ds.$$

Differentiation with respect to the time variable yields (5-3), which also concludes the proof. □

6. Numerical simulation

In this section we present a fully discrete finite element scheme based on (4-1)–(4-2) and show some numerical simulation results. Let $\mathcal{T}_h$ be a standard tetrahedral triangulation of Ω with a match grid on $\partial\Omega_c$. We define

$$\boldsymbol{Y}_h^0 := \{\boldsymbol{Q} \in \widehat{\boldsymbol{H}}_0^1(\Omega) : \boldsymbol{Q}|_{\mathcal{K}} \in (\mathcal{P}_r)^3 \text{ for all } \mathcal{K} \in \mathcal{T}_h\}, \\ W_h := \{\psi \in H^1(\Omega) : \psi|_{\mathcal{K}} \in \mathcal{P}_r \text{ for all } \mathcal{K} \in \mathcal{T}_h\},$$

where $\mathcal{P}_r$ is the space of polynomials with degree $\leq r$. Let $\boldsymbol{V}_h := \boldsymbol{Y}_h^0 \times W_h/\mathbb{R}$.

Algorithm. Given the initial value $(\boldsymbol{A}_0, \phi_0, u_0)$, we suggest a computing scheme for obtaining the solution $(\boldsymbol{A}_{h,i}, \phi_{h,i}, u_{h,i})$ for every time step $t = t_i$:

(1) Let i be given and assume that $\boldsymbol{A}_{h,i-1}$, $\phi_{h,i-1}$, $u_{h,i-1}$, λ_i and $\boldsymbol{B}_{s,i}$ are known.

(2) Find $(\boldsymbol{A}_{h,i}, \phi_{h,i}) \in \boldsymbol{V}_h$ such that

$$\begin{aligned}&(\sigma(u_{h,i-1})\delta\boldsymbol{A}_{h,i} + \sigma(u_{h,i-1})\delta\nabla\phi_{h,i},\ \boldsymbol{Q}_h + \nabla\psi_h)_{\Omega_c}\\ &+(\nu\boldsymbol{M}(\nabla\times\boldsymbol{A}_{h,i} - \boldsymbol{B}_{s,i}), \nabla\times\boldsymbol{Q}_h)_{\Omega} + (\nu\nabla\cdot\boldsymbol{A}_{h,i}, \nabla\cdot\boldsymbol{Q}_h)_{\Omega} = 0 \quad \text{for all } (\boldsymbol{Q}_h, \psi_h) \in \boldsymbol{V}_h.\end{aligned}$$

(3) Find $u_{h,i} \in W_h$ such that

$$(\delta\theta(u_{h,i}), v_h)_{\Omega_c} + (\lambda_i\nabla u_{h,i}, \nabla v_h)_{\Omega_c} = (\mathcal{R}_r(\sigma(u_{h,i-1})|\delta\boldsymbol{A}_{h,i} + \delta\nabla\phi_{h,i}|^2), v_h)_{\Omega_c} \quad \text{for all } v_h \in W_h.$$

(4) Set $i = i + 1$ and repeat the process.

Experiment 6.1. This experiment is to check the change of the error as the time step decreases under the fixed standard tetrahedral triangulation.

The workpiece and the computational domain are given in Figure 2, left. The source current density is shown in Figure 2, right. Unknown functions representing nonlinearities are chosen accordingly to satisfy

$$\boldsymbol{M} = (1+\exp(-|\nabla\times\boldsymbol{A}|))\nabla\times\boldsymbol{A}, \qquad \sigma(u) = 4 - \left(1 + \frac{1}{1+u}\right)^{1+u}, \qquad \theta = u + \sqrt{u}.$$

The electric field is decomposed as $\boldsymbol{E} = \partial_t\boldsymbol{A} + \partial_t\nabla\phi$. The initial values and the parameters are given as

$$\boldsymbol{A}_0 = \boldsymbol{0}, \qquad \phi_0 = 0, \qquad u_0 = 273.15, \qquad \lambda = 0.1, \qquad \nu = 1.$$

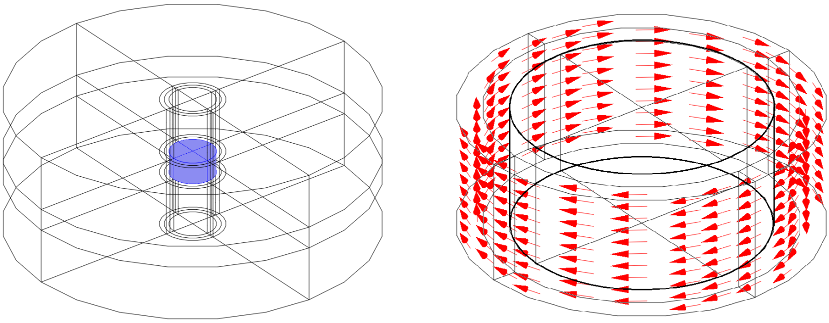


Figure 2. Induction hardening model. Left: workpiece and computational domain. Right: source current density.

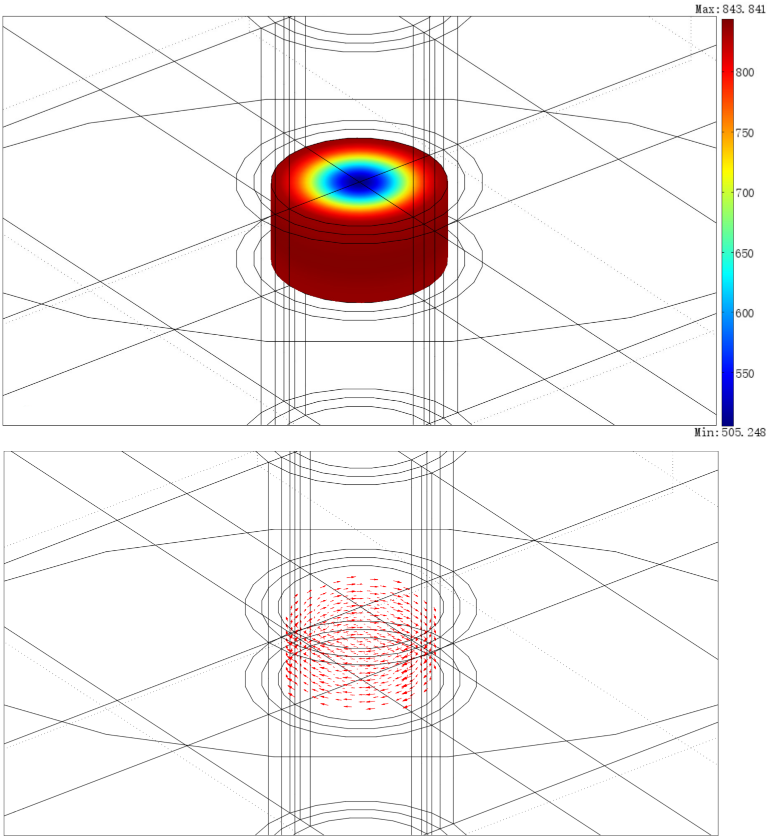


Figure 3. Reference solutions in the workpiece at $t = 1.0$. Top: temperature distribution. Bottom: electric field.

We choose the source current density

$$\boldsymbol{J}_s = 500\sin(2\pi t)\begin{pmatrix} -y/(x^2+y^2) \\ x/(x^2+y^2) \\ 0 \end{pmatrix}.$$

We partition the time interval [0, 1] into 1280 equidistant parts and solve the system at each time step by using linear nodal elements as implemented in the software package COMSOL. We denote the obtained solutions for the electric field and the temperature function as reference solutions $\boldsymbol{E}_{\text{ref}}$ and u_{ref}, respectively, which are plotted in Figure 3.

To show the convergence of the approximate solutions of our scheme, we compute other numerical solutions at time steps $\tau = 1/(2^n \times 10)$, $n = 0, 1, \ldots, 6$, and compare them with $\boldsymbol{E}_{\text{ref}}$ and u_{ref}. We consider some specific measurement points distributed in the conducting domain and analyze these solutions at time steps

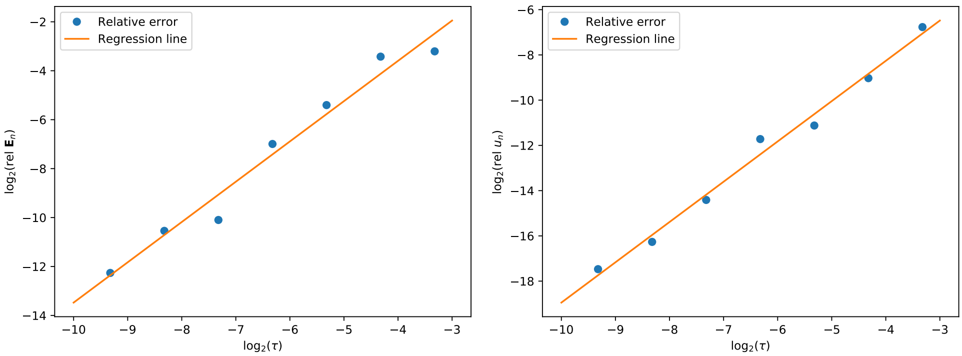


Figure 4. Logarithmically scaled plot of the decreasing time step τ and the relative errors. Left: relative error of the electric field $\boldsymbol{E}$ with respect to a decreasing time step τ. Right: relative error of the temperature u with respect to a decreasing time step τ.

$t_i = 0.1i$, $i = 0, 1, \ldots, 10$. Relative errors of a given numerical solution $\boldsymbol{E}_n$ from the reference solution $\boldsymbol{E}_{\text{ref}}$ and u_n from u_{ref} are then calculated as

$$|\boldsymbol{E}_{\text{ref}}| = \sum_{p_j \in P} \sum_{i=0}^{10} |\boldsymbol{E}_{\text{ref}}(P_j, t_i)|, \qquad |u_{\text{ref}}| = \sum_{p_j \in P} \sum_{i=0}^{10} |u_{\text{ref}}(P_j, t_i)|,$$

$$|\boldsymbol{E}_{\text{ref}} - \boldsymbol{E}_n| = \sum_{p_j \in P} \sum_{i=0}^{10} |\boldsymbol{E}_{\text{ref}}(P_j, t_i) - \boldsymbol{E}_n(P_j, t_i)|,$$

$$|u_{\text{ref}} - u_n| = \sum_{p_j \in P} \sum_{i=0}^{10} |u_{\text{ref}}(P_j, t_i) - u_n(P_j, t_i)|,$$

$$\text{Rel } \boldsymbol{E}_n = \frac{|\boldsymbol{E}_{\text{ref}} - \boldsymbol{E}_n|}{|\boldsymbol{E}_{\text{ref}}|}, \qquad \text{Rel } u_n = \frac{|u_{\text{ref}} - u_n|}{|u_{\text{ref}}|},$$

where P is the set of measurement points and the index n stands for the number of the time partition. The evolution of the relative errors with decreasing the time step τ is illustrated in Figure 4. The regression line in Figure 4, left, is $\log_2(\text{Rel } \boldsymbol{E}_n) = 1.6464 \log_2 \tau + 2.9898$ and in Figure 4, right, $\log_2(\text{Rel } u_n) = 1.7810 \log_2 \tau - 1.1387$, resulting in the convergence of the approximate solutions to the reference solutions.

Experiment 6.2. For the model (see Figure 1), this experiment is to give some numerical simulations and show the eddy current and temperature distributions when varying the current frequency ω.

As shown in Figure 5, we set the source current density

$$\boldsymbol{J}_s = 500 \sin(\omega t) \begin{pmatrix} -y/(x^2+y^2) \\ x/(x^2+y^2) \\ 0 \end{pmatrix}.$$

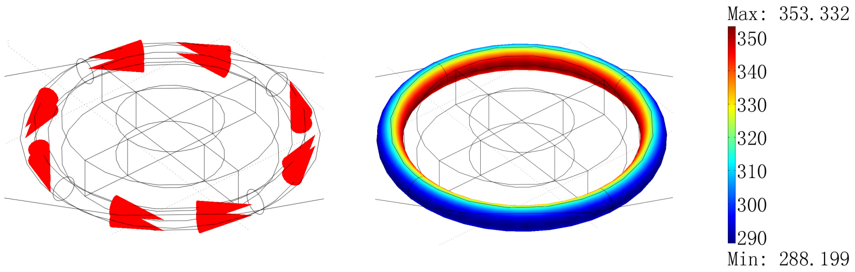
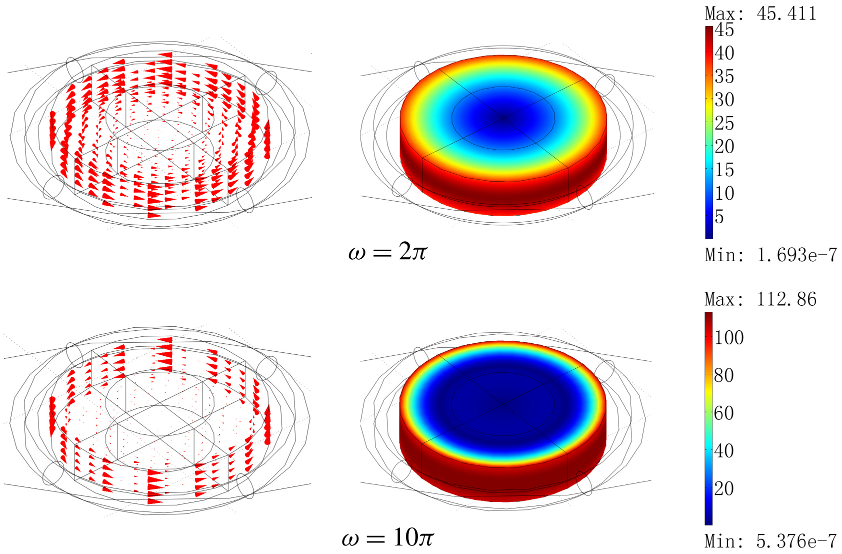


Figure 5. Source current density at $\omega = 2\pi$, $t = 0.2$. Left: arrow plot. Right: distribution plot.

The time interval is [0, 1] and the time step is $\tau = 0.2\pi/\omega$. To simulate the induction harden process as closely as possible to the previous theoretical analysis, we set

$$\boldsymbol{E} = \partial_t \boldsymbol{A} + \partial_t \nabla\phi, \quad \boldsymbol{M}(\nabla\times\boldsymbol{A}) = (1+e^{-|\nabla\times\boldsymbol{A}|})\nabla\times\boldsymbol{A},$$
$$\theta(u) = 100\sqrt{u}, \quad \sigma(u) = 4 - \left(1+\frac{1}{1+u}\right)^{1+u},$$
$$\lambda = 1, \qquad \nu = 1, \qquad \sigma_0 = 10^{-12}, \qquad u_0 = 293.$$

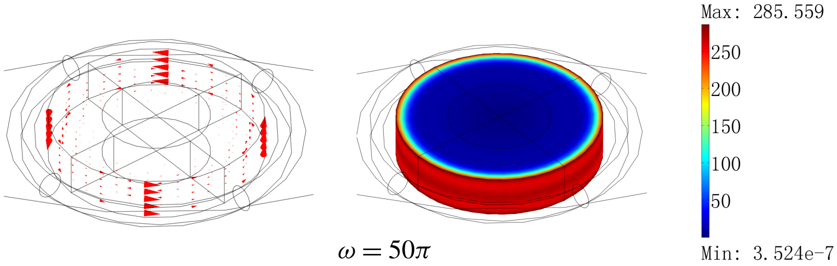
We adopt linear nodal elements by using COMSOL to simulate the induction harden process. Let ω take the values 2π, 10π, and 50π. The eddy current density $\sigma\boldsymbol{E} = -\sigma\partial_t(\boldsymbol{A}+\nabla\phi)$ for different coefficients ω at $t = 1.0$ is computed and shown in Figures 6 and 7, while the temperature u is shown in Figure 8.



$\omega = 2\pi$

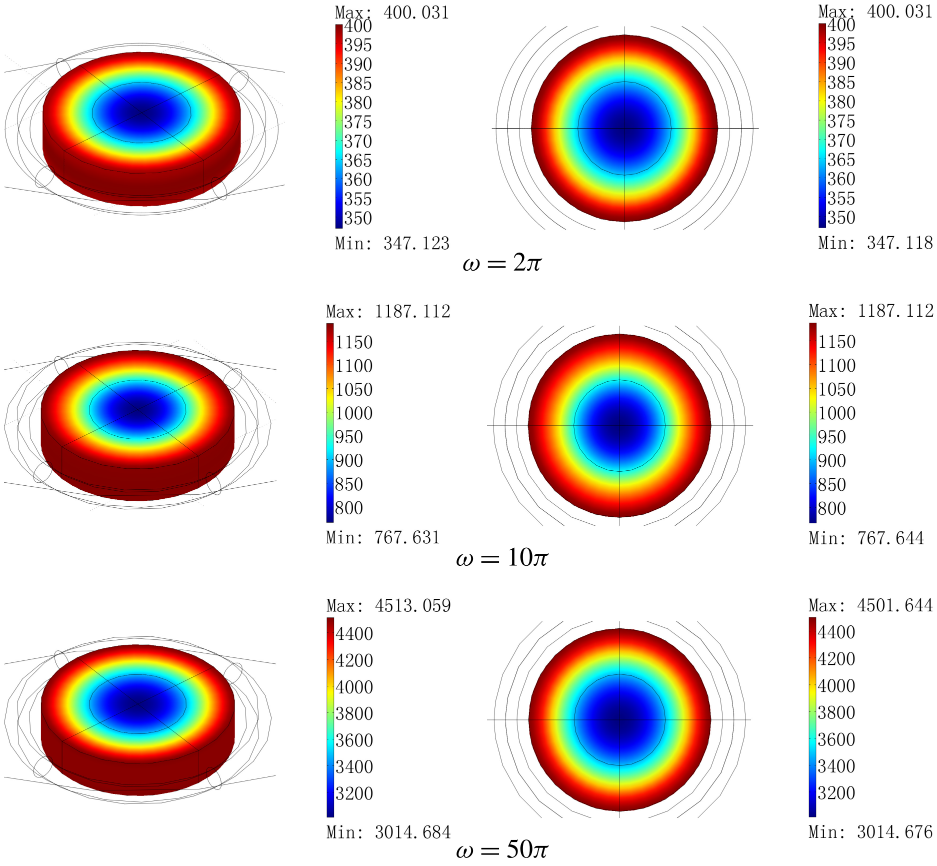
$\omega = 10\pi$

Figure 6. Eddy current density at $t = 1.0$ for $\omega = 2\pi$ (top) and $\omega = 10\pi$: arrow plot and distribution plot.



$\omega = 50\pi$

Figure 7. Eddy current density at $t = 1.0$ for $\omega = 50\pi$: arrow plot and distribution plot.



$\omega = 2\pi$

$\omega = 10\pi$

$\omega = 50\pi$

Figure 8. Temperature distribution at $t = 1.0$ for $\omega = 2\pi$, $\omega = 10\pi$ and $\omega = 50\pi$: side view (left) and center section (right).

The numerical simulation results show that the induced eddy currents in the workpiece dissipate energy and bring about Joule heating. The magnitude of the eddy currents decreases with growing distance from the workpiece surface. As the current frequency ω increases, the eddy current distribution concentrates on the outer side of the workpiece. Moreover, the temperature distribution is also affected by ω. The overall temperature of the workpiece rises with the frequency ω increasing, and the temperature difference between the inner and outer sides of the workpiece gradually grows. It shows that the change in temperature is consistent with the distribution of eddy currents.

7. Conclusion

In this paper we introduce the $\boldsymbol{A}$-ϕ method based on decomposition of the electric field to study an induction hardening model with a nonlinear relation between the magnetic field and the magnetic induction field. We have proven the existence of a weak solution only in the conducting domain. Due to technological reasons, we cannot analyze convergence for the model in both conducting and nonconducting domains. Some results of the numerical simulation shown here are reasonable. Since we do not prove uniqueness of the weak solution, we could not further study the convergence of the fully discrete scheme rigorously. In the future we would like to investigate the general case and provide a proof of a unique solution. But we need to overcome some difficulties in mathematical analysis, which come from the coupling between the nonlinear $\boldsymbol{A}$-ϕ equations and the heat equation in the form of the temperature-dependent function.

Acknowledgement

We thank China–Sri Lanka Joint Center for Education and Research, Chinese Academy of Sciences for research infrastructure assistance.

References

[1] G. Akrivis and S. Larsson, *Linearly implicit finite element methods for the time-dependent Joule heating problem*, BIT **45** (2005), no. 3, 429–442. MR Zbl

[2] R. Albanese and G. Rubinacci, *Formulation of the eddy-current problem*, IEE Proc. A **137** (1990), no. 1, 16–22.

[3] A. Alonso Rodríguez and A. Valli, *Eddy current approximation of Maxwell equations: theory, algorithms and applications*, Modeling, Simulation and Applications, no. 4, Springer, 2010. MR Zbl

[4] C. Amrouche, C. Bernardi, M. Dauge, and V. Girault, *Vector potentials in three-dimensional non-smooth domains*, Math. Methods Appl. Sci. **21** (1998), no. 9, 823–864. MR Zbl

[5] J. Barglik, I. Doležel, P. Karban, and B. Ulrych, *Modelling of continual induction hardening in quasi-coupled formulation*, COMPEL **24** (2005), no. 1, 251–260. Zbl

[6] A. Bermúdez, D. Gómez, M. C. Muñiz, and P. Salgado, *Transient numerical simulation of a thermoelectrical problem in cylindrical induction heating furnaces*, Adv. Comput. Math. **26** (2007), no. 1-3, 39–62. MR Zbl

[7] O. Bíró and K. Preis, *On the use of the magnetic vector potential in the finite-element analysis of three-dimensional eddy currents*, IEEE T. Magn. **25** (1989), no. 4, 3145–3159.

[8] O. Bíró, K. Preis, G. Buchgraber, and I. Tičar, *Voltage-driven coils in finite-element formulations using a current vector and a magnetic scalar potential*, IEEE T. Magn. **40** (2004), no. 2, 1286–1289.

[9] J. Chovan, C. Geuzaine, and M. Slodička, *A-φ formulation of a mathematical model for the induction hardening process with a nonlinear law for the magnetic field*, Comput. Methods Appl. Mech. Engrg. **321** (2017), 294–315. MR

[10] C. M. Elliott and S. Larsson, *A finite element model for the time-dependent Joule heating problem*, Math. Comp. **64** (1995), no. 212, 1433–1453. MR Zbl

[11] L. C. Evans, *Partial differential equations*, Graduate Studies in Mathematics, no. 19, American Mathematical Society, Providence, RI, 1998. MR Zbl

[12] D. Hömberg, *A mathematical model for induction hardening including mechanical effects*, Nonlinear Anal. Real World Appl. **5** (2004), no. 1, 55–90. MR Zbl

[13] D. Hömberg, T. Petzold, and E. Rocca, *Analysis and simulations of multifrequency induction hardening*, Nonlinear Anal. Real World Appl. **22** (2015), 84–97. MR Zbl

[14] T. Kang, T. Chen, H. Zhang, and K. I. Kim, *Fully discrete A-ϕ finite element method for Maxwell's equations with nonlinear conductivity*, Numer. Methods Partial Differential Equations **30** (2014), no. 6, 2083–2108. MR Zbl

[15] ———, *A-ϕ finite element method with composite grids for time-dependent eddy current problem*, Appl. Math. Comput. **267** (2015), 365–381. MR Zbl

[16] T. Kang and K. I. Kim, *Fully discrete potential-based finite element methods for a transient eddy current problem*, Computing **85** (2009), no. 4, 339–362. MR Zbl

[17] J. Kačur, *Method of Rothe in evolution equations*, Teubner-Texte zur Mathematik, no. 80, Teubner, Leipzig, 1985. MR Zbl

[18] K. I. Kim and T. Kang, *A potential-based finite-element for time-dependent Maxwell's equations*, Int. J. Comput. Math. **83** (2006), no. 1, 107–122. MR Zbl

[19] B. Li, H. Gao, and W. Sun, *Unconditionally optimal error estimates of a Crank–Nicolson Galerkin method for the nonlinear thermistor equations*, SIAM J. Numer. Anal. **52** (2014), no. 2, 933–954. MR Zbl

[20] J. Nečas, *Introduction to the theory of nonlinear elliptic equations*, Wiley, Chichester, 1986. MR Zbl

[21] M. Slodička, *A time discretization scheme for a non-linear degenerate eddy current model for ferromagnetic materials*, IMA J. Numer. Anal. **26** (2006), no. 1, 173–187. MR Zbl

[22] M. Slodička and J. Chovan, *Solvability for induction hardening including nonlinear magnetic field and controlled Joule heating*, Appl. Anal. **96** (2017), no. 16, 2780–2799. MR Zbl

[23] D. Sun, V. S. Manoranjan, and H.-M. Yin, *Numerical solutions for a coupled parabolic equations arising induction heating processes*, Discrete Contin. Dyn. Syst. Suppl. (2007), 956–964. MR Zbl

[24] M. M. Vaĭnberg, *Variational method and method of monotone operators in the theory of nonlinear equations*, Halsted, New York, 1973. MR Zbl

[25] S. Yan, J.-M. Jin, C.-F. Wang, and J. D. Kotulski, *Numerical study of a time-domain finite element method for nonlinear magnetic problems in three dimensions*, Prog. Electromagn. Res. **153** (2015), 69–91.

[26] H.-M. Yin and W. Wei, *Regularity of weak solution for a coupled system arising from a microwave heating model*, Euro. J. Appl. Math. **25** (2014), no. 1, 117–131. Zbl

[27] H.-M. Yin, *On a nonlinear Maxwell's system in quasi-stationary electromagnetic fields*, Math. Models Methods Appl. Sci. **14** (2004), no. 10, 1521–1539. MR Zbl

[28] ———, *Regularity of weak solution to Maxwell's equations and applications to microwave heating*, J. Differential Equations **200** (2004), no. 1, 137–161. MR Zbl

Received August 9, 2018. Revised April 22, 2019.

TONG KANG: kangtong@cuc.edu.cn
Department of Applied Mathematics, School of Sciences, Communication University of China, Beijing, China

RAN WANG: ranwang.osbert@outlook.com
Department of Applied Mathematics, School of Sciences, Communication University of China, Beijing, China

and

Key Laboratory of Computational Geodynamics, University of Chinese Academy of Sciences, Beijing, China

HUAI ZHANG: hzhang@ucas.ac.cn
Key Laboratory of Computational Geodynamics, University of Chinese Academy of Sciences, Beijing, China

and

Laboratory for Marine Mineral Resources, Qingdao National Laboratory for Marine Science and Technology, Qingdao, China

COMM. APP. MATH. AND COMP. SCI.
Vol. 14, No. 2, 2019
dx.doi.org/10.2140/camcos.2019.14.207

COMPUTING THE QUASIPOTENTIAL FOR HIGHLY DISSIPATIVE AND CHAOTIC SDES AN APPLICATION TO STOCHASTIC LORENZ'63

MARIA CAMERON AND SHUO YANG

The study of noise-driven transitions occurring rarely on the time scale of systems modeled by SDEs is of crucial importance for understanding such phenomena as genetic switches in living organisms and magnetization switches of the Earth. For a gradient SDE, the predictions for transition times and paths between its metastable states are done using the potential function. For a nongradient SDE, one needs to decompose its forcing into a gradient of the so-called quasipotential and a rotational component, which cannot be done analytically in general.

We propose a methodology for computing the quasipotential for highly dissipative and chaotic systems built on the example of Lorenz'63 with an added stochastic term. It is based on the ordered line integral method, a Dijkstra-like quasipotential solver, and combines 3D computations in whole regions, a dimensional reduction technique, and 2D computations on radial meshes on manifolds or their unions. Our collection of source codes is available on M. Cameron's web page and on GitHub.

1. Introduction

Suppose a system is evolving according to a stochastic differential equation (SDE) of the form

$$d\boldsymbol{x} = \boldsymbol{b}(\boldsymbol{x})\,dt + \sqrt{\epsilon}\,d\boldsymbol{w}, \quad \boldsymbol{x} \in \mathbb{R}^d, \tag{1}$$

where $\boldsymbol{b}(\boldsymbol{x})$ is a continuously differentiable vector field, $d\boldsymbol{w}$ is the standard Brownian motion, and ϵ is a small parameter. The quasipotential is a key function of the large deviation theory (LDT) [14] that allows one to find a collection of useful asymptotic estimates for long-time dynamics of such systems. They include the invariant probability measure, expected escape times from neighborhoods of attractors of the corresponding ODE $\dot{\boldsymbol{x}} = \boldsymbol{b}(\boldsymbol{x})$ lying within their basins, and maximum likelihood escape paths from the basins. The quasipotential can be viewed as an analogue to the potential function $V(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^d$, for a gradient SDE with deterministic term

MSC2010: 65N99, 65P99, 58J65.

Keywords: quasipotential, ordered line integral method, Lorenz'63, maximum likelihood transition path.

$-\nabla V(\boldsymbol{x})$. The quasipotential is defined as the solution to the Freidlin–Wentzell action functional minimization problem. The quasipotential is Lipschitz-continuous in any bounded domain but not necessarily continuously differentiable [3]. Unfortunately, it can be found analytically only in special cases, for example, for linear SDEs [8; 7].

Ordered line integral methods (OLIMs) for computing the quasipotential for SDEs of the form (1) in whole regions on regular rectangular meshes were introduced in [11] for 2D and extended to 3D in [35]. They are Dijkstra-like solvers that advance the solution from mesh points with smaller values to those with larger values[1] without iteration. Their general structure is inherited from the ordered upwind method (OUM) [27; 28], but there are important differences. First, unlike the OUM that uses the upwind finite difference scheme, the OLIMs solve a local functional minimization problem at every step approximating a segment of curve with a segment of straight line, and the integral along it by an at least second-order accurate quadrature rule. This renders their observed rate of convergence superlinear for some cases, and reduces error constants by two to three orders of magnitude in comparison with the OUM. Second, while the OUM is practical only for 2D problems due to large CPU times in larger dimensions, the OLIMs have been successfully extended for 3D. This became possible due to the hierarchical update strategy [11; 35], the use of the Karush–Kuhn–Tucker optimality conditions to eliminate unnecessary updates, and a number of implementational rationalizations.

In previous works [11; 10; 35], the OLIMs were developed for computing the quasipotential for mild-to-moderate ratio $\Xi(\boldsymbol{x})$ of the magnitudes of the rotational and potential components of the vector field $\boldsymbol{b}(\boldsymbol{x})$ in (1). In all test problems considered in [11; 10; 35], $\Xi(\boldsymbol{x})$ did not exceed 10 within the important region around the attractor with respect to which the quasipotential was computed. For all these test problems, the black-box algorithms [11; 10; 35] produced numerical solutions with small relative errors.

Unfortunately, if one applies the black-box `olim3D` quasipotential solver from [35] to a highly dissipative and chaotic system such as Lorenz’63 with an added small white noise, the relative error of the numerical solution might be large leading to completely wrong estimates for escape rates. For the parameter values $\sigma = 10$, $\beta = \frac{8}{3}$, and $\rho \gtrsim 15$, the quasipotential computed with respect to one of the point attractors will become progressively inaccurate as ρ increases. We show in this work that, as ρ approaches $\rho_2 \approx 24.74$ (where a subcritical Hopf bifurcation happens), the upper bound for the ratio $\Xi(\boldsymbol{x})$ blows up at any point of the computational domain of interest. Even if one uses a very good desktop computer,[2] this problem

[1] This is only approximately true. See [28] for details.

[2] We use a 2017 iMac with a 4.2 GHz Intel Core i7 processor and 64 GB of 2400 MHz DDR4 memory.

cannot be cured by mesh refinement due to the computer's limited memory: the size of a 3D mesh cannot exceed 1000^3 by much.

In this work, we propose an approach for computing the quasipotential, finding maximum likelihood transition paths, and estimating escape times from basins of attractors for highly dissipative and possibly chaotic systems perturbed by small white noise. This approach is suitable for systems where the 3D dynamics, after some short transition time, takes place in a small neighborhood of a 2D manifold or a union of 2D manifolds consisting of certain characteristics of the corresponding ODE (see Assumption 4.2 in Section 4B below). Whether or not this phenomenon takes place can be identified from the plots of the 3D level sets of the computed quasipotential. We develop a technique for extracting these manifolds and generating so-called radial meshes on them. We adjust and test the OLIM for 2D radial meshes and compute the quasipotential on the constructed 2D manifolds or their unions.

The proposed techniques have been developed on the stochastic Lorenz'63:

$$d\boldsymbol{x} = \begin{bmatrix} \sigma(x_2 - x_1) \\ x_1(\rho - x_3) - x_2 \\ x_1x_2 - \beta x_3 \end{bmatrix} dt + \sqrt{\epsilon}\, d\boldsymbol{w}, \quad \text{where } \boldsymbol{x} \equiv \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \tag{2}$$

with $\sigma = 10$, $\beta = \frac{8}{3}$, and $0.5 \le \rho < \rho_2 \approx 24.74$. To the best of our knowledge, this is the first time when the quasipotential is computed for a chaotic 3D system in the whole region and 3D computations are refined by 2D computations on certain manifolds. We study transitions between the stable equilibria at $\rho = 12$, 15, and 20, and between the stable equilibria and the strange attractor at $\rho = 24.4$, and find a collection of quasipotential barriers for them. Our transition paths obtained by a direct integration using the computed quasipotential can be compared to those found in [37] using the minimum action method, a path-based method consisting of a direct minimization of the Freidlin–Wentzell action in the path space. At $\rho = 24.4$, we compare two plausible transition mechanisms from the strange attractor to the equilibria. We offer a number of plots of 3D level sets of the quasipotential at various values of ρ varying from 0.5 to 24.4 and supplement them with links to YouTube videos for a better 3D visualization. For $\rho \ge 15$, when 2D approximation becomes accurate enough, we perform refined 2D computations of the quasipotential.

Aiming at making our results readily reproducible, we made most of the codes developed for this project publicly available at M. Cameron's web page [5] — see the package `Qpot4lorenz63.zip` — and on GitHub [4]. All codes mentioned throughout this paper are included in this package. A user guide for the codes is also provided there.

The techniques developed in this work can be used for analysis of other stochastic systems. For example, the computation of the quasipotential for the 3D genetic switch model from [23] would benefit from performing a refined 2D computation on a radial mesh on a 2D manifold as suggested by Figure 9 in [35]. Gissinger's 3D model [15] relevant for the reversals of the magnetic field of the Earth can be analyzed using the tools developed in this work.

The rest of the paper is organized as follows. In Section 2, some necessary background on the quasipotential is given. A brief overview of the dynamics of Lorenz'63 at $\sigma = 10$, $\beta = \frac{8}{3}$, and $0 < \rho < \infty$ is offered in Section 3 and Appendix B. Numerical techniques for computing the quasipotential are described in Section 4. The application to stochastic Lorenz'63 is presented in Section 5. We summarize our findings in Section 6. Some technical details are explained in Appendices A–G.

2. Definition and significance of the quasipotential

To explain what the quasipotential is [14], we first assume that the vector field $\boldsymbol{b}(\boldsymbol{x})$ in SDE (1) admits the smooth orthogonal decomposition

$$\boldsymbol{b}(\boldsymbol{x}) = -\tfrac{1}{2}\nabla u(\boldsymbol{x}) + \boldsymbol{l}(\boldsymbol{x}), \qquad \nabla u(\boldsymbol{x}) \cdot \boldsymbol{l}(\boldsymbol{x}) = 0. \tag{3}$$

If $\boldsymbol{l}(\boldsymbol{x}) \equiv \boldsymbol{0}$, i.e., if the field $\boldsymbol{b}(\boldsymbol{x})$ were gradient, the Gibbs measure

$$\mu(\boldsymbol{x}) = Z^{-1} e^{-u(\boldsymbol{x})/\epsilon} \tag{4}$$

would be the invariant probability density for SDE (1). Suppose $\boldsymbol{l}(\boldsymbol{x})$ is not identically zero. Plugging the Gibbs measure (4) into the stationary Fokker–Planck equation for SDE (1)

$$\tfrac{1}{2}\Delta\mu(\boldsymbol{x}) - \nabla \cdot (\mu(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x})) = 0, \tag{5}$$

we find that it is invariant if and only if $\boldsymbol{l}(\boldsymbol{x})$ is divergence-free, i.e., $\nabla \cdot \boldsymbol{l}(\boldsymbol{x}) \equiv 0$. In this case, the function $u(\boldsymbol{x})$ would play the role of a potential.

Unfortunately, the orthogonal decomposition (3) where $\boldsymbol{l}(\boldsymbol{x})$ is divergence-free does not typically exist. However, a function $U(\boldsymbol{x})$ called the quasipotential that gives asymptotic estimates for the invariant probability measure near attractors of $\dot{\boldsymbol{x}} = \boldsymbol{b}(\boldsymbol{x})$ in the limit $\epsilon \to 0$ can be designed [14].

Suppose that the vector field $\boldsymbol{b}(\boldsymbol{x})$ is continuously differentiable. In addition, we assume that the ODE

$$\dot{\boldsymbol{x}} = \boldsymbol{b}(\boldsymbol{x}) \tag{6}$$

has a finite number of attractors, and every trajectory of (6) remains in a bounded region as $t \to \infty$. Let A be an attractor of (6). The quasipotential with respect to A is defined as the solution of the minimization problem

$$U(\boldsymbol{x}) = \inf_{\phi, T_0, T_1} \{ S_{T_0,T_1}(\phi) \mid \phi(T_0) \in A,\ \phi(T_1) = \boldsymbol{x} \}, \tag{7}$$

where the infimum of the Freidlin–Wentzell action

$$S_{T_0,T_1}(\phi) = \frac{1}{2}\int_{T_0}^{T_1} \|\dot{\phi} - \boldsymbol{b}(\phi)\|^2\, dt \tag{8}$$

is taken over the set of absolutely continuous paths ϕ with endpoints at A and $\boldsymbol{x}$, and all times $T_0, T_1 \in \mathbb{R}$. The infimum with respect to T_0 and T_1 can be taken analytically [14; 19; 18] resulting in the geometric action (see Appendix A)

$$S(\psi) = \int_0^L (\|\psi'\|\|\boldsymbol{b}(\psi)\| - \psi' \cdot \boldsymbol{b}(\psi))\, ds, \tag{9}$$

where the path ψ is parametrized by its arclength, and L is the length of ψ. As a result, the definition of the quasipotential can be rewritten in terms of the geometric action:

$$U(\boldsymbol{x}) = \inf_{\psi}\{S(\psi) \mid \psi(0) \in A,\ \psi(L) = \boldsymbol{x}\}. \tag{10}$$

We have been using definition (10) to develop quasipotential solvers.

Using Bellman's principle of optimality [1], one can show [3] that the quasipotential $U(\boldsymbol{x})$ satisfies the Hamilton–Jacobi equation (see Appendix A)

$$\tfrac{1}{2}\|\nabla U(\boldsymbol{x})\|^2 + \boldsymbol{b}(\boldsymbol{x}) \cdot \nabla U(\boldsymbol{x}) = 0, \quad U(A) = 0. \tag{11}$$

Equation (11) implies that

$$\boldsymbol{b}(\boldsymbol{x}) = -\tfrac{1}{2}\nabla U(\boldsymbol{x}) + \boldsymbol{l}(\boldsymbol{x}), \quad \text{where } \boldsymbol{l}(\boldsymbol{x}) := \boldsymbol{b}(\boldsymbol{x}) + \tfrac{1}{2}\nabla U(\boldsymbol{x}) \text{ is orthogonal to } \nabla U(\boldsymbol{x}). \tag{12}$$

We will refer to $-\frac{1}{2}\nabla U(\boldsymbol{x})$ and $\boldsymbol{l}(\boldsymbol{x})$ as the potential and rotational components, respectively.

We remark that the boundary value problem (BVP) (11) is ill-posed. It always has the trivial solution identically equal to zero and may or may not have a smooth nontrivial solution. The quasipotential defined by (7) or (10) is a viscosity solution[3] to (11) [9]. The other complication is that even a nontrivial solution to this BVP, classical or viscosity, may not be unique due to the fact that the boundary condition is imposed on an attractor [20]. For example, if $\boldsymbol{b}(\boldsymbol{x}) = B\boldsymbol{x}$ where B is a matrix with all eigenvalues having negative real parts, the number of solutions of (11) with the BC $u(\boldsymbol{0}) = 0$ is equal to the number of invariant subspaces for B.

Nonetheless, (11) is instrumental in deriving the equation for *minimum action paths* (MAPs) also known as *maximum likelihood paths* or *instantons* that minimize

[3] A viscosity solution to a first-order nonlinear PDE $f(\boldsymbol{x}, u, \nabla u) = 0$ is a continuous but possibly nondifferentiable function obtained as the limit of a sequence of smooth solutions to $f(\boldsymbol{x}, u, \nabla u) = \epsilon \Delta u$ as $\epsilon \to \infty$.

the geometric action (9) [14; 3] (see Appendix A):

$$\psi'(s) = \frac{\boldsymbol{b}(\psi(s)) + \nabla U(\psi(s))}{\|\boldsymbol{b}(\psi(s)) + \nabla U(\psi(s))\|}. \tag{13}$$

Once the quasipotential is computed, one can shoot a MAP from a given point $\boldsymbol{x}$ back to the attractor A by integrating (13) backward in s. Alternatively, MAPs can be found by path-based methods [13; 38; 19; 18] that directly minimize the Freidlin–Wentzell action or the geometric action.

The mentioned asymptotic estimate for the invariant probability density within a level set of the quasipotential completely lying in the basin $\mathcal{B}(A)$ of A is [14]

$$\mu(\boldsymbol{x}) \asymp e^{-U(\boldsymbol{x})/\epsilon}, \quad \text{i.e.,} \quad \lim_{\epsilon\to 0}(-\epsilon \log \mu(\boldsymbol{x})) = U(\boldsymbol{x}). \tag{14}$$

The symbol $\asymp$ denotes the logarithmic equivalence clarified in (14). The expected escape time from $\mathcal{B}(A)$ can also be estimated up to exponential order [14]:

$$\mathbb{E}[\tau_{\mathcal{B}(A)}] \asymp e^{U(\boldsymbol{x}^*)/\epsilon}, \quad \text{where } U(\boldsymbol{x}^*) = \min_{\boldsymbol{x}\in\partial\mathcal{B}(A)} U(\boldsymbol{x}). \tag{15}$$

In some common special cases, a sharp estimate for the expected escape time can be obtained [2].

The term *transition state* is often encountered in chemical physics literature. Mostly it refers to a saddle lying on the manifold separating two basins of attraction. The dynamics of the Lorenz system are complicated, and basins of its attractors are tightly interlaced for $\rho \gtrsim 20$. To accommodate such situations, we will define the term *escape state*.

Definition 2.1. Consider a system evolving according to SDE (1). Let A be an attractor of the corresponding ODE (6). The escape state from A is the set of points minimizing the quasipotential with respect to A over the boundary of the basin of A.

The quasipotential at the escape state of A defines the expected escape time from the basin of A up to exponential order according to (15).

3. A brief overview of Lorenz'63

The Lorenz'63 system

$$\begin{aligned} \dot{x_1} &= \sigma(x_2 - x_1), \\ \dot{x_2} &= x_1(\rho - x_3) - x_2, \\ \dot{x_3} &= x_1 x_2 - \beta x_3 \end{aligned} \tag{16}$$

is one of the most fascinating and transformative ODE models proposed in the twentieth century. E. Lorenz [22] derived it from Saltzman's 2D cellular convection model [26] using a Fourier expansion and truncating the trigonometric series to include a total of three terms. He proved that the resulting system exhibits a new type of long-term behavior. All trajectories of (16) stay in a bounded region. For $\sigma = 10$, $\beta = \frac{8}{3}$, and $\rho = 28$, their ω-limit sets form an "infinite complex of surfaces", i.e., a fractal, whose Hausdorff dimension is 2.06 [33], later named the Lorenz attractor. The Lorenz map [22], a 1D map $z_{n+1} = f(z_n)$, where z_n is the n-th maximum of the z-component of a trajectory, and f is the function estimated numerically, explained the divergence of arbitrarily close characteristics. It has become instrumental for analysis of chaotic dynamical systems.

The study of the Lorenz'63 system burst in the mid-1970s, perhaps due to the progress in the computer industry. A number of remarkable properties and quantitative characteristics have been discovered. The topological structure of the Lorenz attractor was studied in [16; 25; 34]. The phenomenon called preturbulence was described in [21]. The value $\rho_1 \approx 24.06$ at which the Lorenz attractor is born for $\sigma = 10$ and $\beta = \frac{8}{3}$ was found in [36] using a functional fit to the Lorenz map. Homoclinic explosions, period-doubling cascades, and periodicity windows were investigated in [30]. A beautiful overview of the Lorenz system is given in [32, Chapters 9–12]. Nowadays, the Lorenz system is a popular test model for new methods in such fields as machine learning and forecasting (e.g., [12; 29; 17]).

It is easy to check that (16) is invariant under the symmetry transformation $(x_1, x_2, x_3) \mapsto (-x_1, -x_2, x_3)$. We fix the parameters $\sigma = 10$ and $\beta = \frac{8}{3}$ and consider the dynamics of (16) as ρ grows from zero to infinity. The notation and bifurcations important for the rest of the paper are summarized in Table 1. A more detailed description of the dynamics of (16) for $0 < \rho < \infty$ is given in Appendix B.

In this work, we consider the Lorenz system perturbed by small white noise (2). The noise term regularizes the chaotic deterministic dynamics of (16) in the sense that one can predict the future probability density function given the current one by solving the Fokker–Planck equation. On the other hand, the presence of the noise term enables escapes from any neighborhood of an attractor of (16). If ρ is such that there are multiple attractors, noise-induced transitions between their neighborhoods become possible.

4. Numerical methods

In this section, we describe numerical techniques developed for computing the quasipotential for highly dissipative and chaotic systems where the ratio of the magnitudes of the rotational and potential components is of the order of 10^3.

4A. ***A brief overview of ordered line integral methods (OLIMs).*** We start with a brief overview the OLIMs. A comprehensive description of the implementation of the OLIM in 3D is provided in [35]. It involves many technical details that are important for making the solver fast. A C source code `olim3D4Lorenz63.c` set up to compute the quasipotential for (2) and instructions on how to run it are available in [5; 4].

range of ρ	comments and notation
$0 < \rho < 1$	The origin is the unique globally attracting equilibrium.
$\rho = 1$	Supercritical pitchfork bifurcation.
$1 < \rho < \rho_0 \approx 13.926$	The origin is a Morse index-one saddle for $1 < \rho < \infty$. Equilibria $C_\pm$ are located at $C_\pm = (\pm\sqrt{\beta(\rho-1)}, \pm\sqrt{\beta(\rho-1)}, \rho-1).$ $C_\pm$ are asymptotically stable for $1 < \rho < \rho_2$.
$\rho = \rho_0 \approx 13.926$	Homoclinic orbits starting and ending at the origin exist.
$\rho_0 < \rho < \rho_1 \approx 24.06$	$C_\pm$ are surrounded by saddle cycles $\gamma_\pm$, respectively. Chaotic dynamics ("preturbulence") is developing as ρ grows. We introduce cones $\Upsilon_\pm$ with vertices at $C_\pm$ and passing through $\gamma_\pm$, respectively: $\Upsilon_+ := \{C_+ + t(\boldsymbol{x} - C_+) \mid t \ge 0,\ \boldsymbol{x} \in \gamma_+\}.$
$\rho = \rho_1 \approx 24.06$	The birth of the Lorenz attractor A_L (a strange attractor).
$\rho_1 < \rho < \rho_2 \approx 24.74$	A_L coexists with asymptotically stable equilibria $C_\pm$.
$\rho = \rho_2 \approx 24.74$	A subcritical Hopf bifurcation: $\gamma_\pm$ shrink to $C_\pm$, respectively.

Table 1. A summary of bifurcations and notation for Lorenz'63 (16) for $\sigma = 10$, $\beta = \frac{8}{3}$, and $0 < \rho \le \rho_2 \approx 24.74$.

The OLIMs belong to the family of label-setting algorithms [6] and inherit their set of labels from the OUM [27; 28]. Labels of mesh points indicate their statuses. A mesh point is Accepted if the value of the computed function (the quasipotential in our case) is finalized at it and all its nearest neighbors also have finalized values. Accepted points are not used for updating values at other mesh points. A mesh point is Accepted Front if the value at it is finalized but it has at least one nearest neighbor with an unfinalized value. Considered mesh points are those with unfinalized tentative values that have at least one Accepted Front nearest neighbor. Unknown mesh points have no Accepted Front nearest neighbors and the values at them have not been proposed yet.

The OLIMs use several kinds of neighborhoods of mesh points. The neighborhoods are defined via distances between indices of the mesh points. Let $\boldsymbol{p} := (i, j, k) \in \mathbb{Z}^3$ and $\boldsymbol{p}_0 := (i_0, j_0, k_0) \in \mathbb{Z}^3$ be the lattice points corresponding to the mesh points $\boldsymbol{x}$ and $\boldsymbol{x}_0$, respectively. In other words, $\boldsymbol{p}$ and $\boldsymbol{p}_0$ are the indices of the mesh points $\boldsymbol{x}$ and $\boldsymbol{x}_0$, respectively. Recall that the l_q, $q = 1, 2$, and l_∞ distances between $\boldsymbol{p}$ and $\boldsymbol{p}_0$ are defined as

$$\|\boldsymbol{p} - \boldsymbol{p}_0\|_q := [|i - i_0|^q + |j - j_0|^q + |k - k_0|^q]^{1/q},$$
$$\|\boldsymbol{p} - \boldsymbol{p}_0\|_\infty := \max\{|i - i_0|, |j - j_0|, |k - k_0|\},$$

respectively. Let $\mathcal{I}$ be the set of indices of all mesh points.

- The near neighborhood typically containing 26 points

  $$\mathcal{N}_{\text{near}}(\boldsymbol{p}_0) := \{\boldsymbol{p} \in \mathcal{I} \mid \|\boldsymbol{p} - \boldsymbol{p}_0\|_1 \le 3 \text{ and } \|\boldsymbol{p} - \boldsymbol{p}_0\|_\infty = 1\}$$

  is used for recruiting Unknown points to Considered and changing the status of Accepted Front points to Accepted. Correspondingly, the near neighborhood of the mesh point $\boldsymbol{x}_0$ is defined as

  $$\mathcal{N}_{\text{near}}(\boldsymbol{x}_0) := \{\boldsymbol{x} \mid \boldsymbol{p} \in \mathcal{N}_{\text{near}}(\boldsymbol{p}_0)\}.$$

- The far neighborhood $\mathcal{N}^K_{\text{far}}(\boldsymbol{p}_0)$, where K is the update factor (a positive integer chosen by the user), consists approximately[4] of all lattice points $\boldsymbol{p} \in \mathcal{I}$ such that $\boldsymbol{p} \ne \boldsymbol{p}_0$ and the l_2 distance $\|\boldsymbol{p} - \boldsymbol{p}_0\|_2 \le K$. It is used for updating Considered points. Correspondingly, the far neighborhood of the mesh point $\boldsymbol{x}_0$ is defined as

  $$\mathcal{N}^K_{\text{far}}(\boldsymbol{x}_0) := \{\boldsymbol{x} \mid \boldsymbol{p} \in \mathcal{N}^K_{\text{far}}(\boldsymbol{p}_0)\}.$$

  If the mesh steps in x_i, $i = 1, 2, 3$, are all equal to h, then the far neighborhood of $\boldsymbol{x}_0$ is approximately the ball centered at $\boldsymbol{x}_0$ of radius Kh.

At the start, all mesh points are Unknown. Initialization consists of computing tentative values at the mesh points lying near the attractor, switching their status to Considered, and adding them to the binary tree. The binary tree maintains the heap sort of the values at Considered points so that the smallest Considered value is always at the root of the tree. At each step of the main body of the OLIM, a Considered mesh point $\boldsymbol{x}_{\text{new}}$ with the smallest tentative value becomes Accepted Front. Then the hierarchical update procedure proposed in [11] and further developed in [35] is implemented. It consists of two substeps. First, for all Considered points in $\mathcal{N}^K_{\text{far}}(\boldsymbol{x}_{\text{new}})$ proposed update values involving $\boldsymbol{x}_{\text{new}}$ are computed. Second, each Unknown point $\boldsymbol{x}$ in $\mathcal{N}_{\text{near}}(\boldsymbol{x}_{\text{new}})$ becomes Considered and a tentative value at $\boldsymbol{x}$ is computed using the Accepted Front points in $\mathcal{N}^K_{\text{far}}(\boldsymbol{x})$. This algorithm is summarized in the pseudocode below. The details of each step are elaborated in [35].

Now we outline the hierarchical update strategy. All details of it are worked out in [35]. There are three types of updates done in the order

$$\text{one-point updates} \to \text{triangle updates} \to \text{simplex updates}.$$

Let $\boldsymbol{x}$ be a Considered point to be updated, and $\boldsymbol{y} \in \mathcal{N}^K_{\text{far}}(\boldsymbol{x})$ be Accepted Front.

[4] More precisely, $\boldsymbol{p} \in \mathcal{N}^K_{\text{far}}(\boldsymbol{p}_0)$ if and only if $\boldsymbol{p} \ne \boldsymbol{p}_0$, $\boldsymbol{p} \in \mathcal{I}$, and $|i - i_0| \le K$, $|j - j_0| \le \text{ceil}(\sqrt{K^2 - |i - i_0|^2})$, and $|k - k_0| \le \text{ceil}(\sqrt{K^2 - \min\{|i - i_0|^2 + |j - j_0|^2, K^2\}})$. Defined so, $\boldsymbol{p} \in \mathcal{N}^K_{\text{far}}(\boldsymbol{p}_0)$ is slightly larger than $\{\boldsymbol{p} \in \mathcal{I} \mid \boldsymbol{p} \ne \boldsymbol{p}_0,\ \|\boldsymbol{p} - \boldsymbol{p}_0\|_2 \le K\}$.

```
Initialization. Start with all mesh points being Unknown. Set values of $U$ at them to $\infty$.
Let $x^*$ be an asymptotically stable equilibrium located at a mesh point. Compute tentative
values of $U$ at the points $x \in \mathcal{N}_{\text{near}}(x^*)$ and change their status to Considered.
The main body.
while the boundary of the mesh has not been reached and the set of Considered points is not empty do
    1. Change the status of the Considered point $x_{\text{new}}$ with the smallest tentative value of $U$ to Accepted Front.
    2. Change the status of all Accepted Front points in $\mathcal{N}_{\text{near}}(x_{\text{new}})$ that no longer have Considered points in their $\mathcal{N}_{\text{near}}$-neighborhoods to Accepted.
    3. Update all Considered points $x \in \mathcal{N}_{\text{far}}^{K}(x_{\text{new}})$. The updates must involve $x_{\text{new}}$.
    4. Change the status of each Unknown point $x \in \mathcal{N}_{\text{near}}(x_{\text{new}})$ to Considered and update them using the Accepted Front points in $\mathcal{N}_{\text{far}}^{K}(x)$.
```

Algorithm 1. A coarse-grained pseudocode of the OLIM.

One-point update. We connect $\boldsymbol{x}$ and $\boldsymbol{y}$ with a line segment and approximate the geometric action (9) along it using the midpoint quadrature rule $\mathcal{Q}_M(\boldsymbol{y}, \boldsymbol{x})$. Then the proposed value of the quasipotential at $\boldsymbol{x}$ is

$$\mathsf{Q}_1(\boldsymbol{y}, \boldsymbol{x}) = U(\boldsymbol{y}) + \mathcal{Q}_M(\boldsymbol{y}, \boldsymbol{x}). \tag{17}$$

If $\mathsf{Q}_1(\boldsymbol{y}, \boldsymbol{x})$ is less than the current tentative value $U(\boldsymbol{x})$, we replace $U(\boldsymbol{x})$ with it. Otherwise, we leave $U(\boldsymbol{x})$ unchanged. Furthermore, we compare $\mathsf{Q}_1(\boldsymbol{y}, \boldsymbol{x})$ with the current minimizer of the one-point update at $\boldsymbol{x}$ and update it if $\mathsf{Q}_1(\boldsymbol{y}, \boldsymbol{x})$ is smaller. In step 3 of Algorithm 1, the only one-point update computed is $\mathsf{Q}_1(\boldsymbol{x}_{\text{new}}, \boldsymbol{x})$. In step 4, one-point updates are computed for all Accepted Front points $\boldsymbol{y} \in \mathcal{N}_{\text{far}}^{K}(\boldsymbol{x})$.

Triangle update. Triangle updates always involve the minimizer of the one-point update $\boldsymbol{x}_0$. The base of an admissible triangle is a line segment connecting $\boldsymbol{x}_0$ and an Accepted Front point $\boldsymbol{x}_1$ satisfying $\|\boldsymbol{p}_1 - \boldsymbol{p}_0\|_1 \le 2$ and $\|\boldsymbol{p}_1 - \boldsymbol{p}_0\|_\infty = 1$ where $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$ are the indices of $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$, respectively. The points on the line segment $[\boldsymbol{x}_0, \boldsymbol{x}_1]$ are parametrized by $\lambda \in [0, 1]$: $\boldsymbol{x}_\lambda := \boldsymbol{x}_0 + \lambda(\boldsymbol{x}_1 - \boldsymbol{x}_0)$. The values of U on $[\boldsymbol{x}_0, \boldsymbol{x}_1]$ are found by linear interpolation: $U(\boldsymbol{x}_\lambda) \equiv U_\lambda := U(\boldsymbol{x}_0) + \lambda(U(\boldsymbol{x}_1) - U(\boldsymbol{x}_0))$. Then the triangle update is done by solving the constrained minimization problem

$$\mathsf{Q}_2(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}) = \min_{\lambda \in [0,1]} \{U_\lambda + \mathcal{Q}_M(\boldsymbol{x}_\lambda, \boldsymbol{x})\} \tag{18}$$

and replacing the current tentative value $U(\boldsymbol{x})$ with the proposed value $\mathsf{Q}_2(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x})$ if and only if the latter is less than the former. This replacement may take place only if an interior point solution is found. Hence, we are interested in the solution to (18) only if the minimizer $\lambda^* \in (0, 1)$. Therefore, we take the derivative of the function being minimized in the right-hand side of (18), compare its signs at the endpoints, and proceed with solving the nonlinear equation only if the signs are different.

Simplex update. One of the vertices of the triangle at the base of an admissible simplex must be the minimizer of the one-point update $\boldsymbol{x}_0$, and one of its sides adjacent to $\boldsymbol{x}_0$, let's call it $[\boldsymbol{x}_0, \boldsymbol{x}_1]$, must be such that the constrained minimization problem (18) has given an inner-point solution $\lambda^* \in (0, 1)$. The third vertex of the base of an admissible simplex must be an Accepted Front point $\boldsymbol{x}_2$ such that l_∞ distances between the indices of $\boldsymbol{x}_0$, $\boldsymbol{x}_1$, and $\boldsymbol{x}_2$ are all 1, and at most one of the l_1 distances between their indices is 2, while the other ones are 1. The proposed value produced by the simplex update is the solution of the constrained minimization problem

$$\mathrm{Q}_3(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}) = \min_{\lambda\in[0,1]} \{U_\lambda + \mathcal{Q}_M(\boldsymbol{x}_\lambda, \boldsymbol{x})\}, \tag{19}$$

$$\begin{aligned} \text{where} \quad & \boldsymbol{x}_\lambda = \boldsymbol{x}_0 + \lambda_1(\boldsymbol{x}_1 - \boldsymbol{x}_0) + \lambda_2(\boldsymbol{x}_2 - \boldsymbol{x}_0), \\ & U_\lambda = U(\boldsymbol{x}_0) + \lambda_1(U(\boldsymbol{x}_1) - U(\boldsymbol{x}_0)) + \lambda_2(U(\boldsymbol{x}_2) - U(\boldsymbol{x}_0)), \\ \text{subject to} \quad & \lambda_1 \ge 0, \quad \lambda_2 \ge 0, \quad \lambda_1 + \lambda_2 \le 1. \end{aligned} \tag{20}$$

The warm start for solving (19) is the vector $\lambda := [\lambda^*, 0]$ where λ^* is the minimizer of (18). As we do it for the triangle update, we wish to quickly reject the simplex update if its minimizer is certainly lying on the boundary of the triangle (20). We use the Karush–Kuhn–Tucker (KKT) optimality conditions [24, Chapter 12] to do so. They boil down (see Appendix C) to checking whether

$$\frac{\partial}{\partial\lambda_2}(U_\lambda + \mathcal{Q}_M(\boldsymbol{x}_\lambda, \boldsymbol{x})) \ge 0. \tag{21}$$

If (21) holds, then $[\lambda^*, 0]$ is a local solution to (19), and hence, we reject the simplex update. Otherwise we proceed with numerical minimization using Newton's method. If an interior point solution is found, we replace the current tentative value $U(\boldsymbol{x})$ with $\mathrm{Q}_3(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x})$ provided that $\mathrm{Q}_3(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}) < U(\boldsymbol{x})$. Otherwise, $U(\boldsymbol{x})$ remains unchanged.

We remark that the computation of the quasipotential terminates as soon as a boundary mesh point becomes Accepted Front. This is important because the MAP that leaves the computational domain via this point might return to it, and it is crucial for an accurate computation of the quasipotential that the computation follows the MAPs.

4B. ***Challenges of computing the quasipotential for stochastic Lorenz'63.*** An important characteristic of the vector field in SDE (1) in a neighborhood of an

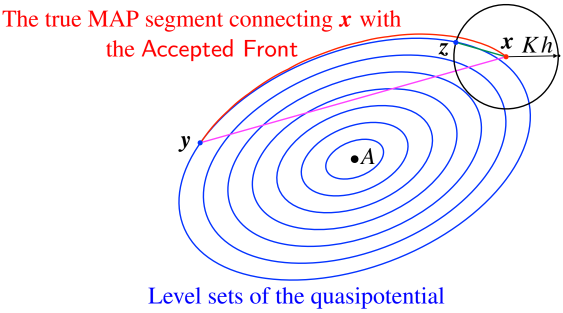

The true MAP segment connecting x with
the Accepted Front
z
x
Kh
y
A
Level sets of the quasipotential

Figure 1. An illustration for the difficulty of computing the quasipotential in the case where the ratio $\Xi(\boldsymbol{x})$ given by (22) is large. The blue closed curves represent some level sets of the quasipotential, and $\boldsymbol{x}$ is a Considered point that is up for an update. The green segment $[z, \boldsymbol{x}]$ is the best linear approximation to the MAP connecting $\boldsymbol{x}$ with the Accepted Front within the given update radius Kh.

attractor A is the ratio of the magnitude of the rotational component to that of the potential one [35]:

$$\Xi(\boldsymbol{x}) := \frac{\|\boldsymbol{l}(\boldsymbol{x})\|}{\left\|\frac{1}{2}\nabla U(\boldsymbol{x})\right\|}. \tag{22}$$

If $\Xi(\boldsymbol{x})$ is not too large (does not exceed 10) in the basin of A, except, perhaps in some small neighborhoods of the attractor or the escape state, the OLIMs give accurate results on uniform rectangular meshes of reasonable sizes [11; 10; 35]. However, if $\Xi(\boldsymbol{x})$ is large (much larger than 10) in a significant part of the basin of A, the accuracy of the numerical solution by the OLIM on a regular rectangular mesh deteriorates [11, §4]. The problem is illustrated in Figure 1. Suppose the computation has reached the level set of the quasipotential depicted with the largest closed blue curve. All mesh points inside it are either Accepted if they have no Unknown or Considered nearest neighbors, or Accepted Front, if they do. Let $\boldsymbol{x}$ be a Considered point up for an update. If $\Xi(\boldsymbol{x})$ is large, the segment of the MAP arriving at $\boldsymbol{x}$ from the span of Accepted Front mesh points is long. A rough estimate for its length is $\Xi(\boldsymbol{x})h$ where h is the mesh step. Let $\boldsymbol{y}$ be the point where this MAP segment starts at the span of the Accepted Front. Even if the update factor K were chosen large enough so that $\boldsymbol{y}$ lies in the ball centered at $\boldsymbol{x}$ of radius Kh, the straight line segment (the magenta line segment from $\boldsymbol{x}$ to $\boldsymbol{y}$ in Figure 1) and the midpoint quadrature rule would give poor approximations for the MAP segment and the geometric action along it, respectively, resulting in an inaccurate update value at $\boldsymbol{x}$. It is shown in [11; 35] that too large an update factor may deteriorate the accuracy. A safer but still too rough approximate solution would be obtained if the update radius is reasonably small, i.e., chosen according to the proposed rules of thumb in [11; 35]. Then the segment of MAP would be approximated with the green line segment $[z, \boldsymbol{x}]$ in Figure 1.

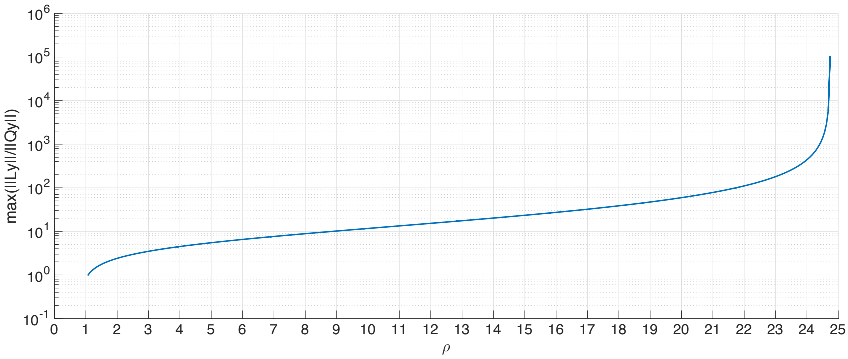


Figure 2. The graph of the maximal ratio Ξ of the magnitudes of the rotational and potential components of the linear SDE $d\boldsymbol{y} = J\boldsymbol{y}\,dt + \sqrt{\epsilon}\,d\boldsymbol{w}$ where J is the Jacobian matrix of the right-hand side of the Lorenz system (16) evaluated at the equilibrium C_+ for the range $1 < \rho < \rho_2 \approx 24.74$ where C_+ is asymptotically stable.

Now imagine the case where $\Xi(\boldsymbol{x}) \sim 10^3$ as it is for stochastic Lorenz’63 with $\rho_1 < \rho < \rho_2$ where the stable equilibria and the strange attractor coexist. 3D computations on regular rectangular meshes will give a qualitative idea about the geometry of the level sets of the quasipotential, but the found quasipotential barriers will be completely off.

The ratio $\Xi(\boldsymbol{x})$ for the Lorenz system at $1 < \rho < \rho_2 \approx 24.74$ can be estimated from that for the linearized system at C_+ (see Appendix D). The graph of Ξ for the linearized system is displayed in Figure 2. It shows that the maximum of $\Xi(\boldsymbol{x})$ blows up as $\rho \to \rho_2$. At $\rho = 24.4$, the largest ρ at which we present the results of our computations, the maximal value of $\Xi(\boldsymbol{x})$ for the linearized system is 973.4.

Challenged by this problem, we have developed an approach that allows us to obtain reasonably accurate values of the quasipotential barriers. It consists of finding approximate 2D manifolds (or unions of 2D manifolds) where the MAPs emanating from the attractor are located, building so-called radial meshes on them, and adjusting the OLIM for performing computations on radial meshes. This approach is suitable for any 3D SDE where the level sets of the quasipotential are thin, i.e., close to some 2D manifolds (see Assumption 4.2 below), which can be determined by visual inspection of the computed 3D level sets. Note that this is a safe diagnosis as the 3D OLIM tends to make the level sets thicker than the true ones if $\Xi(\boldsymbol{x})$ is large. In this case, the MAP going from the attractor to the escape state will be very close to any 2D manifold (or union of manifolds) approximating the level set containing the escape state. We find such a manifold using the characteristics of the corresponding ODE. The following lemma is instrumental for this approximation.

Lemma 4.1. *Let A be an attractor of $\dot{\boldsymbol{x}} = \boldsymbol{b}(\boldsymbol{x})$, where $\boldsymbol{b} \in C^1(\mathbb{R}^3)$. Let*

$$\mathcal{V}_a := \{\boldsymbol{x} \in \mathbb{R}^3 \mid U(\boldsymbol{x}) \le a\}$$

be a sublevel set of the quasipotential completely lying in the basin of A, and γ be a curve lying on the boundary of $\mathcal{V}_a$; i.e., for any $\boldsymbol{x} \in \gamma$, $U(\boldsymbol{x}) = a$. Let $\mathcal{M}'$ and $\mathcal{M}$ be the manifolds consisting, respectively, of the MAPs going from A to γ, and the characteristics starting at γ and running to A. Then $\mathcal{M}' \subset \mathcal{V}_a$ and $\mathcal{M} \subset \mathcal{V}_a$.

A proof of Lemma 4.1 can be found in Appendix E.

Let γ be an unstable limit cycle serving as the escape state from the basin of an attractor A. Let the quasipotential at γ be U_γ. We can consider a sublevel set $\mathcal{V}_a$ for $a < U_\gamma$ and arbitrarily close to U_γ. By Lipschitz continuity of the quasipotential [3], a can be chosen so that the distance between γ and $\mathcal{V}_a$ is smaller than any given positive number. Correspondingly, we can pick a curve γ' lying on the boundary of $\mathcal{V}_a$ located arbitrarily close to the limit cycle γ. By Lemma 4.1, the manifolds $\mathcal{M}'$ and $\mathcal{M}$ consisting of MAPs/characteristics running to/from γ' will lie in $\mathcal{V}_a$.

Assumption 4.2. Suppose that the level set $\mathcal{V}_a$ is close to both manifolds $\mathcal{M}$ and $\mathcal{M}'$, i.e., the Hausdorff distances[5] between $\mathcal{V}_a$ and $\mathcal{M}$ and between $\mathcal{V}_a$ and $\mathcal{M}'$ are less than some small $\delta > 0$:

$$d_H(\mathcal{V}_a, \mathcal{M}) < \delta \quad \text{and} \quad d_H(\mathcal{V}_a, \mathcal{M}') < \delta.$$

Under Assumption 4.2, the triangle inequality implies that the Hausdorff distance between $\mathcal{M}$ and $\mathcal{M}'$ is bounded by 2δ:

$$d_H(\mathcal{M}, \mathcal{M}') \le d_H(\mathcal{M}, \mathcal{V}_a) + d_H(\mathcal{M}', \mathcal{V}_a) < 2\delta. \tag{23}$$

We will employ Assumption 4.2 for $15 \le \rho \le 24.4$. Figures 7 and 9 below illustrate it: compare the MAPs (the dark red curves) and the characteristics (the dark blue curves) in these figures and observe that they lie on close manifolds located inside visibly thin level sets.

Note that the manifold $\mathcal{M}$ can be readily sampled by shooting characteristics from γ' to A. In the next section, we describe how to build radial meshes on $\mathcal{M}$, adjust the OLIM for them, and test its performance.

4C. ***Radial meshes on manifolds.*** We call a mesh *radial* if it is set up as follows. Let γ_0 be a point or a closed curve, and let γ be another closed curve. We pick a finite set of simple closed curves that do not intersect pairwise and index them γ_i, $i = 1, \dots, N_r - 2$. We add γ_0 and $\gamma_{N_r-1} \equiv \gamma$ to this set. These curves will be referred to as *parallels*. We also pick a finite set of curves, *meridians*, going from γ_0 to γ and crossing each γ_i exactly once in the order of increase of their indices. We index the meridians from 0 to $N_a - 1$ and identify meridian 0 with meridian N_a. The resulting mesh has size $N_r \times N_a$. Examples of radial meshes for the Lorenz system defined on manifolds consisting of all characteristics going from saddle

[5] $d_H(\mathcal{X}, \mathcal{Y}) = \max\{\sup_{\boldsymbol{x}\in\mathcal{X}} \inf_{\boldsymbol{y}\in\mathcal{Y}} \|\boldsymbol{x} - \boldsymbol{y}\|, \sup_{\boldsymbol{y}\in\mathcal{Y}} \inf_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{x} - \boldsymbol{y}\|\}$.

cycles to asymptotically stable equilibria at $\rho = 15$ and $\rho = 24.4$ are shown in Figures 8, top, and 13, left, respectively. A radial mesh defined between two closed curves, the saddle cycle γ_- and a closed curve approximating an "eye" of the strange attractor at $\rho = 24.4$, is displayed in Figure 14, top left. Our technique for building radial meshes is described in Appendix F and implemented in the Matlab code `make2Dmesh.m`.

To adjust the OLIM for radial meshes, we redefine the neighborhood $\mathcal{N}_{\text{far}}((i_r, i_a))$ from which a mesh point indexed (i_r, i_a) can be updated using two update factors, radial K_r and angular K_a, as follows: $\mathcal{N}_{\text{far}}((i_r, i_a))$ consists of all mesh points (j_r, j_a) satisfying

$$\max\{0, i_r - K_r\} \le j_r \le \min\{i_r + K_r, N_r - 1\},$$
$$|(j_a - i_a) \bmod N_a| \le K_a.$$

Let us check whether the OLIM applied to a system with large ratio Ξ produces small enough errors on 2D radial meshes of reasonable sizes and these errors properly decay with mesh refinement. We set up an ad hoc 2D example with an asymptotically stable spiral point at the origin and an unstable limit cycle $\|\boldsymbol{x}\| = 1$:

$$\begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix} = \begin{bmatrix} \|\boldsymbol{x}\|^2 - 1 & a \\ -a & \|\boldsymbol{x}\|^2 - 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} dt + \sqrt{\epsilon}\, d\boldsymbol{w}. \tag{24}$$

We pick $a = 10^3$; then $\Xi \ge 10^3$. The exact quasipotential for (24) with respect to the origin is given by

$$U(\boldsymbol{x}) = \begin{cases} \|\boldsymbol{x}\|^2(1 - 0.5\|\boldsymbol{x}\|^2), & \|\boldsymbol{x}\| \le 1, \\ 0.5, & \|\boldsymbol{x}\| > 1. \end{cases} \tag{25}$$

We have conducted two experiments with computing the quasipotential for (24). The goal of the first experiment is to establish the dependence of the numerical error on the relationship between N_r, N_a, K_r, and K_a. We set $N_r = 1024$ and run the solver for $N_a = 2^q N_r$, $q = 0, 1, 2, 3$, and K_r varying from 1 to $\texttt{round}(N_r/40) = 25$ and $K_a = 2^q K_r$, respectively. The computational domain is the unit circle. The dependence of the normalized maximal absolute error

$$E := \frac{\max_{i_r, i_a} |U(i_r, i_a) - U_{\text{exact}}(i_r, i_a)|}{\max_{i_r, i_a} U_{\text{exact}}(i_r, i_a)} \tag{26}$$

on K_r is shown in Figure 3, left. The normalized maximal absolute error (the red curve) for the 1024 × 1024 rectangular mesh defined on the square $[-1, 1]^2$ is also provided for comparison. These results eloquently demonstrate the superiority of the radial meshes for computing the quasipotential in the case where the ratio Ξ is large. Also, the choice $K_r = \texttt{round}(N_r/40)$ and $K_a = \texttt{round}(N_a/40)$ is reasonable and can be used as a default setting for radial meshes.

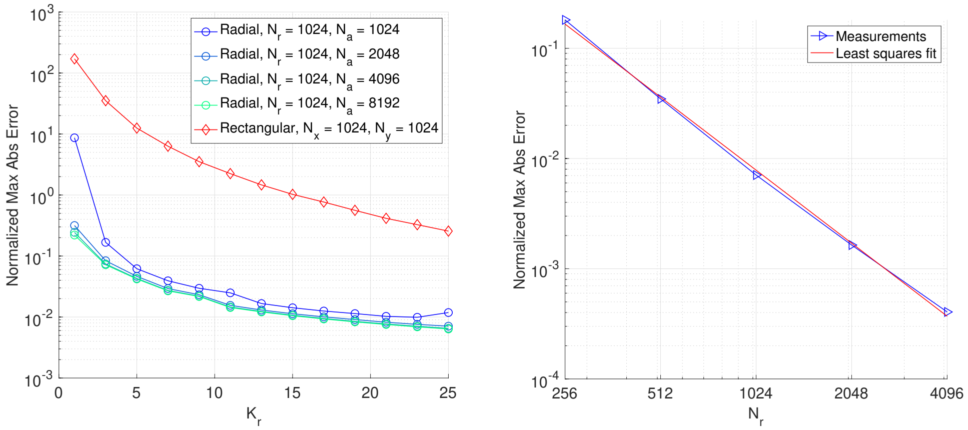


Figure 3. Measurements of numerical errors for radial meshes $N_r \times N_a$ in computing the quasipotential for SDE (24). Left: the dependence of the normalized maximal absolute error (26) on the update parameter K_r. The parameter K_a was chosen so that $N_a/N_r = K_a/K_r$. Right: the dependence of the normalized maximal absolute error (26) (the blue plot) on N_r with $N_a = 2N_r$, $K_r = \texttt{round}(N_r/40)$, and $K_a = 2K_r$. The least squares fit (27) is included for comparison.

The goal of the second experiment is to verify error decay with mesh refinement. We have run computations with $N_r = 2^p$, $p = 8, 9, 10, 11, 12$, $N_a = 2N_r$, $K_r = \texttt{round}(N_r/40)$, and $K_a = 2K_r$. The plot of the normalized maximal absolute error in Figure 3, right, shows the desired convergence. The least squares fit gives a superquadratic convergence:

$$E = 3.3 \cdot 10^4 \cdot N_r^{-2.2}. \tag{27}$$

The superiority of radial meshes over rectangular ones for the computation of the quasipotential in the basins of spiral point attractors of vector fields with large rotational components is due to the fact that the radial meshes have update regions better adjusted to the geometry of the MAPs than the rectangular ones. This phenomenon is illustrated in Figure 4. The update regions of radial meshes are small near the equilibrium where the MAP has high curvature and grow away from it where the MAP's curvature decreases. In contrast, the update regions of rectangular meshes remain uniform. As a result, they are too large near the equilibrium and not large enough away from it.

In summary, our experiments with SDE (24) with a stable spiral point, an unstable limit cycle, and $\Xi \ge 10^3$ have demonstrated that the computation of the quasipotential on radial meshes of moderate sizes gives accurate and reliable results.

Remark 4.3. We emphasize that we still use line segments in the OLIM on radial meshes to approximate MAP segments. We have explored a variant of OLIM where the minimizer for each local constraint minimization problem is sought on the set

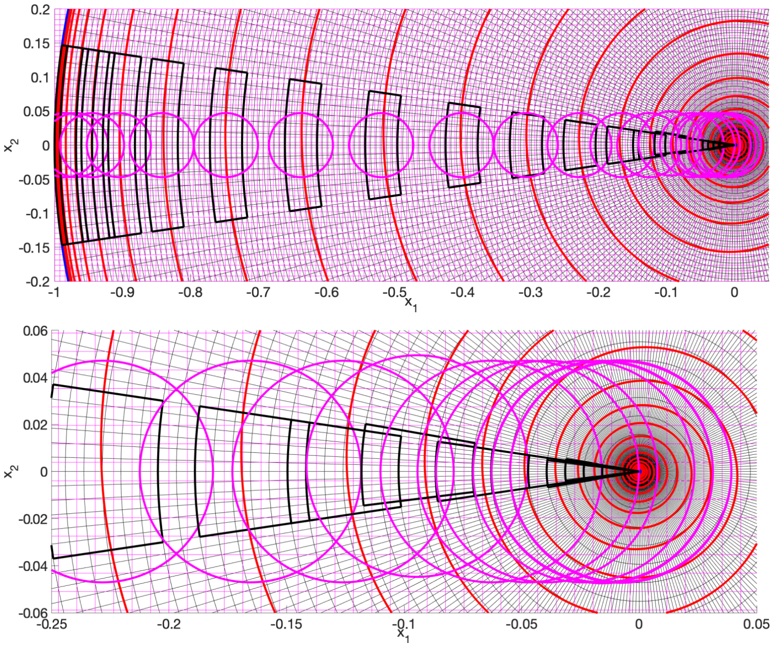


Figure 4. An illustration explaining the advantage of radial meshes over rectangular ones for the computation of the quasipotential on the example of SDE (24) with $a = 40$. Two computations were performed. The first one was done on the radial mesh with $N_r = 128$, $N_a = 256$, $K_r = 3$, and $K_a = 6$. The maximal absolute and RMS errors for this computations are $1.00 \cdot 10^{-2}$ and $2.44 \cdot 10^{-3}$, respectively. The second computation was performed on the rectangular mesh with $N = 256$ and $K = 6$ and gave the maximal absolute and RMS errors of $1.39 \cdot 10^{-1}$ and $6.43 \cdot 10^{-2}$, respectively, which are more than an order of magnitude larger than those for the radial mesh. The CPU times for the radial and rectangular meshes are approximately the same: 0.24 and 0.22 seconds, respectively, Top: the thick red curve is the exact MAP going from the equilibrium at the origin to the unstable limit cycle $r = 1$. The thin black mesh is the radial mesh. The thick black curves bound some samples of its update regions. The thin magenta mesh is the rectangular mesh, and the thick magenta circles are samples of its update regions. Bottom: a zoom-in of the top.

of curves of the form

$$\{(r(t), \theta(t)) \mid t \in [0, 1],\ r(t) = r_1 + t(r_2 - r_1),\ \theta(t) = \theta_1 + t(\theta_2 - \theta_1)\}$$

where (r_i, θ_i), $i = 1, 2$, are the polar coordinates of the endpoints of the curve. We have found that the use of line segments as in the original OLIM gives more accurate results, so we stick with line segments.

5. Results

In this section, we present a collection of plots of the level sets of the computed quasipotential in 3D for the Lorenz system at $\rho = 0.5$, 12, 15, 20, and 24.4. Where

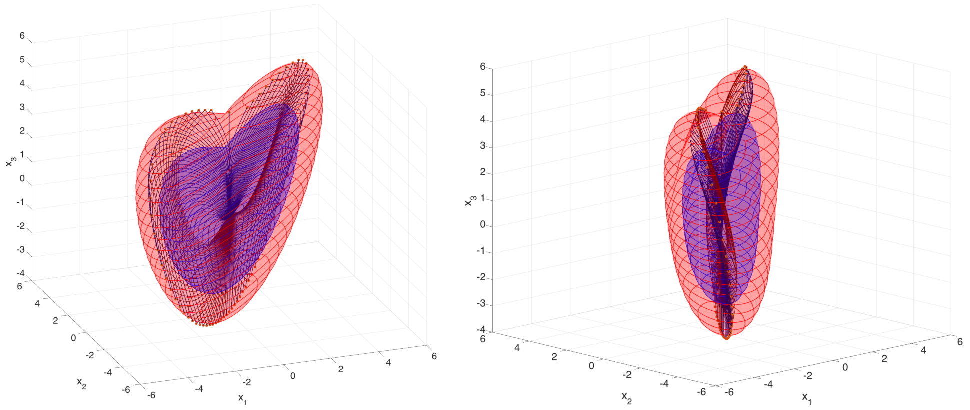


Figure 5. Two views of the level sets of the quasipotential at $\rho = 0.5$ corresponding to $U = 20$ (the blue surface) and $U = 40$ (the red surface). The thin blue and red closed curves lying on the corresponding level sets are shown to aid 3D visualization. The dark blue curves depict a collection of the characteristics starting at the set of points marked by large orange dots and approaching the origin. The dark red curves represent a collection of the MAPs emanating from the origin and arriving at the same set of points. A movie with this figure rotating around the x_3-axis is available at https://youtu.be/YscXN18lgyU.

appropriate, we perform 2D computations on radial meshes on manifolds and refine the estimates for the quasipotential barriers between different basins or regions of the phase space. Our collection of MAPs computed by integrating (13) backwards in s (code `ShootMAPs.c` [5]) can be compared with that obtained in [37] for a somewhat different set of values of ρ using the minimum action method (MAM). Note that, while the MAM is easier to program than the OLIM and is suitable for any phase-space dimension, its output is biased by the initial guess for the path and hence might converge to a local minimizer in the path space instead of the global one. Furthermore, MAM does not allow one to visualize the level sets of the quasipotential. Estimates for quasipotential barriers are not provided in [37] while we do it here.

5A. $0 < \rho < 1$. For $0 < \rho < 1$, the origin is globally attracting. Two level sets of the quasipotential for $\rho = 0.5$ are shown in Figure 5. The computation was performed on a $513 \times 513 \times 513$ mesh with the update factor $K = 14$. This choice of K for $N = 513$ was suggested in [35]. The level sets are heart-shaped and oriented approximately along the plane $x_1 = x_2$. Let X be a level set, and let γ_X be the intersection of X with the vertical plane $x_1 = x_2$. The curve γ_X runs approximately along the edge of the heart-shaped level set X. We pick X to be a level set corresponding to one of the largest computed values of the quasipotential and find a collection of points marked with large orange dots lying on the corresponding curve γ_X and forming angles from 0 to 2π with step $\pi/72$. The characteristics of (16) (the dark blue curves) and the MAPs of (2) (the dark red curves) starting and arriving at this set of points, respectively, are notably different. The set of characteristics starting at γ_X and the set of MAPs arriving at γ_X form visibly distinct 2D manifolds.

Let us find the directions along which typical characteristics and typical MAPs approach the origin and emanate from it, respectively. It is hard to see in Figure 5 whether they coincide or not. Let J be the Jacobian matrix of the right-hand side of (16) evaluated at the origin:

$$J = \begin{bmatrix} -\sigma & \sigma & 0 \\ \rho & -1 & 0 \\ 0 & 0 & -\beta \end{bmatrix}. \tag{28}$$

For the linear SDE

$$d\boldsymbol{x} = J\boldsymbol{x}\,dt + \sqrt{\epsilon}\,d\boldsymbol{w}, \tag{29}$$

the quasipotential decomposition is given by $J\boldsymbol{x} = -Q\boldsymbol{x} + L\boldsymbol{x}$ (see Appendix D), where Q and L are matrices. The quasipotential is the quadratic form $U(\boldsymbol{x}) = \boldsymbol{x}^\top Q\boldsymbol{x}$ where Q can be found analytically [3]:

$$Q = \begin{bmatrix} Q_1 & \\ & \beta \end{bmatrix}, \tag{30}$$

$$\text{where} \quad Q_1 = \frac{\sigma+1}{d}\begin{bmatrix} \sigma(\sigma+1)+\rho(\rho-\sigma) & -\rho-\sigma^2 \\ -\rho-\sigma^2 & (\sigma+1)-\sigma(\rho-\sigma) \end{bmatrix},$$

$$d = (\sigma+1)^2 + (\rho+\sigma)^2. \tag{31}$$

The rotational matrix $L = J + Q$ is

$$L = \begin{bmatrix} L_1 & \\ & 0 \end{bmatrix}, \tag{32}$$

$$\text{where} \quad L_1 = \frac{\rho-\sigma}{d}\begin{bmatrix} \rho+\sigma^2 & -(\sigma+1)+\sigma(\rho-\sigma) \\ \sigma(\sigma+1)+\rho(\rho-\sigma) & -\rho-\sigma^2 \end{bmatrix}.$$

For the linear SDE (29), MAPs are the characteristics of $\dot{\boldsymbol{x}} = (Q+L)\boldsymbol{x}$. Obtaining spectral decompositions of $J = -Q + L$ and $\tilde{J} = Q + L$ for $\rho = 0.5$, we find that typical characteristics of (16) approach the origin tangent to the line span $\boldsymbol{v}$, while typical MAPs emanate from the origin tangent to the line span $\tilde{\boldsymbol{v}}$, where

$$\boldsymbol{v} \approx \begin{bmatrix} 0.7241 \\ 0.6897 \\ 0 \end{bmatrix}, \quad \tilde{\boldsymbol{v}} \approx \begin{bmatrix} 0.6924 \\ 0.7215 \\ 0 \end{bmatrix}. \tag{33}$$

5B. $1 < \rho < \rho_0 \approx 13.926$. In this interval, the equilibria $C_\pm$ switch from stable nodes to stable spiral points at $\rho \approx 2.1546$. Figure 6 displays the level sets of the quasipotential for $\rho = 12$ with respect to each stable equilibrium. It was computed

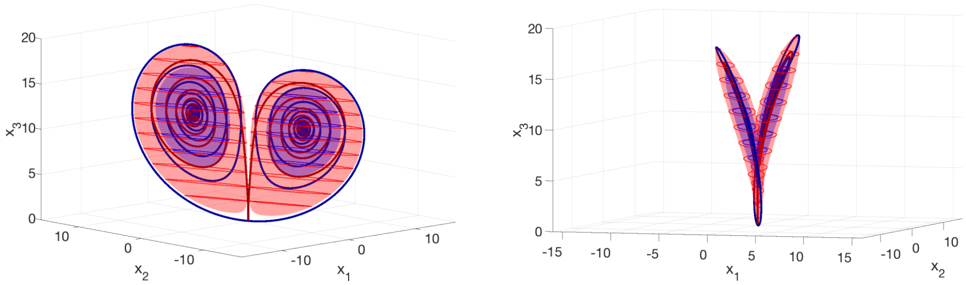


Figure 6. Two views of the level sets of the quasipotential at $\rho = 12$ corresponding to $U = 10$ (the blue surface) and $U = 19.42$ (the red surface). The dark blue curves are the characteristics emanating from the origin along its unstable directions $\pm\boldsymbol{\xi}$ (42) and arriving at $C_\pm$, respectively. The dark red curves are the MAPs going from $C_\pm$ to the origin. The MAP from $C_\pm$ to $C_\mp$ is obtained by the concatenation of the MAP from $C_\pm$ to the origin (a dark red curve) and the characteristic from the origin to $C_\mp$ (a dark blue curve). A movie with this figure rotating around the x_3-axis is available at https://youtu.be/-ABbuD8oDjI.

on a $513 \times 513 \times 513$ mesh with $K = 14$. The found value of the quasipotential at the origin that serves as the transition state between $C_\pm$ is 19.47. Therefore, at $\rho = 12$, the expected escape time from the basin of C_+ scales as

$$\mathbb{E}[\tau_{C_+}] \asymp e^{19.47/\epsilon}. \tag{34}$$

The MAP from C_+ to C_- is obtained by the concatenation of the computed MAP from C_+ to the origin (the dark red curve starting at C_+) and the characteristic from the origin to C_- (the dark blue curve ending at C_-). Figure 6, left, shows that the MAPs and the characteristics connecting $C_\pm$ and the origin lie on close 2D manifolds.

We did a consistency check by finding the quasipotential barrier by integrating the geometric action (9)–(10) along the found MAP and got the value 19.89, which is in reasonable agreement with 19.47 found by our 3D computation.

5C. $13.926 \approx \rho_0 < \rho < \rho_1 \approx 24.06$. In this range, the escape states from C_+ and C_- are the saddle limit cycles γ_+ and γ_-, respectively. We have computed the quasipotential for two values of ρ: $\rho = 15$ and $\rho = 20$.

5C1. $\rho = 15$. The computed quasipotential for $\rho = 15$ with respect to C_+ is visualized in Figure 7. First, we picked a large computational domain to embrace the level set of the quasipotential enclosing both of the stable equilibria $C_\pm$ and used a $613 \times 613 \times 613$ mesh and $K = 15$. Second, we chose a smaller domain just to enclose γ_+. It was a cube with side length 13 centered at C_+, and the mesh in it was $1001 \times 1001 \times 1001$. K was set to 20. The found quasipotential is nearly constant on γ_+: it varies between 17.42 and 17.45. The saddle cycles $\gamma_\pm$ are depicted with thick bright red curves. A maximum likelihood transition path from C_+ to C_- can

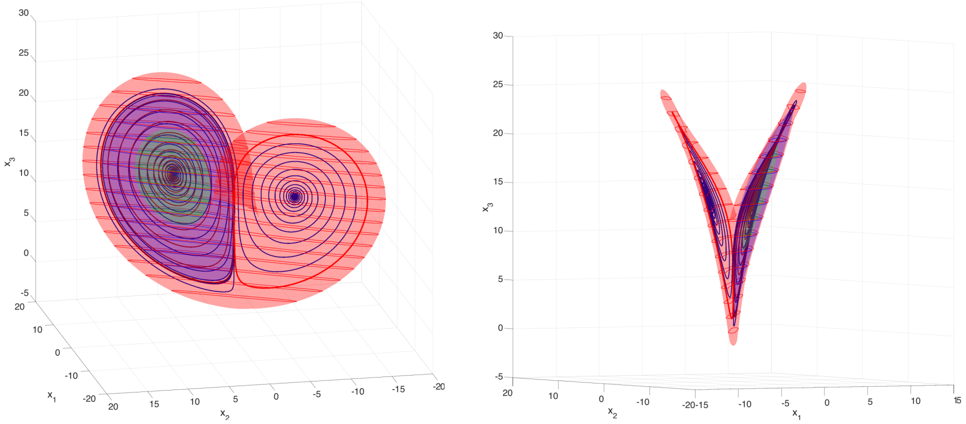


Figure 7. Two views of the level sets of the quasipotential at $\rho = 15$ corresponding to $U = 8$ (the green surface), $U = 17.37$ (the blue surface), and $U = 20$ (the red surface). The thick bright red curves are the saddle cycles $\gamma_{\pm}$. The dark blue curves are characteristics running from γ_+ and approaching $C_{\pm}$. The dark red curve is a MAP starting at C_+ and approaching γ_+. A movie with this figure rotating around the x_3-axis is available at https://youtu.be/mzdUD-ngqYs.

be obtained by the concatenation of a MAP from C_+ to γ_+, the saddle cycle γ_+, and a characteristic going from γ_+ to C_-. One such MAP and one such characteristic are the dark red and dark blue curves in Figure 7, respectively.

Willing to refine our relatively rough 3D computation and find a more accurate value of the quasipotential on γ_+ with respect to C_+, we perform 2D computations on the manifold $\mathcal{M}_+$ consisting of all characteristics going from γ_+ to C_+ using the code `olim2DEquilibLimitCycle.c`. Figure 7 suggests that $\mathcal{M}_+$ is close to the 2D manifold consisting of all MAPs from C_+ to γ_+. So we neglect the discrepancy between them. We generate 2D radial meshes on $\mathcal{M}_+$ (see Appendix F) whose coarsened version is shown in Figure 8, left. The computed quasipotential on $\mathcal{M}_+$ is shown in Figure 8, right. We first ran the OLIM on a radial mesh of size 2001×7200 and then repeated the computation on a refined mesh of size 4001×14400. The radial update factors K_r were 50 and 100, respectively, and the angular update factors K_a were 180 and 360, respectively. For the coarser mesh, the resulting values of the quasipotential on γ_+ varied from 18.19488 to 18.19501, averaging 18.19495. For the finer mesh, these numbers were, respectively, 18.19536, 18.19541, and 18.19536. These results suggest the following estimate for expected escape time from C_+ at $\rho = 15$:

$$\mathbb{E}[\tau_{C_+}] \asymp e^{18.2/\epsilon}. \tag{35}$$

For comparison and a consistency check, we have also found the quasipotential barrier by integrating the geometric action along the MAP going from C_+ to γ_+.

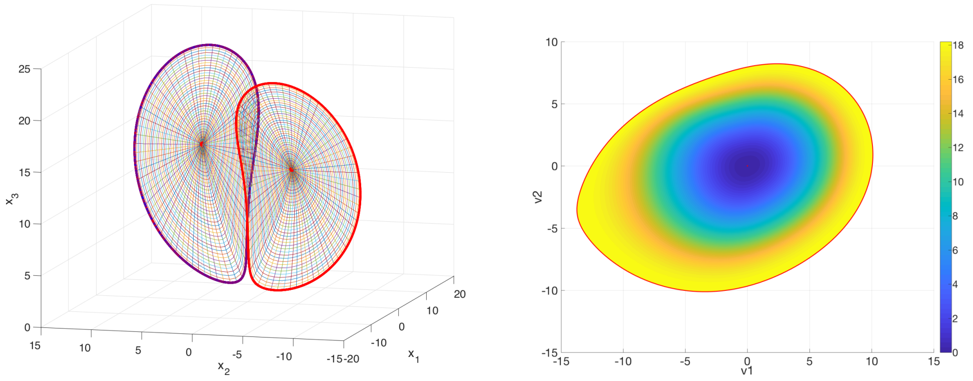


Figure 8. Left: radial meshes on the manifold $\mathcal{M}_\pm$ consisting of all characteristics going from the saddle cycles γ_+ (the thick purple curve) and γ_- (the thick red curve) to the equilibria $C_\pm$ (the large red dots), respectively. Right: the quasipotential computed on $\mathcal{M}_+$.

Note that the length of this MAP is infinite. However, the contribution to the geometric action from the integration along its infinite piece lying within a δ-tube around γ_+ tends to zero as $\delta \to 0$ as the quasipotential is Lipschitz-continuous [3]. Therefore, it suffices to take a finite piece of the MAP starting at C_+ and ending near γ_+. We took a piece of MAP of length 308.7 and obtained the value of the quasipotential barrier 19.3, which is closer to 18.2 as found by the 2D computation rather than to 17.4 as found by the 3D one. The result 19.3 is affected by numerical errors in the MAP and by the quadrature error amplified by the large length of the MAP. As ρ increases to $\rho_2 \approx 24.74$, the MAP spirals denser and denser, and integration of the geometric action along it becomes less and less accurate. So we abandon this consistency check for values of ρ larger than 15.

5C2. $\rho = 20$. For $\rho = 20$, we performed a computation in the cube with side length 26 centered at C_+ on a $1001 \times 1001 \times 1001$ mesh with $K = 20$. This cube encloses γ_+. The values of the computed quasipotential on γ_+ range from 6.59 to 6.62 and average 6.61. The level sets corresponding to $U = 3.3$ and $U = 6.58$ are shown in Figure 9. A 2D computation on the manifold $\mathcal{M}_+$ similar to the one described in Section 5C1 gave $U(\gamma_+) \in [6.1172, 6.1175]$ with the average at 6.1172. The MAP going from C_+ to γ_+ as well as the characteristics going from γ_+ to C_+ spiral notably denser than their counterparts at $\rho = 15$, and the level sets of the quasipotential are thinner. The saddle cycles are the escape states from the basins of $C_\pm$ to a chaotic region [21] where it is hard to predict for a characteristic which attractor, C_+ or C_-, it will eventually approach. We traced 1000 trajectories starting on the cone Υ_+ (see Table 1) at the points of the form $\boldsymbol{y}_i := \boldsymbol{x}_i + 0.002(\boldsymbol{x}_i - C_+)$ where $\boldsymbol{x}_i \in \gamma_+$, $i = 1, \dots, 1000$, are equispaced, and recorded whether they converged to C_+ or C_- as $t \to \infty$: 508 and 492 trajectories

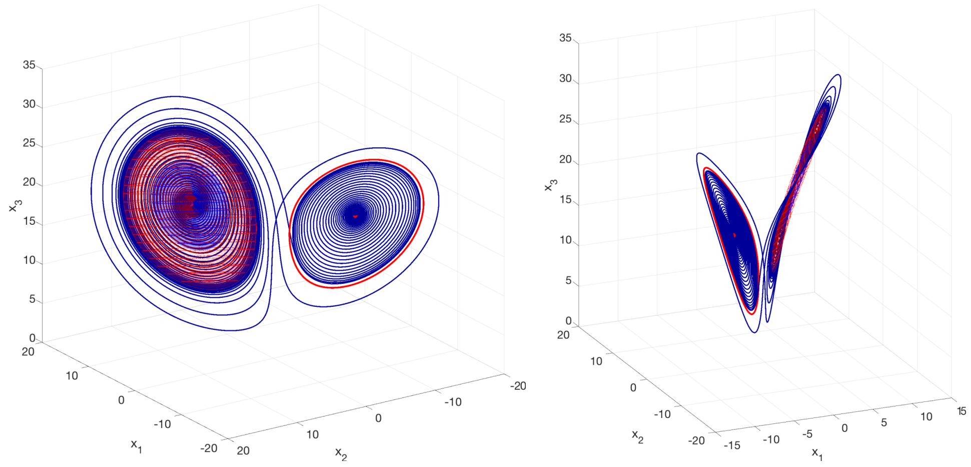


Figure 9. Two views of the level sets of the quasipotential at $\rho = 20$ corresponding to $U = 3.3$ (the blue surface), and $U = 6.58$ (the red surface). The thick bright red curves are the saddle cycles $\gamma_{\pm}$. The dark blue curves are characteristics going from γ_{+} to C_{+} and C_{-}. The dark red curve is a MAP starting at C_{+} and approaching γ_{+}. A movie with this figure rotating around the x_3-axis is available at https://youtu.be/JhBU0-dnos8.

converged to C_{+} and C_{-}, respectively. Then we subdivided γ_{+} into 100 intervals of equal length and used the recorded data to estimate the probability for a trajectory starting at each $\boldsymbol{y}_i$ corresponding to $\boldsymbol{x}_i$ in each interval to converge to C_{+}. The result is shown in Figure 10, left. The probabilities for γ_{-} are obtained by symmetry. Note that a similar calculation for $\rho = 15$ gave the probability distribution depicted in Figure 10, right: 975 out of 1000 trajectories starting at the analogous points of the cone Υ_{+} eventually approached C_{-}, while 25 returned to C_{+}. The uncertainty for where the trajectory of (2) that escapes all level sets of the quasipotential not containing the saddle cycle will eventually go, to C_{+} or to C_{-}, appears where the saddle cycles $\gamma_{\pm}$ come close to each other.

Summarizing our findings for $\rho = 20$, we predict that the expected escape time from $C_{\pm}$ to the chaotic region scales as

$$\mathbb{E}[\tau_{C_{+}}] \asymp e^{6.1/\epsilon}. \tag{36}$$

5D. $24.06 \approx \rho_1 < \rho < \rho_2 \approx 24.74$. It was recognized by Lorenz [22] that the strange attractor is an “infinite complex of surfaces”, i.e., a fractal, which is a very complicated geometric object. The addition of small white noise to the Lorenz system regularizes and simplifies its dynamics in the sense that it renders the fine structure of the Lorenz attractor irrelevant and allows for a description of the dynamics in terms of probability measures. Taking this into account, we approximate the strange attractor A_L with a union of four manifolds as shown in Figure 11.

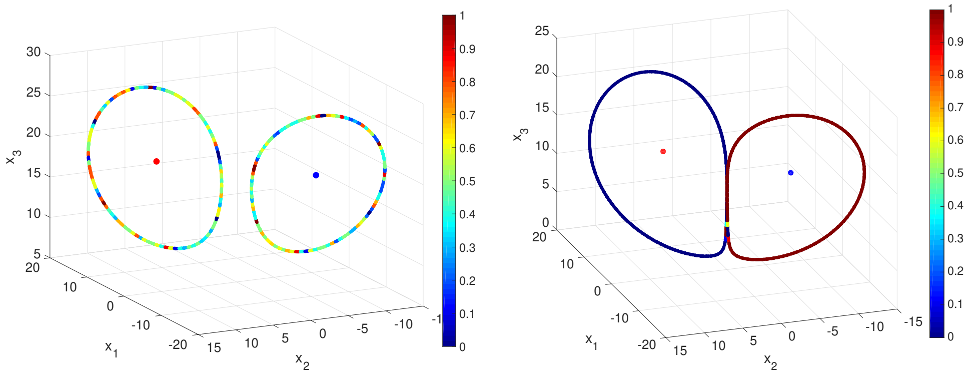


Figure 10. The probability for a trajectory starting on the cones $\Upsilon_{\pm}$ at the point of the form $\boldsymbol{x} + 0.002(\boldsymbol{x} - C_{\pm})$ where $\boldsymbol{x} \in \gamma_{\pm}$, respectively, to converge to C_{+}. Left: $\rho = 20$. Right: $\rho = 15$.

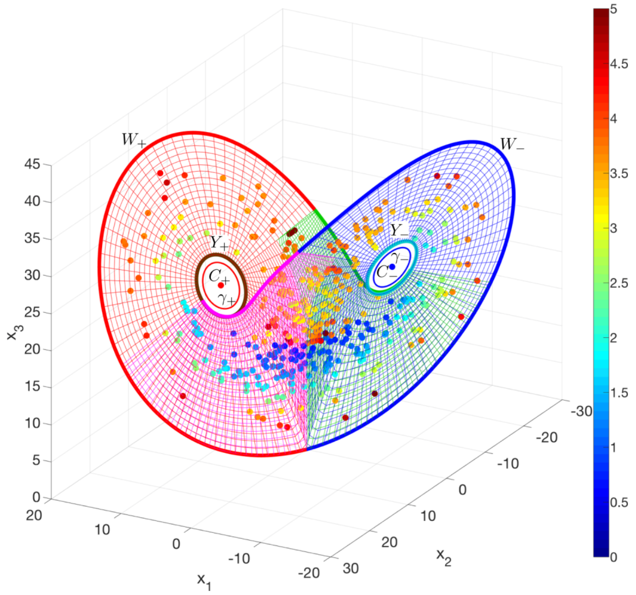


Figure 11. The strange attractor A_L at $\rho = 24.4$ is approximated by a union of four manifolds: red, magenta, blue, and green. The color of the large dots on the manifolds indicate the thickness of the fractal (the Lorenz attractor) at the corresponding locations. The colorbar corresponds to $-\log_{10} w(\boldsymbol{x})$ where $w(\boldsymbol{x})$ is the thickness of the fractal near the location $\boldsymbol{x}$. Hence, dark blue dots indicate thickness $\sim 10^{-1}$, light blue ones $\sim 10^{-2}$, yellow ones $\sim 10^{-3}$, orange ones $\sim 10^{-4}$, and red ones $\sim 10^{-5}$.

These manifolds were obtained using the code `StrangeAttractorMesh.m` in a way similar to the one described in Appendix F. The key component of this construction is finding a trajectory going into the saddle at the origin. We will refer to the inner

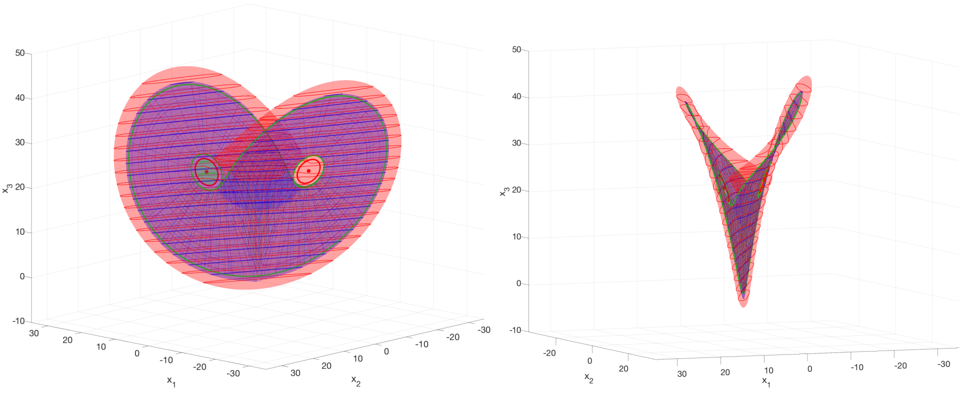


Figure 12. $\rho = 24.4$. Two views of the level sets of the quasipotential computed with respect to C_+. The green surface corresponds to the quasipotential value slightly less than the one at γ_+. The blue and red ones correspond to $U = 2$ and $U = 20$, respectively. The strange attractor is depicted with a mesh visible inside the blue and red surfaces. A movie with this figure rotating around the x_3-axis is available at https://youtu.be/ELqkeb8M1fg.

boundaries of the red and blue manifolds plotted with brown and cyan, respectively, as the eyes Y_+ and Y_-. The union of the red and green boundaries will be called wing W_+. Similarly, the union of the blue and magenta boundaries forms the wing W_-. In order to understand what the minimal reasonable value of the parameter ϵ in (2) that makes such an approximation sensible is, we have estimated the thickness of the strange attractor at 398 randomly picked points. Details are provided in Appendix G. The thickness map in Figure 11 indicates that the thickness of A_L does not exceed 10^{-2} wherever it is approximated by a single manifold. Larger values of thickness are found in places where we approximate A_L with two close manifolds. Hence, they are just an artifact of our thickness measurement method. The thickness map suggests that $\sqrt{\epsilon}$ in SDE (2) should be at least 10^{-2}, i.e., $\epsilon \gtrsim 10^{-4}$.

We performed a 3D computation of the quasipotential with respect to C_+ aiming at obtaining the overall picture. The computational domain was a box centered at C_+ and embracing the strange attractor. Note that this computation is too rough to give accurate numbers; nevertheless, it captures the geometry of the level sets. The level sets of the computed quasipotential shown in Figure 12 agree with our expectations: the quasipotential grows until it reaches the strange attractor, remains nearly constant on it, and then grows fast away from it, mainly along the union of manifolds that extends the strange attractor. Again, we performed a 2D computation on the manifold $\mathcal{M}_+$ on a radial 6001×7200 mesh with $K_r = 150$ and $K_a = 500$ and found the quasipotential at γ_+ to be equal to 0.03466 (see Figure 13). For comparison, the 3D computation performed in a cube with size 6 centered at C_+ on a $1001 \times 1001 \times 1001$ mesh with $K = 20$ gave the quasipotential on γ_+ around 0.25, which is more than 7 times larger due to the issues illustrated in Figure 1. This shows

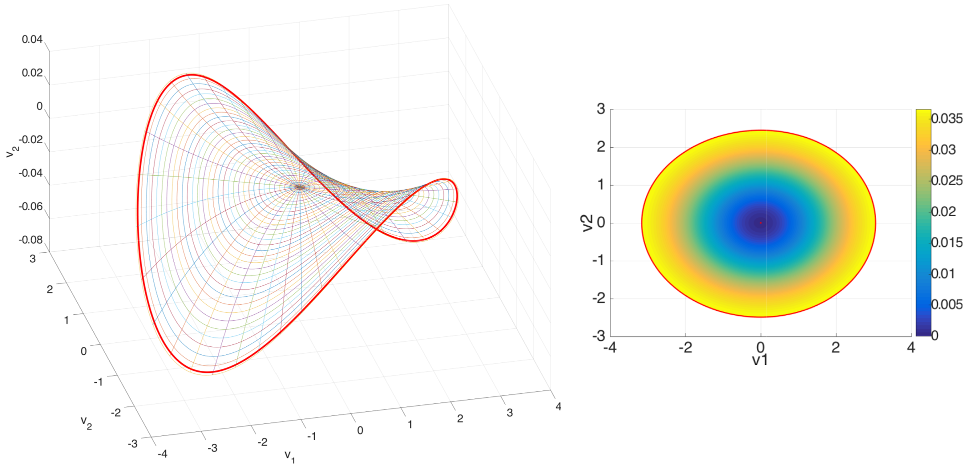


Figure 13. Left: a coarsened radial mesh on the manifold $\mathcal{M}_+$ at $\rho = 24.4$. The coordinate system is associated with the directions of eigenvectors of the quasipotential matrix for the Jacobian evaluated at C_+. Right: the quasipotential computed on this mesh.

that our reduction to 2D is very important for obtaining accurate quasipotential barriers.

Figure 11 shows that the quasipotential level sets primarily grow along the edge of the strange attractor while remaining quite thin. This observation suggests two possible transition mechanisms from the strange attractor to C_+. The first one would start near the eye Y_+, climb up to γ_+, and then switch to spiraling toward C_+. The second one would involve sliding toward γ_+ from the neighborhood of the wing W_- to a region lying between the eye and γ_+ and starting spiraling toward γ_+ and then toward C_+. Note that a MAP for the second mechanism at $\rho = 24.08$ was found in [37]. Coarsened versions of meshes generated for computing the quasipotential barriers for each of these transition mechanisms are displayed in Figure 14, top left and center, respectively. The "eye" mesh in Figure 14, top left, is lying on the unstable loop-shaped manifold of γ_+ between the γ_+ and Y_+. Its size is 1501×6000. The found quasipotential on γ_+ is 0.01543 (see Figure 14, top right). The "wing + eye" mesh in Figure 14, center, is defined on the union of the following two manifolds. The wing manifold is defined by trajectories starting near the negative x_3-semiaxis and bounded by W_+ and a trajectory approaching γ_+. The second one is the loop-shaped unstable manifold of γ_+ located between γ_+ and Y_+. The total mesh size is 1501×26001, of which a 1501×6000 piece covers the loop. The quasipotential computed on it is shown in Figure 14, bottom. Its part corresponding to the loop, naturally, involves significantly smaller values than the one corresponding to the strip around the wing. The quasipotential value on γ_+ for this mesh is 0.01479, which is smaller than the one for the eye mesh.

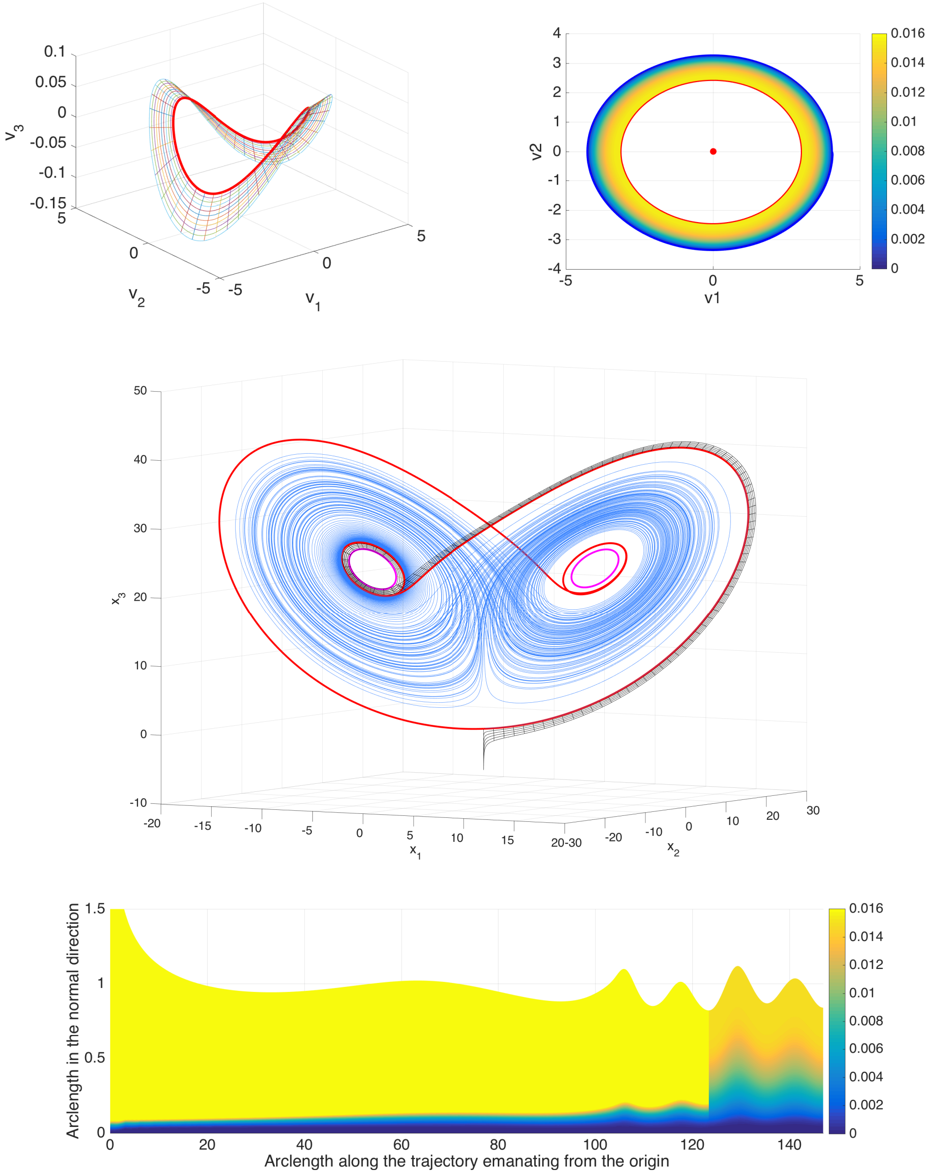


Figure 14. $\rho = 24.4$. Top left: a coarsened version of the“eye” mesh. The coordinate axes $\boldsymbol{v}_i$, $i = 1, 2, 3$, are chosen along the eigenvectors of the quasipotential matrix Q of the linearized near C_+ vector field. Top right: the quasipotential computed on the “eye” mesh. Center: a coarsened version of the “wing + eye” mesh. Bottom: the quasipotential computed on the “wing + eye” mesh. The arclength values less and greater than approximately 125 correspond to the “wing” and “eye” meshes, respectively. The discontinuity along the line where these meshes are glued is caused by the behavior of MAPs. The lightest yellow region of the plot corresponds to values of the quasipotential exceeding the maximal value 0.016 on the colorbar.

ρ	attractor	escape state	barrier
12	C_+	the origin	19.5
15	C_+	γ_+	18.2
20	C_+	γ_+	6.1
24.4	C_+	γ_+	0.0247
24.4	A_L	γ_+	0.0154 (“eye”)
24.4	A_L	γ_+	0.0148 (“wing + eye”)

Table 2. Quasipotential barriers for stochastic Lorenz’63 (2) at $\sigma = 10$, $\beta = \frac{8}{3}$, and a set of values of ρ.

As we have mentioned above, the strange attractor has a finite width varying roughly from 0 to 10^{-2}. This means that, in order to treat it as a union of four manifolds as shown in Figure 11 while considering the dynamics according to SDE (2), the parameter ϵ should be chosen at least as large as 10^{-4}. The discussed transition mechanisms from A_L to $C_\pm$ are associated with close quasipotential barriers: the difference between them is about $5 \cdot 10^{-4}$. Therefore, in order to determine which transition mechanism is dominant for $\epsilon \sim 10^{-4}$, one needs to compute the preexponential factors of the corresponding transition rates. Estimation of these prefactors is beyond the scope of the present work. We leave the development of numerical methods for their evaluation for the future.

We summarize the found quasipotential barriers in Table 2.

5E. ***Perspectives and challenges for large ρ.*** Our numerical experiments show that the level sets of the quasipotential thin out and the diameter of the strange attractor increases as ρ grows (Figure 15). On one hand, this creates an underresolution problem for 3D computations as mesh planes cannot be aligned with the level sets of the quasipotential because they are not flat. Handling this issue by means of mesh refinement is limited by the computer’s memory. For example, for $\rho = 100.75$ where two attracting limit cycles exist, the minimal level set of the quasipotential computed with respect to one of these cycles and enclosing the other one is thinner than the mesh step at some places.

On the other hand, thinning out of the level sets allows us to use 2D computations provided that we have an insight about possible transition mechanisms as we have had for $\rho = 24.4$. This insight for larger values of ρ can be gained from a 3D computation conducted not in a box but on a specially designed mesh.

6. Conclusions

We have developed a methodology for computing the quasipotential and finding quasipotential barriers for highly dissipative and possibly chaotic 3D dynamical systems perturbed by small white noise. The proposed approach combines 3D computations on regular rectangular meshes with, if relevant, dimensional reduction techniques and 2D computations on radial meshes. This methodology has been developed on and applied to stochastic Lorenz'63 with $\sigma = 10$, $\beta = \frac{8}{3}$, and a number of values of ρ ranging from 0.5 to 24.4.

We have shown that, as ρ increases, the level sets of the quasipotential thin out and the ratio of magnitudes of the rotational and potential components grows dramatically. On one hand, these facts render the numbers produced by 3D computations progressively less accurate. On the other hand, the manifolds consisting of characteristics going from escape states to attractors and those consisting of MAPs running the other way around become very close to each other. This observation motivated us to approximate the manifolds formed by the MAPs with those consisting of the characteristics.

We have developed a technique for generating radial meshes on manifolds consisting of such characteristics and tested our 2D OLIM quasipotential solver on an ad hoc system where the magnitude of the rotational component exceeds that of the potential one by a factor at least as large as 10^3, approximately as it is for $\rho = 24.4$ in (2). The least squares fit for this example has given a superquadratic convergence and small normalized maximal absolute errors on practical mesh sizes.

Using a combination of 3D and 2D computations, we found quasipotential barriers for the escapes from the basins of $C_\pm$ at $\rho = 12$, 15, 20, and 24.4. Furthermore, we estimated quasipotential barriers for the escape from the basin of the Lorenz attractor at $\rho = 24.4$ via two escape mechanisms. These barriers for 24.4 are close to each other: the difference between them is of the same order of magnitude as the minimal value of ϵ that makes traversing between different sheets of the Lorenz attractor easy. Therefore, estimates for the preexponential factors for these escape rates are necessary in order to determine which transition mechanism is dominant. We have left the development of techniques for computing these prefactors for the future.

An important advantage of computing the quasipotential in 3D is that it allows us to visualize the stochastic dynamics. Plots of quasipotential level sets reveal the hierarchy of regions of the phase space reachable by the system perturbed by small white noise on different timescales. In particular, the visualization of the level sets of the quasipotential at $\rho = 24.4$ suggested we consider and compare two possible transition mechanisms between the strange attractor and the stable equilibria.

Our C and Matlab programs developed for the application to Lorenz'63 are posted on M. Cameron's web site [5] (see the package `Qpot4Lorenz63.zip`) and on GitHub [4].

The numerical techniques developed in this work can be used for the quasipotential analysis of certain classes of other 2D and 3D SDEs. The dimensional reduction to 2D can be beneficial for any 3D SDEs where the quasipotential with respect to an attractor grows primarily along some 2D manifold. The use of radial meshes can dramatically improve the accuracy of found quasipotential thresholds in the case if the attractor is a stable spiral point and, perhaps, the transition state is an unstable limit cycle.

The application to the Lorenz’63 model allows us to see the limitations of the 3D quasipotential solver: the growth of required computational domains together with thinning out of the level sets results in underresolving the latter even with the use of 1001^3 mesh sizes. This motivates the directions of the future research associated with (i) combining the 3D OLIMs with techniques for generating a 3D mesh adapted for the geometry of the problem and (ii) advancing the techniques for learning 2D manifolds near which the stochastic dynamics are effectively focused.

Appendix A: Derivation of some equations in Section 2

***The geometric action* (9).** Let $\phi : [T_0, T_1] \to \mathbb{R}^d$ be a path with the endpoints $\phi(T_0) \in A$ and $\phi(T_1) = \boldsymbol{x}$. Expanding the squared norm in (8) and using the inequality

$$\|\dot{\phi}\|^2 + \|\boldsymbol{b}(\phi)\|^2 \geq 2\|\dot{\phi}\|\,\|\boldsymbol{b}(\phi)\|,$$

we obtain

$$S_{T_0,T_1}(\phi) \geq \int_{T_0}^{T_1} (\|\boldsymbol{b}(\phi)\|\,\|\dot{\phi}\| - \boldsymbol{b}(\phi)\cdot\dot{\phi})\,dt. \tag{37}$$

The equality holds if and only if $\|\boldsymbol{b}(\phi)\| = \|\dot{\phi}\|$. Since we are taking the infimum of $S_{T_0,T_1}(\phi)$ in particular with respect to T_0 and T_1, we choose the parametrization of ϕ so that $\|\boldsymbol{b}(\phi)\| = \|\dot{\phi}\|$ and change T_0 and T_1 accordingly. Note that T_0 and T_1 are allowed to be $-\infty$ and $+\infty$, respectively. Next, we observe that the integral in the right-hand side of (37) is invariant under reparametrization of the path ϕ. We denote the path ϕ reparametrized by its arclength by ψ and obtain (9).

***The Hamilton–Jacobi equation* (11) *for the quasipotential and* (13) *for the MAP*.** Let the path ψ parametrized according to its arclength (i.e., $\|\psi'\| = 1$) be the minimizer of the geometric action (9) among all absolutely continuous paths with one endpoint at $\boldsymbol{x}$ and the other one at A. Let us pick a small number $\delta > 0$. Using Bellman’s optimality principle [1] and Taylor expansion of U, we obtain

$$\begin{aligned} U(\boldsymbol{x}) &= \inf_{\|\psi'\|=1}\left\{\int_0^{\delta}(\|\boldsymbol{b}(\psi)\| - \boldsymbol{b}(\psi)\cdot\psi')\,ds + U\left(\boldsymbol{x} - \int_0^{\delta}\psi'\,ds\right)\right\} \\ &= \inf_{\|\psi'\|=1}\{\delta(\|\boldsymbol{b}(\psi)\| - \boldsymbol{b}(\psi)\cdot\psi' - \nabla U(\boldsymbol{x})\cdot\psi') + U(\boldsymbol{x}) + O(\delta^2)\}. \end{aligned}$$

Canceling $U(\boldsymbol{x})$ on both sides and dividing by δ we get

$$0 = \inf_{\|\psi'\|=1}\{\|\boldsymbol{b}(\psi)\| - \boldsymbol{b}(\psi)\cdot\psi' - \nabla U(\boldsymbol{x})\cdot\psi' + O(\delta)\}.$$

Taking the limit as $\delta \to 0$, we obtain

$$\inf_{\|\psi'\|=1} \{\|\boldsymbol{b}(\boldsymbol{x})\| - (\boldsymbol{b}(\boldsymbol{x}) + \nabla U(\boldsymbol{x})) \cdot \psi'\} = 0. \tag{38}$$

The infimum is attained when the term $(\boldsymbol{b}(\boldsymbol{x}) + \nabla U(\boldsymbol{x})) \cdot \psi'$ is maximal, i.e., when

$$\psi' = \frac{\boldsymbol{b}(\boldsymbol{x}) + \nabla U(\boldsymbol{x})}{\|\boldsymbol{b}(\boldsymbol{x}) + \nabla U(\boldsymbol{x})\|}. \tag{39}$$

Observing that $\boldsymbol{x}$ is the point of the path ψ at which ψ' is evaluated, we see that (39) coincides with (13). Plugging (39) into (38), we get

$$\|\boldsymbol{b}(\boldsymbol{x})\| = \|\boldsymbol{b}(\boldsymbol{x}) + \nabla U(\boldsymbol{x})\|. \tag{40}$$

Taking squares of both sides of (38), canceling $\|\boldsymbol{b}(\boldsymbol{x})\|^2$, and dividing by 2, we obtain the desired Hamilton–Jacobi equation (11):

$$\tfrac{1}{2}\|\nabla U(\boldsymbol{x})\|^2 + \boldsymbol{b}(\boldsymbol{x}) \cdot \nabla U(\boldsymbol{x}) = 0.$$

Appendix B: The dynamics of the Lorenz system (16)

Let us fix the parameters $\sigma = 10$ and $\beta = \frac{8}{3}$. As ρ grows from zero to infinity, the dynamics of (16) go through a number of bifurcations [21; 30; 31; 32].

- For all $0 < \rho < \infty$, the origin is a fixed point of (16). It is the only equilibrium for $0 < \rho < 1$, and it is globally attracting. At $\rho = 1$, a supercritical pitchfork bifurcation occurs transforming the origin into a Morse index-one saddle and giving birth to two equilibria

$$C_\pm = \bigl(\pm\sqrt{\beta(\rho-1)}, \pm\sqrt{\beta(\rho-1)}, \rho-1\bigr). \tag{41}$$

They remain asymptotically stable for $1 < \rho < \rho_2 \approx 24.74$. The unstable manifold of (16) linearized near the saddle at the origin for $1 < \rho < \infty$ is the span of the vector

$$\boldsymbol{\xi} = \begin{bmatrix} \sigma \\ (\sigma-1)/2 + \sqrt{((\sigma+1)/2)^2 + \sigma(\rho-1)} \\ 0 \end{bmatrix}. \tag{42}$$

To delineate the evolution of the dynamics of (16) as ρ grows from 1 to infinity, we have plotted the bifurcation diagram displayed in Figure 15. For each ρ from 1.05 to 349.95 with step 0.1, we traced the trajectory starting at $10^{-2}\xi$ for time $0 \le t \le 200$ and recorded its points of intersection with the plane

$$\alpha = \{\boldsymbol{x} \mid x_3 = \rho - 1\}$$

passing through the equilibria $C_\pm$. The x_1-components of these intersects are shown with pink dots in the (ρ, x_1)-plane. The time interval $0 \le t \le 200$ is large enough for

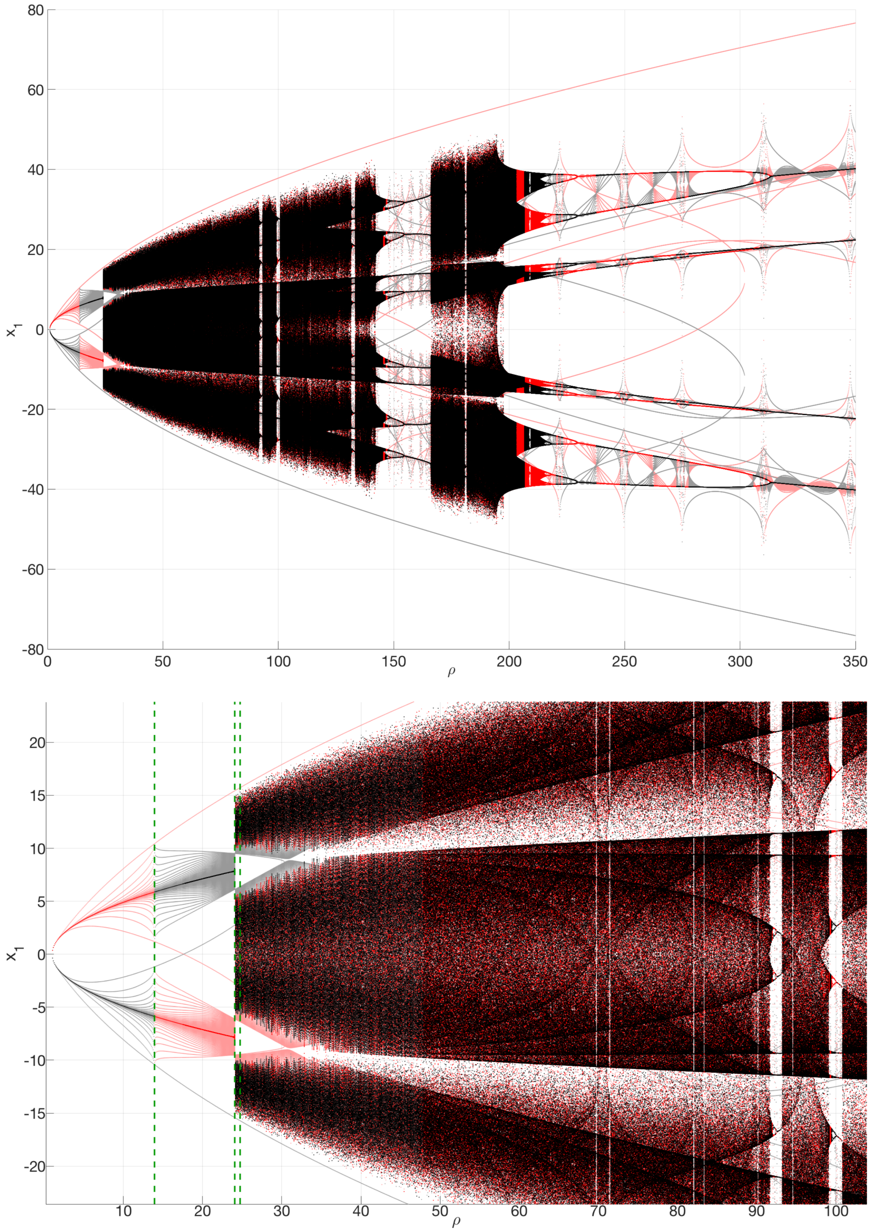


Figure 15. Top: consider the characteristics of (16) emanating from the origin along the directions $\boldsymbol{\xi}$ and $-\boldsymbol{\xi}$ and traced for the time interval $0 \le t \le 200$. The x_1-components of their intersections with the horizontal plane passing through the equilibria $C_{\pm}$ are plotted for $1 \le \rho \le 350$ with pink and gray dots, respectively. Then each characteristic continues to be traced for $200 \le t \le 400$. The resulted x_1-components of their intersections with the same plane are marked with red and black, respectively. Bottom: a zoom-in of the top. The dashed green vertical lines correspond to the critical values of ρ: $\rho_0 \approx 13.926$, $\rho_1 \approx 24.06$, and $\rho_2 \approx 24.74$.

this trajectory to approach an attractor. Then, in order to depict x_1-components of the intersection of the attractor with the plane α, we continued tracing the trajectory for $200 \le t \le 400$ and plotted the x_1-components of its intersects with α with red dots. The corresponding sets of points for the trajectory starting at $-10^{-2}\xi$ are obtained using the aforementioned symmetry of (16). They are plotted with gray and black dots, respectively. This procedure is implemented in the Matlab code `lorenz_diagram.m`.

- For $1 < \rho < \rho_0 \approx 13.926$, the characteristics emanating from the saddle at the origin along the directions $\boldsymbol{\xi}$ and $-\boldsymbol{\xi}$ approach, respectively, C_+ and C_- without crossing the plane $x_1 = 0$ (see Figure 15).

- The interval $13.926 \approx \rho_0 < \rho < \rho_2 \approx 24.74$ is marked by the existence of the saddle limit cycles γ_+ and γ_- surrounding C_+ and C_-, respectively. The equilibria $C_\pm$ remain the only attractors for $\rho_0 < \rho < \rho_1 \approx 24.06$. At $\rho = \rho_0$, there exist homoclinic orbits emanating from the origin and approaching it as $t \to \infty$. For all $\rho_0 < \rho < \rho_1$, the characteristics emanating from the origin along the directions $\boldsymbol{\xi}$ and $-\boldsymbol{\xi}$ go approximately half-way around the limit cycles, cross the plane $x_1 = 0$, and approach C_- and C_+, respectively (see Figure 15). As ρ grows within this interval, there develops a phenomenon called *preturbulence* [21], characterized by chaotic behavior and divergence of close characteristics in a region surrounding $\gamma_\pm$. Let Υ_+ be a cone consisting of all rays starting at C_+ and crossing γ_+, i.e.,

$$\Upsilon_+ := \{C_+ + t(\boldsymbol{x} - C_+) \mid t \ge 0,\ \boldsymbol{x} \in \gamma_+\}. \tag{43}$$

Characteristics starting on Υ_+ near and outside γ_+ perform more and more revolutions around C_+ and C_- prior to settling to spiraling near one of the stable equilibria. Moreover, as ρ tends to ρ_1, it is getting progressively harder and finally impossible to predict using double-precision arithmetic which equilibrium such a characteristic will eventually approach. An example of two characteristics for $\rho = 20$ starting at two close points near γ_+ on the cone Υ_+ and eventually approaching different equilibria is shown in Figure 16. At $\rho = \rho_1$, the characteristics emanating from the origin along the directions $\boldsymbol{\xi}$ and $-\boldsymbol{\xi}$ approach γ_- and γ_+, respectively. This gives birth to a strange attractor also known as the Lorenz attractor. We will denote it by A_L.

- For $24.06 \approx \rho_1 < \rho < \rho_2 \approx 24.74$, there are three attractors: the strange attractor A_L, and the asymptotically stable equilibria $C_\pm$. The characteristics emanating from the origin along $\pm\boldsymbol{\xi}$ miss the saddle cycles $\gamma_\mp$, respectively, and start spiraling away from them. The $\gamma_\pm$ lie on the boundaries of the basins of $C_\pm$, respectively, and as we show in Section 5D play roles of the escape states. At $\rho = \rho_2$, the saddle cycles $\gamma_\pm$ shrink to the corresponding equilibria $C_\pm$, rendering them unstable; i.e., a subcritical Hopf bifurcation takes place.

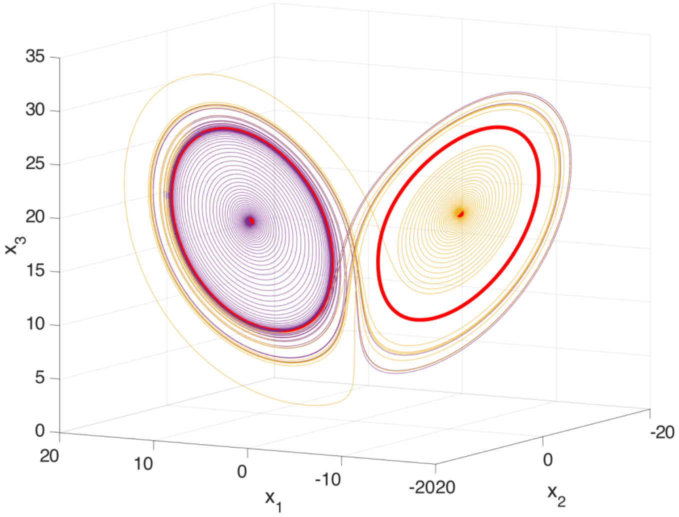


Figure 16. An example of two characteristics at $\rho = 20$ starting at two close points lying near γ_+ on the cone with vertex at C_+ and consisting of all rays passing through γ_+ and eventually diverging and approaching different equilibria.

- For $24.74 \approx \rho_2 < \rho < \infty$, the dynamics are complicated as can be inferred from Figure 15, top. A_L is the only attractor for some open interval of ρ starting at ρ_2 (Figure 15, bottom). It exists for a union of intervals of ρ stretching up to approximately $\rho = 215.364$ [30]. The interval $\rho_2 < \rho \lesssim 215.364$ is cut through by a number of windows of periodicity where there exist attracting limit cycles. The largest of them is $145 \lesssim \rho \lesssim 166$. Other windows are seen around $\rho = 93$, $\rho = 100$, $\rho = 133$, and $\rho = 181.5$. Zooming in, we can spot more windows of periodicity (see Figure 15, bottom) and reveal cascades of period doublings marking the Feigenbaum scenarios of transition to chaos. The final doubling period interval $215.364 \lesssim \rho \lesssim 313$ [30] is clearly visible in Figure 15, top. Near $\rho = 313$, two symmetric attracting limit cycles merge into one resulting in the final limit cycle that remains the only attractor for all larger values of ρ.

Appendix C: The KKT conditions for the simplex update

The Lagrange function for the constrained minimization problem (19)–(20) is

$$\mathscr{L}(\lambda, \mu) = U_\lambda + \mathcal{Q}_M(\boldsymbol{x}_\lambda, \boldsymbol{x}) - \mu_1\lambda_1 - \mu_2\lambda_2 - \mu_3(1 - \lambda_1 - \lambda_2), \tag{44}$$

where $\lambda = [\lambda_1, \lambda_2]$ and $\mu = [\mu_1, \mu_2, \mu_3]$. For brevity, we denote the function to be minimized by f:

$$f(\lambda) := U_\lambda + \mathcal{Q}_M(\boldsymbol{x}_\lambda, \boldsymbol{x}).$$

The KKT optimality conditions applied to (44) are

$$\nabla_\lambda \mathcal{L}(\lambda, \mu) = \nabla f(\lambda) - \mu_1 \begin{bmatrix}1\\0\end{bmatrix} - \mu_2 \begin{bmatrix}0\\1\end{bmatrix} - \mu_3 \begin{bmatrix}-1\\-1\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}, \tag{45}$$

$$\mu_1 \geq 0, \quad \mu_2 \geq 0, \quad \mu_3 \geq 0, \tag{46}$$

$$\lambda_1 \geq 0, \quad \lambda_2 \geq 0, \quad 1 - \lambda_1 - \lambda_2 \geq 0, \tag{47}$$

$$\lambda_1\mu_1 = 0, \quad \lambda_2\mu_2 = 0, \quad (1 - \lambda_1 - \lambda_2)\mu_3 = 0. \tag{48}$$

Let us check whether the initial guess $\lambda = [\lambda^*, 0]$ where λ^* is the minimizer of f on $[\lambda_1, 0]$, $0 < \lambda_1 < 1$, corresponding to the line segment $[\boldsymbol{x}_0, \boldsymbol{x}_1]$, satisfies the KKT conditions (45)–(48). Condition (48) with $\lambda_1 = \lambda^* \in (0, 1)$ and $\lambda_2 = 0$ implies that $\mu_1 = \mu_3 = 0$. Therefore, the first component in (45) is zero as

$$\frac{\partial}{\partial \lambda_1} f(\lambda^*, 0) = 0. \tag{49}$$

The second component of (45) must be also zero; hence,

$$\frac{\partial}{\partial \lambda_2} f(\lambda^*, 0) - \mu_2 = 0. \tag{50}$$

Condition (46) demands that $\mu_2 \geq 0$. Hence, $\lambda = [\lambda^*, 0]$ is a solution of the constrained minimization problem (19)–(20) if

$$\mu_2 = \frac{\partial}{\partial \lambda_2} f(\lambda^*, 0) \geq 0, \tag{51}$$

i.e., if (21) holds. In this case, we reject the simplex update. Otherwise, we proceed with solving the minimization problem (19)–(20).

Appendix D: Quasipotential decomposition for linear SDEs

In this appendix, we explain how one can find the quasipotential for linear SDEs for which the origin is an asymptotically stable equilibrium. This is useful for initializing the OLIMs near asymptotically stable equilibria and for estimating the ratio of the magnitudes of the rotational and potential components of the vector field.

Let J be a $d \times d$ matrix with all eigenvalues having negative real parts. In this work, J is the Jacobian matrix of the vector field $\boldsymbol{b}$ evaluated at an asymptotically stable equilibrium $\boldsymbol{x}^*$ of $\dot{\boldsymbol{x}} = \boldsymbol{b}(\boldsymbol{x})$. We consider the linear SDE for the variable $\boldsymbol{y} := \boldsymbol{x} - \boldsymbol{x}^*$:

$$d\boldsymbol{y} = J\boldsymbol{y}\,dt + \sqrt{\epsilon}\,d\boldsymbol{w}. \tag{52}$$

The problem of finding the quasipotential decomposition for the vector field $J\boldsymbol{y}$ reduces to the problem of finding a symmetric positive definite matrix Q such that [8; 7]

$$\boldsymbol{y}^\top Q(J+Q)\boldsymbol{y}=0 \quad \text{for all } \boldsymbol{y}\in\mathbb{R}^d. \tag{53}$$

The matrices Q and $L := J+Q$ are called the *quasipotential matrix* and the *rotational matrix*, respectively. Condition (53) is equivalent to the requirement that the matrix $Q(J+Q)$ is antisymmetric, i.e., $Q(J+Q)+(J+Q)^\top Q=0$. The last equation for Q is reducible to a Sylvester equation for Q^{-1} and has a unique positive definite solution that can be found using the Bartels–Stewart algorithm implemented in Matlab in the command `sylvester` (see [35] for details).

To make our quasipotential solver for the Lorenz system self-contained and facilitate experiments with various values of ρ, we have developed a C code `LinLorenz.c` for finding the quasipotential decomposition for the Lorenz system linearized near its asymptotically stable equilibria. The quasipotential decomposition is found by an algorithm similar to Bartels–Stewart but simplified and customized for Lorenz’63. A description of it is linked to the provided software package [5].

Once the quasipotential decomposition for a linearized system is available, one can obtain an estimate for the ratio $\Xi(\boldsymbol{x})$ of the magnitudes of the rotational and potential components near asymptotically stable equilibria:

$$\Xi \lesssim \max_{\|\boldsymbol{y}\|=1}\frac{\|L\boldsymbol{y}\|}{\|Q\boldsymbol{y}\|}. \tag{54}$$

The graph of the right-hand side of (54) with J been the Jacobian matrix evaluated at C_+ of (16) is plotted in Figure 2 for the range $1<\rho<\rho_2\approx 24.74$.

Appendix E: Proof of Lemma 4.1

Proof. First we prove that the manifold $\mathcal{M}'$ consisting of MAPs going from the attractor A to the curve γ lies in the sublevel set $\mathcal{V}_a$. Let ψ be a MAP going from A to γ. Since $\mathcal{V}_a$ completely lies in the basin of A, the quasipotential strictly increases along the MAP. Therefore, for any $\boldsymbol{y}$ lying on the path ψ, $U(\boldsymbol{y})\le a$, which means that $\psi\subset\mathcal{V}_a$. Since this is true for all such MAPs, $\mathcal{M}'\subset\mathcal{V}_a$.

Now let us prove that the manifold $\mathcal{M}$ consisting of all characteristics starting at γ and running to A lies in $\mathcal{V}_a$. We proceed from the converse. Suppose a characteristic starting at γ and going to A leaves $\mathcal{V}_a$ at a point $\boldsymbol{x}_0$ and reenters $\mathcal{V}_a$ at a point $\boldsymbol{x}_1$ after that. Let $\boldsymbol{y}$ be a point of this characteristic located between $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$. Since the motion of the characteristic contributes nothing to the Freidlin–Wentzell action (8), $U(\boldsymbol{y})=U(\boldsymbol{x}_0)=a$. This contradicts the assumption that $\boldsymbol{y}\notin\mathcal{V}_a$. Therefore, the characteristic must completely lie in $\mathcal{V}_a$. Since this argument applies to all characteristics constituting $\mathcal{M}$, we conclude that $\mathcal{M}\subset\mathcal{V}_a$. □

Appendix F: Building radial meshes

Suppose we would like to build a radial mesh on a 2D manifold formed by characteristics of $\dot{\boldsymbol{x}} = \boldsymbol{b}(\boldsymbol{x})$ going from an unstable limit cycle γ to an asymptotically stable spiral point $\boldsymbol{x}^*$. First, we pick a set of points $\boldsymbol{x}^k$, $k = 0, 1, \dots, N_a - 1$, equispaced along γ. For each point $\boldsymbol{x}^k$, we define a plane α^k passing through $\boldsymbol{x}^*$ and $\boldsymbol{x}^k$ whose normal $\boldsymbol{a}^k$ lies in the plane spanned by $\boldsymbol{b}(\boldsymbol{x}^k)$ and $\boldsymbol{x}^k - \boldsymbol{x}^*$.

Then, we trace a trajectory $\boldsymbol{y}(t)$ starting near γ and ending upon reaching a δ-ball centered at $\boldsymbol{x}^*$ where δ is a small number. Let $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n$ be the set of intersects of $\boldsymbol{y}(t)$ with the plane α^0 at which the sign of $(\boldsymbol{y}(t) - \boldsymbol{x}^0)^\top \boldsymbol{a}^0$ changes from "$-$" to "$+$". Adding $\boldsymbol{x}^0$ and $\boldsymbol{x}^*$ to this set and interpolating, we get a curve lying in α_0 and connecting γ and $\boldsymbol{x}^*$. We define a set of points $\{z_i^0\}_{i=0}^{N_r-1}$ uniformly distributed along this curve such that $z_0^0 \equiv \boldsymbol{x}^*$ and $z_{Nr-1}^0 \equiv \boldsymbol{x}^0$.

Next, for $k = 0, 1, 2, \dots, N_a - 2$, we trace the trajectories starting at z_i^k, $i = 1, \dots, N_r - 2$, and terminate them as soon as they reach the plane α_{k+1}. As above, we add $\boldsymbol{x}_{k+1}$ and $\boldsymbol{x}^*$ to these terminal points, interpolate them, and pick a set of points z_i^{k+1}, $i = 0, \dots, N_r - 1$, uniformly distributed along the interpolant and such that $z_0^{k+1} \equiv \boldsymbol{x}^*$ and $z_{N_r-1}^{k+1} \equiv \boldsymbol{x}^{k+1}$. As a result, we obtain the radial mesh

$$\{z_i^k \mid 0 \le i \le N_r - 1,\ 0 \le k \le N_a - 1\}.$$

This procedure is implemented in the Matlab code `make2Dmesh.m` in the package `Qpot4Lorenz63.zip` [5].

Similar methodologies have been used to construct radial meshes between two simple closed curves and between two given segments of two distinct characteristics.

Appendix G: Estimating the width of the Lorenz attractor

Let $\boldsymbol{x}$ be a point lying on the Lorenz attractor A_L, and let α be the plane passing through $\boldsymbol{x}$ and normal to $\boldsymbol{b}(\boldsymbol{x})$ where $\boldsymbol{b}$ is the Lorenz vector field; i.e.,

$$\alpha := \{z \in \mathbb{R}^3 \mid (z - \boldsymbol{x})^\top \boldsymbol{b}(\boldsymbol{x}) = 0\}.$$

We trace a trajectory $\boldsymbol{y}(t)$ starting at $\boldsymbol{x}$ for time 10^4 and record the points $\boldsymbol{y}_i$, $1 \le i \le N$, at which the sign of $(\boldsymbol{y}(t) - \boldsymbol{x})^\top \boldsymbol{b}(\boldsymbol{x})$ switches from "$-$" to "$+$". We set up a Cartesian coordinate system (η_1, η_2) in the plane α with the origin at $\boldsymbol{y}_1 \equiv \boldsymbol{x}$ and find the coordinates of the recorded points $\boldsymbol{y}_i$: $\boldsymbol{y}_i \equiv (\eta_1^i, \eta_2^i)$. We pick a square $S := [-0.25 \le \eta_1 \le 0.25] \times [-0.25 \le \eta_2 \le 0.25]$ in this plane and select the subset $I \subset \{1, \dots, N\}$ such that the points $\boldsymbol{y}_i$, $i \in I$, lie in S. Visualizing the set $\boldsymbol{y}_i$, $i \in I$, and zooming in if necessary, we see that they are arranged near two almost parallel lines (see Figure 17). The least squares fit to this set of points with a linear function

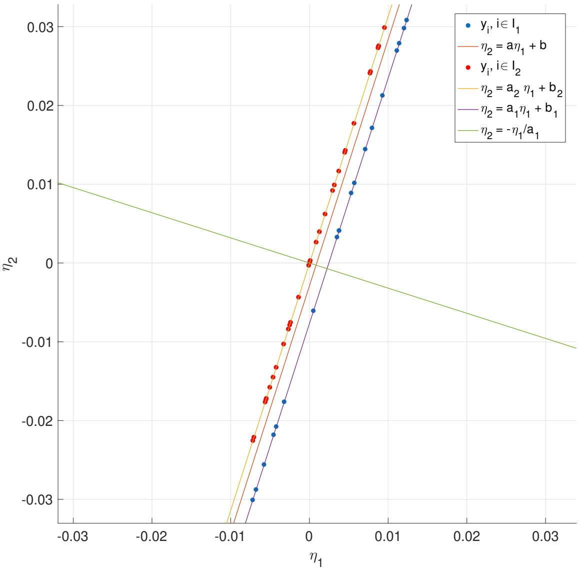


Figure 17. Estimating the thickness of the Lorenz attractor using linear least squares fits in a Poincaré section.

$\eta_2 = a\eta_1 + b$ gives a line dividing it into two subsets:

$$I_1 = \{i \in I \mid \eta_2^i < a\eta_1^i + b\},$$
$$I_2 = \{i \in I \mid \eta_2^i > a\eta_1^i + b\}.$$

Next, we find linear least squares fits $\eta_2 = a_1\eta_1 + b_1$ and $\eta_2 = a_2\eta_1 + b_2$ for the subsets of $\boldsymbol{y}_i$ corresponding to I_1 and I_2, respectively. One of these linear functions must pass very close to the origin because $\boldsymbol{x}$ lies near one of these lines; hence, either b_1 or b_2 is very close to zero in comparison with the other one. Assume that $|b_2| \ll |b_1|$. If this is the other way around, we swap the notations. Also, these lines are almost parallel; hence, a_1 and a_2 are very close. Finally, we find a line orthogonal to $\eta_2 = a_1\eta_1 + b_1$ and passing through the origin: $\eta_2 = -a_1^{-1}\eta_1$. Then the thickness of A_L near $\boldsymbol{x}$ is approximately equal to the distance between the origin and the intersect of $\eta_2 = -a_1^{-1}\eta_1$ and $\eta_2 = a_1\eta_1 + b_1$. This technique is implemented in the Matlab program `thickness.m` [5; 4].

Acknowledgements

We thank Doctor E. S. Kurkina for inspiring discussions on the Lorenz’63 system and sharing numerical techniques for plotting bifurcational diagrams and finding unstable limit cycles. We are also grateful to Professor James Yorke and Professor Kevin Lin for valuable advice regarding preparation of this manuscript. This work is partially supported by NSF grant DMS1554907.

References

[1] R. Bellman, *Dynamic programming*, Princeton University, 1957. MR Zbl

[2] F. Bouchet and J. Reygner, *Generalisation of the Eyring–Kramers transition rate formula to irreversible diffusion processes*, Ann. Henri Poincaré **17** (2016), no. 12, 3499–3532. MR Zbl

[3] M. K. Cameron, *Finding the quasipotential for nongradient SDEs*, Phys. D **241** (2012), no. 18, 1532–1550. MR Zbl

[4] ———, *OLIM-for-Lorenz63*, 2019, C and Matlab code, version 1.1, also available on GitHub.

[5] ———, *OLIM: ordered line integral methods for computing the quasi-potential*, 2019, C and Matlab code.

[6] A. Chacon and A. Vladimirsky, *Fast two-scale methods for eikonal equations*, SIAM J. Sci. Comput. **34** (2012), no. 2, A547–A578. MR Zbl

[7] Z. Chen, *Asymptotic problems related to Smoluchowski–Kramers approximation*, Ph.D. thesis, University of Maryland, 2006.

[8] Z. Chen and M. Freidlin, *Smoluchowski–Kramers approximation and exit problems*, Stoch. Dyn. **5** (2005), no. 4, 569–585. MR Zbl

[9] M. G. Crandall and P.-L. Lions, *Viscosity solutions of Hamilton–Jacobi equations*, Trans. Amer. Math. Soc. **277** (1983), no. 1, 1–42. MR Zbl

[10] D. Dahiya and M. Cameron, *An ordered line integral method for computing the quasi-potential in the case of variable anisotropic diffusion*, Phys. D **382/383** (2018), 33–45. MR Zbl

[11] ———, *Ordered line integral methods for computing the quasi-potential*, J. Sci. Comput. **75** (2018), no. 3, 1351–1384. Zbl

[12] S. V. Dudul, *Prediction of a Lorenz chaotic attractor using two-layer perceptron neural network*, Appl. Soft Comput. **5** (2005), no. 4, 333–355.

[13] W. E, W. Ren, and E. Vanden-Eijnden, *Minimum action method for the study of rare events*, Comm. Pure Appl. Math. **57** (2004), no. 5, 637–656. MR Zbl

[14] M. I. Freidlin and A. D. Wentzell, *Random perturbations of dynamical systems*, 3rd ed., Grundlehren der Mathematischen Wissenschaften, no. 260, Springer, 2012. MR Zbl

[15] C. Gissinger, *A new deterministic model for chaotic reversals*, Eur. Phys. J. B **85** (2012), no. 4, 137–148.

[16] J. Guckenheimer and R. F. Williams, *Structural stability of Lorenz attractors*, Inst. Hautes Études Sci. Publ. Math. (1979), no. 50, 59–72. MR Zbl

[17] F. Hamilton, T. Berry, and T. Sauer, *Predicting chaotic time series with a partial model*, Phys. Rev. E **92** (2015), no. 1, art. id. 010902(R).

[18] M. Heymann and E. Vanden-Eijnden, *The geometric minimum action method: a least action principle on the space of curves*, Comm. Pure Appl. Math. **61** (2008), no. 8, 1052–1117. MR Zbl

[19] ———, *Pathways of maximum likelihood for rare events in nonequilibrium systems: application to nucleation in the presence of shear*, Phys. Rev. Lett. **100** (2008), no. 14, art. id. 140601.

[20] H. Ishii, *A simple, direct proof of uniqueness for solutions of the Hamilton–Jacobi equations of eikonal type*, Proc. Amer. Math. Soc. **100** (1987), no. 2, 247–251. MR Zbl

[21] J. L. Kaplan and J. A. Yorke, *Preturbulence: a regime observed in a fluid flow model of Lorenz*, Comm. Math. Phys. **67** (1979), no. 2, 93–108. MR Zbl

[22] E. N. Lorenz, *Deterministic nonperiodic flow*, J. Atmos. Sci. **20** (1963), no. 2, 130–141. Zbl

[23] C. Lv, X. Li, F. Li, and T. Li, *Constructing the energy landscape for genetic switching system driven by intrinsic noise*, PLOS One **9** (2014), no. 2, art. id. e88167.

[24] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed., Springer, 2006. MR Zbl

[25] D. Rand, *The topological classification of Lorenz attractors*, Math. Proc. Cambridge Philos. Soc. **83** (1978), no. 3, 451–460. MR Zbl

[26] B. Saltzman, *Finite amplitude free convection as an initial value problem, I*, J. Atmos. Sci. **19** (1962), no. 4, 329–341.

[27] J. A. Sethian and A. Vladimirsky, *Ordered upwind methods for static Hamilton–Jacobi equations*, Proc. Natl. Acad. Sci. USA **98** (2001), no. 20, 11069–11074. MR Zbl

[28] ———, *Ordered upwind methods for static Hamilton–Jacobi equations: theory and algorithms*, SIAM J. Numer. Anal. **41** (2003), no. 1, 325–363. MR Zbl

[29] F. Sorrentino and E. Ott, *Using synchronization of chaos to identify the dynamics of unknown systems*, Chaos **19** (2009), no. 3, art. id. 033108. Zbl

[30] C. Sparrow, *The Lorenz equations: bifurcations, chaos, and strange attractors*, Applied Mathematical Sciences, no. 41, Springer, 1982. MR Zbl

[31] ———, *An introduction to the Lorenz equations*, IEEE Trans. Circuits and Systems **30** (1983), no. 8, 533–542. MR

[32] S. H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*, 2nd ed., Westview, Boulder, CO, 2015. MR Zbl

[33] D. Viswanath, *The fractal property of the Lorenz attractor*, Phys. D **190** (2004), no. 1-2, 115–128. MR Zbl

[34] R. F. Williams, *The structure of Lorenz attractors*, Inst. Hautes Études Sci. Publ. Math. (1979), no. 50, 73–99. MR Zbl

[35] S. Yang, S. F. Potter, and M. K. Cameron, *Computing the quasipotential for nongradient SDEs in 3D*, J. Comput. Phys. **379** (2019), 325–350. MR

[36] J. A. Yorke and E. D. Yorke, *Metastable chaos: the transition to sustained chaotic behavior in the Lorenz model*, J. Statist. Phys. **21** (1979), no. 3, 263–277. MR

[37] X. Zhou and W. E, *Study of noise-induced transitions in the Lorenz system using the minimum action method*, Commun. Math. Sci. **8** (2010), no. 2, 341–355. MR Zbl

[38] X. Zhou, W. Ren, and W. E, *Adaptive minimum action method for the study of rare events*, J. Chem. Phys. **128** (2008), no. 10, art. id. 104111.

Received November 22, 2018. Revised June 15, 2019.

MARIA CAMERON: cameron@math.umd.edu
Department of Mathematics, University of Maryland, College Park, College Park, MD, United States

SHUO YANG: shuoyang@math.umd.edu
Department of Mathematics, University of Maryland, College Park, College Park, MD, United States

COMM. APP. MATH. AND COMP. SCI.
Vol. 14, No. 2, 2019
dx.doi.org/10.2140/camcos.2019.14.247

EFFICIENT MULTIGRID SOLUTION OF ELLIPTIC INTERFACE PROBLEMS USING VISCOSITY-UPWINDED LOCAL DISCONTINUOUS GALERKIN METHODS

ROBERT I. SAYE

With an emphasis on achieving ideal multigrid solver performance, this paper explores the design of local discontinuous Galerkin schemes for multiphase elliptic interface problems. In particular, for cases exhibiting coefficient discontinuities several orders in magnitude, the role of viscosity-weighted numerical fluxes on interfacial mesh faces is examined: findings support a known strategy of harmonic weighting, but also show that further improvements can be made via a stronger kind of biasing, denoted herein as viscosity-upwinded weighting. Applying this strategy, multigrid performance is assessed for a variety of elliptic interface problems in 1D, 2D, and 3D, across 16 orders of viscosity ratio. These include constant- and variable-coefficient problems, multiphase checkerboard patterns, implicitly defined interfaces, and 3D problems with intricate geometry. With the exception of a challenging case involving a lattice of vanishingly small droplets, in all demonstrated examples the condition number of the multigrid V-cycle preconditioned system has unit order magnitude, independent of the mesh size h.

1. Introduction

In this work, we consider the design of local discontinuous Galerkin schemes for multiphase elliptic interface problems containing large discontinuities in the ellipticity coefficient. In particular, we explore the possibility of altering certain aspects of the discretization to benefit both multigrid solver performance as well as solution accuracy. The prototype problem considered here consists of solving for a function $u : \Omega \to \mathbb{R}$ such that

$$\begin{cases} -\nabla \cdot (\mu_i \nabla u) = f_i & \text{in } \Omega_i, \\ [\![u]\!] = g_{ij} & \text{on } \Gamma_{ij}, \\ [\![(\mu \nabla u) \cdot \boldsymbol{n}]\!] = h_{ij} & \text{on } \Gamma_{ij}, \\ u = g_\partial & \text{on } \Gamma_D, \\ (\mu \nabla u) \cdot \boldsymbol{n} = h_\partial & \text{on } \Gamma_N, \end{cases} \tag{1}$$

MSC2010: 65F08, 65N30, 65N55.

Keywords: elliptic interface problems, multigrid methods, local discontinuous Galerkin methods, implicitly defined meshes, harmonic weights, viscosity-upwinded weighting, operator coarsening.

where Ω is a domain in $\mathbb{R}^d$ divided into two or more subdomains Ω_i (denoted "phases"), $\Gamma_{ij} := \partial\Omega_i \cap \partial\Omega_j$ is the interface between phases i and j, and Γ_D and Γ_N denote the components of $\partial\Omega$ on which Dirichlet and Neumann boundary conditions are imposed. Here, $[\![\, \cdot \,]\!]$ denotes the jump in a quantity across an interface and $\boldsymbol{n}$ is to be understood from context — on $\partial\Omega$, $\boldsymbol{n}$ denotes the outward unit normal to the domain boundary, whereas for an interface Γ_{ij}, $\boldsymbol{n}$ denotes the unit normal to Γ_{ij}, oriented consistently with the definition of the jump operator $[\![\, \cdot \,]\!]$. In the general elliptic interface problem (1), μ_i is a phase-dependent ellipticity/viscosity coefficient; throughout this work, μ_i is taken to be a (continuous) positive-valued scalar function[1] $\mu_i : \Omega_i \to \mathbb{R}^+$. Finally, f, g, and h provide the data to the elliptic interface problem, and are given functions defined on Ω, its boundary, and internal interfaces.

Our motivation in this work is to develop local discontinuous Galerkin (LDG) [22] methods capable of handling interfacial jumps in viscosity of several orders in magnitude. To design an LDG scheme for (1), one must choose appropriate numerical fluxes for the primary unknown u and its associated auxiliary flux variable $\boldsymbol{q} = \mu\nabla u$. On a typical mesh face, the numerical fluxes $u^\star$ and $\boldsymbol{q}^\star$ are chosen as some convex combination of the trace values of their associated polynomials on either side of the face. The focus of this study is to develop a suitable weighting strategy on interfacial faces. On these faces it is often beneficial to bias the numerical flux towards one phase or the other, depending on the local values of the viscosity coefficient μ_i or μ_j, which could differ by several orders of magnitude. Doing so may not only improve solution accuracy, but can also markedly improve conditioning and multigrid performance — in the next section we provide a physical motivation for why this may be. Following the motivational example, previous work in this area is reviewed.

1.1. *Weighted numerical fluxes.* To physically motivate the possible merits of viscosity-weighted fluxes, we consider here a simple two-phase elliptic interface problem and examine the case of a vanishingly small viscosity ratio. In particular, suppose the domain is divided into two phases, Ω_1 and Ω_ϵ, with viscosity coefficients 1 and $0 < \epsilon \ll 1$, respectively. Rewriting (1) for this case, we have

$$\begin{cases} -\nabla^2 u_1 = f_1 & \text{in } \Omega_1, \\ -\epsilon\nabla^2 u_\epsilon = f_\epsilon & \text{in } \Omega_\epsilon, \\ u_1 - u_\epsilon = g & \text{on } \Gamma, \\ \boldsymbol{n}\cdot\nabla u_1 - \epsilon\boldsymbol{n}\cdot\nabla u_\epsilon = h & \text{on } \Gamma, \end{cases}$$

where $\Gamma = \partial\Omega_1 \cap \partial\Omega_\epsilon$, subject to boundary conditions on $\partial\Omega$ (which are unimportant in this motivational setting). We assume the data f, g, and h are such that the

[1] Comments concerning the more general case that μ_i may be matrix-valued are provided in the concluding remarks.

solution u and its gradient near the interface is $\mathbb{O}(1)$ as ϵ is made vanishingly small. In this limit, the second term in the flux jump condition vanishes, resulting in phase Ω_1 (approximately) having the Neumann boundary condition $\boldsymbol{n} \cdot \nabla u_1 \approx h$ on Γ. Thus, the solution u_1 can (almost) be determined in isolation and essentially decouples from the other phase. Once u_1 is found, the elliptic problem in phase Ω_ϵ essentially reduces to a Dirichlet boundary condition on Γ, i.e., $u_\epsilon|_\Gamma = u_1|_\Gamma - g$. Therefore, for $\epsilon \ll 1$, the two-phase elliptic interface problem (nearly) decouples into two separate single-phase elliptic problems; the phase with unit viscosity coefficient "sees" a Neumann boundary condition on Γ whose data is (nearly) independent of the solution in the other phase, and the phase with vanishingly small viscosity coefficient "sees" a Dirichlet boundary condition on Γ whose data depends on the solution on the other side of the interface.

This simple example is predicated on the assumption that, near the interface, the solution u and its gradient have magnitude independent of $\epsilon \ll 1$. Naturally, this may not hold in practice owing to potential boundary layers in the exact solution; however, the above observation, that the two phases might nearly decouple and see different types of interfacial boundary conditions, illustrates that an apt choice of numerical flux could improve accuracy and conditioning of a numerical discretization. In particular, for an LDG scheme, the numerical flux for $u^\star$ on an interfacial face should bias towards the phase Ω_1 — doing so effectively recasts the numerical flux for phase Ω_1 as it would appear for a Neumann boundary, and for Ω_ϵ as it would appear for a Dirichlet boundary (wherein the interfacial jump data g is also incorporated). In addition, the numerical flux for $\boldsymbol{q}^\star$ should bias towards phase Ω_ϵ — doing so is consistent with specifying Dirichlet boundary conditions for the problem in Ω_ϵ, and also effectively sets boundary conditions $\boldsymbol{q} \cdot \boldsymbol{n} \approx h$ for phase Ω_1.

The same example can be used to provide an indication of an appropriate penalty parameter choice for interfacial faces. Penalty stabilization is often used in DG methods to weakly enforce solution continuity, to weakly impose Dirichlet boundary conditions, and to ensure overall well-posedness of the discrete problem. Generally speaking, penalty parameters should scale with the local ellipticity coefficient — a simple argument for this is that a (single-phase) Poisson problem $-\mu \nabla^2 u = f$ results in a linear system $-\mu \Delta_h u + \tau E = f$, where Δ_h is the discrete Laplacian and E is a penalty operator with its dependence on the penalty parameter τ made explicit; since scaling both sides by μ^{-1} should result in exactly the same discrete solution, τ should therefore scale proportionally with μ. Returning to the above two-phase elliptic interface problem, from the perspective of the Dirichlet problem in phase Ω_ϵ, we observe that the difference between u_ϵ and its effective Dirichlet data of $u_1 - g$ should be penalized with a parameter that is proportional to ϵ, its effective local viscosity.

In summary, and to generalize this intuition to the case of an interface Γ_{ij} between two phases of arbitrary (positive) viscosity, (i) the numerical flux for $u^\star$ should bias towards the phase with (locally) largest viscosity, (ii) $\boldsymbol{q}^\star$ should bias towards the phase with (locally) smallest viscosity, and (iii) DG penalty stabilization parameters should scale proportionally with the smaller of the two viscosity values. Note that, along the extent of an interface, the biasing direction could switch between phases whenever the viscosity ratio changes from less than unity to greater than unity. In the context of LDG methods, the goal of this paper is to determine an ideal strategy for the specific amount of biasing/weighting, as a function of the viscosity ratio.

1.2. ***Previous work.*** The purpose of the above motivation was to make plausible the possible merits of viscosity-weighted fluxes — this idea is not new and viscosity-weighted discretization schemes have been used in a variety of different settings. The most common technique also refers to the particular strategy used to choose the weights, i.e., *harmonic weighting*.[2] Among the first to apply this technique, Dryja [24] used harmonic averaging in a DG-based multilevel additive Schwarz method to derive optimal error bounds for an elliptic interface problem, while Burman and Zunino [17] considered domain decomposition methods for advection-diffusion-reaction problems in a Nitsche finite element setting. Later, Zunino [56] derived a weighted interior penalty DG scheme using harmonic weights; this work was then extended in [28; 18] to general viscosity tensors. The particular choice of harmonic weighting, as well as biasing of penalty parameters, has often been suggested by theoretical error analyses, e.g., for discontinuous Galerkin methods [19; 15], nonconforming finite element methods [27], and unfitted Nitsche methods [35; 16]. Application areas of harmonic weighting include multimaterial Stokes problems [52], Helmholtz problems in which the weighting depends on sound speed [55], as well as incompressible two-phase flow and fluid structure interaction [47]. In cut cell finite element methods, the weighting strategy is sometimes adapted to account not only for differing viscosity coefficients, but also for the measure of the cut element (and in the case of penalty parameters, also the measure of the cut face), as carefully analyzed by Annavarapu et al. [6] (see also [10; 50]); applications of this idea include Stokes problems [33] and two-phase incompressible flow [30], the latter work also suggesting that the weights could take into account the viscosity-to-density ratio of the two fluids. Methods which weight based on viscosity as well as cut element size have recently been adapted to handle extreme cases of these combinations by Gürkan and Massing [32]. Besides the aforementioned works, which mainly consider finite element methods, harmonic weighting has also found applications in finite difference and finite volume methods to treat discontinuous or nonsmooth diffusion coefficients; see, e.g., [13; 37; 3; 23].

In addition, considerable work on high-contrast/large-jump elliptic interface problems has focused on designing efficient solvers, including domain decomposition,

[2] The precise definition of harmonic-weighted numerical fluxes is given later in Section 4.2.

multilevel, and multigrid methods. Generally, the numerical discretization method is fixed ahead of time and the task concerns the design of a solver or preconditioner with the best possible performance. One possibility is to take advantage of the weak decoupling suggested in the above motivational example to solve the elliptic problem in each subdomain, and then assemble into a global solution; see, e.g., [41; 36]. Generally, better performance can be obtained with multilevel or multigrid methods. For example, Dryja et al. [26] considered multilevel Schwarz preconditioners for conforming finite element methods having interpolation operators that bias towards more viscous subdomains. Two- or multilevel domain decomposition and additive Schwarz methods have been developed with convergence rates independent or nearly independent of the viscosity ratio; see, e.g., [24; 54; 53; 31; 25; 7]. A wide array of geometric multigrid methods have also been devised for elliptic interface problems, some of which take into account interface geometry when building the hierarchy [20; 23; 44; 29; 51], including those operating on DG and cut finite element methods schemes derived with harmonic weighting [39; 12], and methods which apply direction-dependent coarsening of the diffusion coefficient using a combination of arithmetic and harmonic averaging; see, e.g., [4; 5; 48]. As an alternative to geometric multigrid methods, algebraic multigrid methods can automate some of the process; these operate through identification of ellipticity-dependent connections in the matrix so as to inform the choice of aggregation procedure; see, e.g., [2; 3; 14; 11]. Other kinds of solvers have been devised according to the particular physics application at hand. For example, "bubbly" geometry problems involve a domain with many small, dispersed subdomains of markedly different ellipticity coefficient (one may think of tiny gas bubbles rising in a liquid); for these problems, it can be beneficial to isolate problematic subdomains and remove them from a Krylov-based solver, e.g., by using deflated conjugate gradient methods [40; 49].

In comparison, this work considers viscosity-weighted fluxes in a LDG framework, with a particular focus on altering the discretization to obtain ideal multigrid performance. Prior work on weighting in DG methods has suggested connections to LDG specifically, e.g., the weighted symmetric interior penalty method [56; 28] and Nitsche methods [35]; however, these works did not explore weighted fluxes in a purely LDG framework. As far as the author is aware, no prior work has considered weighted fluxes in the context of tuning associated geometric multigrid solvers. In particular, the presented results suggests that the best accuracy and conditioning can be obtained by using weighted fluxes that bias even more strongly than harmonic weighting.

1.3. ***Outline.*** The remainder of the paper is organized as follows. In Section 2, a local discontinuous Galerkin framework is outlined for the multiphase elliptic interface problems under consideration. Section 3 describes the construction of the associated multigrid methods and the specific choice of V-cycle preconditioned conjugate gradient algorithms. A one-dimensional investigation is then presented in Section 4 showing the effect of weighted fluxes on solution accuracy, multigrid behavior, and condition numbers of the preconditioned systems. Section 5 follows with a variety of test problems in two and three dimensions, ranging from simple two-phase problems to multiphase variable-coefficient problems, and challenging cases with bubbly geometry. In particular, the presented tests examine ellipticity coefficients ranging across 16 orders of magnitude. Concluding remarks are then given in Section 6.

2. Local discontinuous Galerkin methods

To derive a discontinuous Galerkin method for (1), a standard approach is to introduce an auxiliary variable $\boldsymbol{q} = \mu \nabla u$ and rewrite the system as

$$\begin{cases} \boldsymbol{q} = \mu_i \nabla u & \text{in } \Omega_i, \quad [\![u]\!] = g_{ij} \quad \text{on } \Gamma_{ij}, \quad u = g_\partial \quad \text{on } \Gamma_D, \\ -\nabla \cdot \boldsymbol{q} = f_i & \text{in } \Omega_i, \quad [\![\boldsymbol{q}\cdot\boldsymbol{n}]\!] = h_{ij} \quad \text{on } \Gamma_{ij}, \quad \boldsymbol{q}\cdot\boldsymbol{n} = h_\partial \quad \text{on } \Gamma_N. \end{cases} \tag{2}$$

In this work, we consider discretizations wherein the corresponding meshes arise from Cartesian grids as well as quadtree/octree-based implicitly defined meshes of more complex curved domains. In this setting, it is natural to adopt a tensor-product piecewise polynomial space. Let $\mathscr{E} = \bigcup_i E_i$ denote the set of elements of the mesh; we assume in particular the mesh is interface-conforming, i.e., the multiphase interface does not cut through any element. Let $p \geq 1$ be an integer and define $\mathscr{Q}_p(E)$ to be the space of tensor-product polynomials of degree p on the element E. For example, $\mathscr{Q}_2$ is the space of biquadratic (in 2D) or triquadratic (in 3D) polynomials having dimension 9 or 27, respectively. Define the corresponding spaces of discontinuous piecewise polynomials and vector fields on the mesh as

$$\begin{aligned} V_h(\mathscr{E}) &= \{v : \Omega \to \mathbb{R} \mid v|_E \in \mathscr{Q}_p(E) \text{ for every } E \in \mathscr{E}\}, \\ V_h^d(\mathscr{E}) &= \{\boldsymbol{\omega} : \Omega \to \mathbb{R}^d \mid \boldsymbol{\omega}|_E \in [\mathscr{Q}_p(E)]^d \text{ for every } E \in \mathscr{E}\}. \end{aligned}$$

Our focus in this work is on a local discontinuous Galerkin (LDG) [22] discretization of (2). The particulars of the discretization are relatively standard except for two aspects: (a) interfacial faces have a multivalued numerical flux, and (b) the weak form for $\boldsymbol{q}_h$ and u_h is defined carefully to account for the possibility of quadrature schemes which may not exactly preserve the identity of integration-by-parts for polynomial integrands. This consideration is important in the case of implicitly defined meshes which have curved element geometry specified by one or more level set functions — in this setting, high-order accurate quadrature schemes are used to implement the weak form, but integration-by-parts may only hold up to a high-order truncation error. For extended details, the reader is referred to [44; 45];

these references, however, only consider constant-coefficient elliptic problems, whereas in the present work the possibility of variable μ is considered. A brief description of the extension of these LDG methods to variable μ is provided here.

To establish some notation, regarding the faces of the mesh, we denote *intraphase faces* as those shared by two elements of the same phase, *interphase faces* as those shared by two elements of differing phases (and thus are situated on Γ_{ij} for some i, j), and *boundary faces* as those situated on $\partial\Omega$. Each face has a corresponding unit normal vector $\boldsymbol{n}$; in this work, intraphase faces are always flat and lie in a particular coordinate plane so that $\boldsymbol{n}$ is defined to point from "left-to-right", e.g., for vertical faces in 2D, $\boldsymbol{n} = \hat{\boldsymbol{x}}$ and, for horizontal faces, $\boldsymbol{n} = \hat{\boldsymbol{y}}$. Interphase faces adopt the same normal vector as the interface Γ_{ij} on which they coincide, defined to point from the phase i with smallest phase index into the phase with largest index $j > i$. Boundary faces adopt the natural outwards-pointing normal to the domain boundary. The notation $[\![\cdot]\!]$ denotes the jump of a quantity across an interface or face and is defined consistent with its orientation; in particular, $[\![u]\!] := u^- - u^+$ where $u^\pm(x) = \lim_{\epsilon\to 0^+} u(x \pm \epsilon\boldsymbol{n})$ denotes the left u^- and right u^+ trace values. Last, for an element $E \in \mathscr{E}$, define $\chi(E)$ to be the phase of that element, such that $E \subseteq \Omega_{\chi(E)}$.

In the first of four steps in formulating the LDG method, we define a discrete approximation of ∇u via a "strong-weak form"; given $u \in V_h$, $\boldsymbol{\eta} \in V_h^d$ is defined such that

$$\int_E \boldsymbol{\eta}\cdot\boldsymbol{\omega} = \int_E \nabla u\cdot\boldsymbol{\omega} + \int_{\partial E} (u^\star_{\chi(E)} - u)\boldsymbol{\omega}\cdot\boldsymbol{n} \tag{3}$$

holds for every element $E \in \mathscr{E}$ and every test function $\boldsymbol{\omega} \in V_h^d$. Here, $u^\star_\chi$ is a numerical flux function which could carry a variety of forms; in this work, we use one-sided fluxes for all intraphase faces and a multivalued interphase flux which takes into account the jump data g_{ij} on Γ_{ij} in (2):

$$u^\star_\chi := \begin{cases} u^- & \text{on any intraphase face,} \\ \lambda u^- + (1-\lambda)(u^+ + g_{\chi i}) & \text{on } \Gamma_{\chi i} \text{ if } \chi < i, \\ \lambda(u^- - g_{i\chi}) + (1-\lambda)u^+ & \text{on } \Gamma_{i\chi} \text{ if } \chi > i, \\ u^- & \text{on } \Gamma_N, \\ g_\partial & \text{on } \Gamma_D. \end{cases} \tag{4}$$

(See Figure 1 for a schematic illustration.) Note that the flux is multivalued on interphase faces — on these faces, the interfacial jump condition $[\![u]\!] = g_{ij}$ on Γ_{ij} is taken into account as follows: when an element "reaches across" the interface to evaluate the trace of u on the other side, the trace value is compensated by the jump data to correctly account for the intended discontinuity in the solution. Note also that interfacial fluxes are weighted through a convex combination parameter $\lambda \in [0, 1]$, which can vary from face to face. If $\lambda = 0$, then the numerical flux is

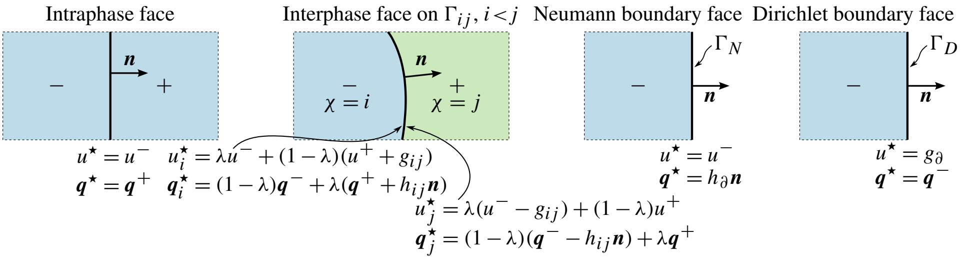

Intraphase face
Interphase face on Γ_{ij}, $i<j$
Neumann boundary face
Dirichlet boundary face
$-$
$\boldsymbol{n}$
$+$
$-$
$\chi = i$
$+$
$\chi = j$
Γ_N
Γ_D
$u^\star = u^-$
$\boldsymbol{q}^\star = \boldsymbol{q}^+$
$u_i^\star = \lambda u^- + (1-\lambda)(u^+ + g_{ij})$
$\boldsymbol{q}_i^\star = (1-\lambda)\boldsymbol{q}^- + \lambda(\boldsymbol{q}^+ + h_{ij}\boldsymbol{n})$
$u_j^\star = \lambda(u^- - g_{ij}) + (1-\lambda)u^+$
$\boldsymbol{q}_j^\star = (1-\lambda)(\boldsymbol{q}^- - h_{ij}\boldsymbol{n}) + \lambda\boldsymbol{q}^+$
$u^\star = u^-$
$\boldsymbol{q}^\star = h_\partial \boldsymbol{n}$
$u^\star = g_\partial$
$\boldsymbol{q}^\star = \boldsymbol{q}^-$

Figure 1. Schematic of the numerical flux functions $u^\star$ and $\boldsymbol{q}^\star$ defined by (4) and (6). Except for interphase faces, the flux is single-valued; on interphase faces, the flux is multivalued so as to incorporate the interfacial jump conditions $[\![u]\!] = g_{ij}$ and $[\![\boldsymbol{q}\cdot\boldsymbol{n}]\!] = h_{ij}$ on Γ_{ij}, $i < j$. A plus or minus sign denotes the elemental value on the right or left of the face, respectively; e.g., for a point x on the face, $u^\pm(x) = \lim_{\epsilon\to 0^+} u(x \pm \epsilon\boldsymbol{n})$.

sourced solely from the right element's trace u^+; if $\lambda = 1$, it is sourced solely from the left element's trace u^-. Choosing the correct value of λ is the essential subject of this work, and we will return to this topic shortly. Note also that the numerical flux $u^\star$ equals g_∂ on all Dirichlet boundary faces, and equals the boundary trace on all Neumann boundary faces.

Second, we define a discrete approximation to $\boldsymbol{q} \in V_h^d$, which is essentially $\boldsymbol{\eta}$ multiplied by the local ellipticity coefficient μ. To do so, we define $\boldsymbol{q}$ as the L^2 projection of the function $\mu\boldsymbol{\eta}$, i.e., $\boldsymbol{q} \in V_h^d$ is the unique piecewise polynomial function such that

$$\int_E \boldsymbol{q}\cdot\boldsymbol{\omega} = \int_E \mu\boldsymbol{\eta}\cdot\boldsymbol{\omega} \tag{5}$$

holds for every element $E \in \mathscr{E}$ and every test function $\boldsymbol{\omega} \in V_h^d$. In the case that μ is piecewise constant, calculating this L^2 projection is a particularly simple matter of multiplying $\boldsymbol{\eta}$ by a scalar. When μ is variable, one possible simplification is to avoid the L^2 projection and replace it with a symmetry-preserving interpolant operator approximating $\mu\boldsymbol{\eta}$; this approach, however, may fail to attain optimal high-order accuracy, especially when μ is not particularly smooth. In this work, $\boldsymbol{q}$ is always computed through an L^2 projection using sufficiently high-order accurate quadrature schemes to evaluate the product of the three functions on the right-hand side of (5).

In the third step, we consider the weak formulation for computing the divergence of $\boldsymbol{q}$. This proceeds similarly to defining the discrete gradient of u, except numerical fluxes act in the opposite direction. (For simplicity of presentation, the following numerical flux for $\boldsymbol{q}$ is vector-valued; however, only the normal component of the flux is used.) Given $\boldsymbol{q} \in V_h^d$, define $w \in V_h$ as the discrete divergence of $\boldsymbol{q}$ such that

$$\int_E wv = -\int_E \boldsymbol{q}\cdot\nabla v + \int_{\partial E} v\boldsymbol{q}^\star_{\chi(E)}\cdot\boldsymbol{n}$$

holds for every test function $v \in V_h$ and every element $E \in \mathscr{E}$ with phase $\chi(E)$. Here, the numerical flux is defined by (see also Figure 1)

$$\boldsymbol{q}^\star_\chi := \begin{cases} \boldsymbol{q}^+ & \text{on any intraphase face,} \\ (1-\lambda)\boldsymbol{q}^- + \lambda(\boldsymbol{q}^+ + h_{\chi i}\boldsymbol{n}) & \text{on } \Gamma_{\chi i} \text{ if } \chi < i, \\ (1-\lambda)(\boldsymbol{q}^- - h_{i\chi}\boldsymbol{n}) + \lambda\boldsymbol{q}^+ & \text{on } \Gamma_{i\chi} \text{ if } \chi > i, \\ h_\partial \boldsymbol{n} & \text{on } \Gamma_N, \\ \boldsymbol{q}^- & \text{on } \Gamma_D. \end{cases} \tag{6}$$

As in the numerical flux for $u^\star$, the interfacial jump condition $[\![\boldsymbol{q}\cdot\boldsymbol{n}]\!] = h_{ij}$ on Γ_{ij} is taken into account via the multivalued interfacial flux, such that whenever an element reaches across the interface, the neighboring element's trace is compensated by h_{ij} to correctly put it in the context of the source element.

Finally, it is often necessary to add penalty stabilization terms to ensure the well-posedness of the discrete problem [9; 34]. These terms weakly impose continuity between neighboring element polynomials and weakly impose Dirichlet boundary conditions. We classify them according to three types: boundary (τ_D), intraphase (τ_i), and interphase (τ_{ij}) penalization parameters. Let $E_g : V_h \to V_h$ be the operator such that, for each $u \in V_h$,

$$\int_\Omega E_g(u)v = \sum_i \int_{\Gamma_i} \tau_i [\![u]\!][\![v]\!] + \sum_{i<j} \int_{\Gamma_{ij}} \tau_{ij}([\![u]\!] - g_{ij})[\![v]\!] + \int_{\Gamma_D} \tau_D(u^- - g_\partial)v^- \tag{7}$$

holds for every test function $v \in V_h$; here, Γ_i denotes the set of intraphase faces in phase i. The penalization operator E_g is added to the discrete Laplacian to define the final linear system discretizing (1). In general, the values of τ_i, τ_{ij}, and τ_D could vary from face to face. Generally speaking:

- Strictly positive parameters are sufficient to ensure well-posedness of the final linear system (i.e., it has trivial kernel, or a one-dimensional kernel in the case Γ_D is empty). However, this is not a necessary condition. For example, on a regular Cartesian grid, with purely one-sided intraphase numerical fluxes for $u^\star$ and $\boldsymbol{q}^\star$ (as used here), one can set the intraphase penalty to zero, $\tau_i = 0$ [21]. On the other hand, a penalty parameter which is too large in value can impact discretization accuracy as well as conditioning and multigrid performance.
- If Γ_D is nonempty, then τ_D should be positive to ensure well-posedness.
- Although LDG schemes do not require any particular lower bound on τ, for consistent discretization behavior, a variety of different methods can be used to show that a nonzero penalty parameter should scale inversely proportional to the mesh size h, i.e., $\tau = \mathbb{O}(h^{-1})$ as $h \to 0$. Such a scaling is consistent with other forms of DG methods for elliptic problems, such as symmetric interior penalty methods which require $\tau \geq C/h$ for well-posedness. For anisotropic meshes, one can be more precise and say the value of τ on a particular mesh

face should scale proportionally to the measure of the face divided by the measure of the elements on either side.

- To ensure correct scaling with ellipticity coefficient, penalty parameters should also scale with the local value of viscosity. For example, $\tau_D \sim \mu^-$ and $\tau_i \sim \mu_i$. For interphase penalty parameters, τ_{ij} should scale linearly with an appropriate function of μ^- or μ^+, i.e., the trace values of μ_i or μ_j on either side of the interface.
- One can also choose to scale τ with the polynomial degree (see, e.g., [11]), which can be important for studying DG methods with very high-degree polynomials; however in this work, we consider only moderate-order polynomials and neglect this effect.

Further details on the precise values of the penalty parameters are deferred to the presented results in Section 4.

To summarize the steps of the LDG construction, one (i) computes the discrete gradient of $u \in V_h$ to find $\boldsymbol{\eta}$, (ii) finds the L^2 projection of $\mu\boldsymbol{\eta}$ to define $\boldsymbol{q}$, (iii) computes the discrete divergence of $\boldsymbol{q}$, (iv) adds penalty stabilization terms, and finally (v) sets the result equal to the L^2 projection of the right-hand side, f. We refer the reader to [44] for an in-depth derivation[3] and instead state the final result wherein the auxiliary variable $\boldsymbol{q}$ is eliminated: u solves the linear problem

$$\left(\sum_{i=1}^{d} G_i^T M_\mu G_i\right)u + ME_0u = M\mathbb{P}_{V_h}(f) + J_h(h_{ij}, h_\partial) + J_g(g_{ij}, g_\partial) \tag{8}$$

where:

- $G = (G_1, \dots, G_d) : V_h \to V_h^d$ is the discrete gradient operator that implements the construction of $\boldsymbol{\eta}$ in (3) and (4) assuming homogeneous source data.
- M is the symmetric positive definite block-diagonal mass matrix and M_μ is its μ-weighted counterpart such that
$$u^T M_\mu v = \sum_i \int_{\Omega_i} u\mu_i v$$
holds for all functions $u, v \in V_h$. (Here, we are slightly abusing notation by consider u and v as both functions in V_h and as coefficient vectors in the chosen basis.[4]) In particular, we note that the L^2 projection of μu for $u \in V_h$ is given by $M^{-1}M_\mu u$.

[3] The cited work mainly considers the case of piecewise constant μ, but its results can be straightforwardly generalized to applications using the L^2 projection of $\mu\boldsymbol{\eta}$.

[4] A tensor-product Gauss–Lobatto nodal basis is employed in this work, suitable for low-to-moderate degree DG methods. The analysis presented in this paper holds for any chosen basis, provided it is understood that every basis-dependent matrix (e.g., the mass matrix M or its μ-weighted counterpart M_μ) are defined consistently relative to the chosen basis.

- E_0 is the matrix implementing the penalty stabilization terms in (7), assuming homogeneous Dirichlet boundary and jump data.
- $\mathbb{P}_{V_h}(f)$ is the L^2 projection of f onto V_h, which in many applications could simply be approximated by a nodal interpolant of f.
- The terms J_h and J_g collect the entire influence of the jump data g_{ij}, h_{ij} and boundary data g, h, including that which is incorporated in penalization in (7) and the numerical fluxes (4) and (6).

One can show the linear system (8) is symmetric positive semidefinite (positive definite if Γ_D is nonempty and $\tau_D > 0$) and is amenable to conjugate gradient methods preconditioned by multigrid algorithms. The subject of this paper is to determine how to choose the value of λ on interfacial faces so as to optimize multigrid performance for the cases of large jumps in ellipticity coefficient.

3. Multigrid methods

The multigrid algorithms used in this work follow the operator-coarsening schemes presented by Fortunato et al. [29], except with two important modifications: (i) the methods are generalized to handle variable viscosity, and (ii) penalty parameters are halved in strength each level down the mesh hierarchy. (Further details on these modifications are provided shortly.) These multigrid methods are based on the idea of separately coarsening the discrete gradient operator G and discrete divergence operator $D = -\operatorname{adj}(G)$ across each level of the multigrid hierarchy; these coarsened operators are then multiplied together to find the discrete Laplacian operator on each level. Through this approach, one obtains a multigrid scheme which is equivalent in function to a purely geometric multigrid method — i.e., one in which the mesh is explicitly built, and the LDG discretization is explicitly formulated, on every level of the hierarchy. In particular, the approach automatically constructs coarsened operators which are consistent with the chosen numerical fluxes on the finest mesh; for example, if weighted numerical fluxes are used on the finest mesh, the same weighting is automatically inherited by the coarse-mesh operators.

3.1. ***Operator-coarsening multigrid.*** Here, a brief description of the multigrid algorithms is given; for further details and motivation, the reader is referred to [29]. The essential components of the multigrid methods are as follows:

- *Mesh hierarchy*. In this work, quadtrees and octrees are used to define the finest mesh or the background grid in the case of implicitly defined meshes (see Section 5). The tree structure naturally defines a hierarchical procedure for agglomerating elements to create a hierarchy of nested meshes for use in h-multigrid; generally, the mesh is spatially coarsened by a factor of two in each dimension on each level. Importantly, element agglomeration is only permitted between elements of the same phase — as such, the interface of an elliptic interface problem is sharply preserved throughout the entire multigrid hierarchy. (An example is shown in Figure 4.8 of [29].)

- *Interpolation operator.* The interpolation operator I^h_{2h} transfers a piecewise polynomial function on a coarse mesh to a piecewise polynomial function on the fine mesh. In the present setting, I^h_{2h} is naturally defined by injection: $(I^h_{2h}u)|_{E_f} = u|_{E_c}$, where E_f is a fine mesh element and $E_c \supseteq E_f$ is its corresponding coarse mesh element.
- *Restriction operator.* The restriction operator R^{2h}_h is defined to be the adjoint of the interpolation operator. Equivalently, for a piecewise polynomial function u on a fine mesh, $R^{2h}_h u$ is defined as the L^2 projection of u onto the coarse mesh. It is related to the interpolation operator via $R^{2h}_h = M^{-1}_{2h}(I^h_{2h})^T M_h$ where M_h and M_{2h} are the mass matrices on the fine and coarse meshes, respectively, and $(I^h_{2h})^T$ is the transpose of the interpolation operator matrix.
- *Coarsening of a general operator.* Given an operator $A : V_h \to V_h$ defined on a fine mesh, its coarsened counterpart on a coarse mesh is defined variationally, such that $\mathscr{C}(A) : V_{2h} \to V_{2h}$ satisfies

  $$(\mathscr{C}(A)u, v)_{V_{2h}} = (AI^h_{2h}u, I^h_{2h}v)_{V_h}$$

  for all $u, v \in V_{2h}$; here $(\cdot, \cdot)_{V_h}$ denotes the standard inner product on V_h. Equivalently, as a matrix acting on coefficient vectors in the chosen basis, $\mathscr{C}(A) = R^{2h}_h A I^h_{2h}$.

In [29], operator-coarsening multigrid methods are derived for single-phase Poisson problems $-\nabla^2 u = f$ as follows. On the finest mesh, the LDG discretization results in the linear system $(-DG + \tau E)u = \mathbb{P}_{V_h} f$, where G is the discrete gradient operator, $D = -\operatorname{adj}(G) = -M^{-1}G^T M$ is the discrete divergence operator, and E is a penalty stabilization operator. The coarse-mesh operator, e.g., as would be used in a multigrid V-cycle, is then defined as $-\mathscr{C}(D)\mathscr{C}(G) + \tau\mathscr{C}(E)$. In particular, it is shown that this coarse-mesh operator is identical to the one which would be obtained if an LDG discretization with the same numerical fluxes was directly applied to the coarse mesh problem. However, one advantage to constructing the coarse-mesh operator via the $\mathscr{C}$ functional is that doing so does not require the coarse mesh problem to be explicitly discretized; i.e., the coarse mesh does not need to be explicitly found (instead, it is implicitly formed via the interpolation/element agglomeration hierarchy), quadrature schemes for coarse mesh elements do not need to be computed, coarse lifting and penalty operators and L^2 projections do not need to be constructed, and so forth. Two modifications to the operator-coarsening approach are made in the present work:

(1) In addition to coarsening the discrete gradient and penalty operators, the viscosity-weighted L^2 projection operator is also coarsened. Let $\Theta_\mu : V_h \to V_h$ be defined such that $\Theta_\mu u$ is the L^2 projection of μu onto V_h, i.e., $(\Theta_\mu u, v)_{V_h} = (\mu u, v)_{V_h}$ holds for all $v \in V_h$. Then, the fine mesh discrete elliptic interface problem derived in (8) essentially reads as

$$(-D\Theta_\mu G + E_\tau)u = \mathbb{P}_{V_h} f + J(h) + J(g).$$

Here, E_τ is the penalty operator with penalty parameters for intraphase, interphase, and boundary faces, baked inside its definition. The coarse-mesh operator is defined as

$$-\mathscr{C}(D)\mathscr{C}(\Theta_\mu)\mathscr{C}(G) + \tfrac{1}{2}\mathscr{C}(E_\tau). \tag{9}$$

Using similar methods as was shown in [29], one can show that this coarse-mesh operator is equivalent to that which would be obtained if the coarse-mesh problem was explicitly discretized with LDG. In particular, the coarsened μ-weighted identity operator $\mathscr{C}(\Theta_\mu)$ effectively coarsens the influence of μ on the fine mesh to larger and larger elements throughout the hierarchy, consistently with performing an L^2 projection of μ multiplied by piecewise polynomial functions on the coarse meshes.

(2) The second modification concerns the choice of penalty parameters on coarse-level meshes. In [29], penalty parameters were chosen for the finest-level mesh and these were left unaltered throughout the entire hierarchy. However, in the present work it was found that this is a suboptimal strategy and can lead to worsening V-cycle performance as the fine mesh problem is refined. Instead, a simple fix is to appropriately adjust the value of the penalty parameters τ on each level to reflect the observation that the effective h value entering the guideline penalty parameter scaling of $\tau \sim \mu/h$ is doubling every time the mesh is coarsened. This modification is implemented via the factor[5] of $\frac{1}{2}$ in (9).

Algorithm 1 summarizes the essential construction of the coarse-mesh operators, to be applied recursively down the mesh hierarchy; here M_h is the mass matrix on a fine mesh, $M_{\mu,h}$ is its μ-weighted counterpart, G_h is the discrete gradient operator, $\widetilde{E}_h := M_h E_h$ is the penalty operator premultiplied by the mass matrix, and A_{2h} defines the final overall operator for the elliptic interface problem on the coarse mesh (corresponding to the discretization of the operator $-\nabla \cdot (\mu \nabla)$ on the coarse mesh, left-multiplied by the coarse-mesh's mass matrix).

3.2. ***Multigrid preconditioned conjugate gradient.*** The V-cycle preconditioned conjugate gradient method employed in this work is outlined in Algorithm 2. In particular:

[5] In more sophisticated settings using adaptive mesh refinement, the factor of $\frac{1}{2}$ would take into account the possibility elements may change size by differing factors.

```
$M_{2h} := (I_{2h}^h)^T M_h I_{2h}^h$
$M_{\mu,2h} := (I_{2h}^h)^T M_{\mu,h} I_{2h}^h$
$G_{2h} := M_{2h}^{-1} (I_{2h}^h)^T M_h G_h I_{2h}^h$
$\widetilde{E}_{2h} := \frac{1}{2} (I_{2h}^h)^T \widetilde{E}_h I_{2h}^h$
$A_{2h} := G_{2h}^T M_{\mu,2h} G_{2h} + \widetilde{E}_{2h}$
```

Algorithm 1. Construction of coarse-mesh operators, given fine-mesh operators M_h, $M_{\mu,h}$, G_h, and $\widetilde{E}_h$.

```
if $\mathscr{E}_h$ is the bottom level then
    Solve $A_h x_h = b_h$ with bottom solver
else
    Apply smoother $\nu$ times
    $r_{2h} := (I_{2h}^h)^T (b_h - A_h x_h)$
    $x_{2h} := V(\mathscr{E}_{2h}, 0, r_{2h})$
    $x_h \leftarrow x_h + I_{2h}^h x_{2h}$
    Apply smoother (in reverse ordering) $\nu$ times
return $x_h$
```

Algorithm 2. Multigrid V-cycle $V(\mathscr{E}_h, x_h, b_h)$ on a mesh $\mathscr{E}_h$ with ν pre- and postsmoothing steps.

- A multicolored block Gauss–Seidel iteration is used as the relaxation/smoothing method. In a setup phase, a graph-coloring algorithm is applied to the element connectivity graph defined by the blockwise sparsity of the operator A on each level of the hierarchy. The algorithm approximately finds the minimum number of colors needed using a DSATUR algorithm [38]; on a standard Cartesian grid, with one-sided intraphase fluxes, this approach recovers the optimal red-black ordering associated with a standard 5-point (2D) or 7-point (3D) Laplacian stencil. The primary reason for coloring the Gauss–Seidel method is to achieve parallel speedup in a multithreaded environment, wherein all elements of the same color can be processed in parallel.

- In the case of large, three-dimensional studies, in addition to multithreading, a standard domain decomposition approach using MPI is used. In this case, each subdomain applies Gauss–Seidel with a ghost layer of elemental values which are frozen at the beginning of each iteration — as such, the relaxation method is a processor-block Gauss–Seidel method [1], which in the limit of one element per processor decays to a block-Jacobi iteration. Since block-Jacobi relaxation is not convergent for DG methods, a small amount of damping is applied. In brief, for the presented three-dimensional studies, the relaxation method is a processor-block, damped, elementwise-block Gauss–Seidel iteration with damping parameter $\omega = 0.875$, chosen through experiment so as to ensure reliable convergence using approximately the smallest damping possible.

- Three pre- and postsmoothing steps are applied in the V-cycle. By reversing the ordering of the Gauss–Seidel sweep in the postsmoothing phase, the associated V-cycle linear operator is symmetric.

The application of one V-cycle to approximately solve the system $Ax = b$ with initial guess zero results in a linear operator acting on b; the corresponding matrix is denoted in the following by V. To solve the linear systems arising from the multiphase elliptic interface problems considered in this work, a single V-cycle is used as a preconditioner in the conjugate gradient method. According to standard convergence theory, the two-norm condition number of VA can be used to bound the number of iterations required to reduce the residual by a given tolerance. Consequently, the primary metric used in this work to assess the efficacy of multigrid performance is $\kappa(VA)$; for an optimally performing multigrid method, $\kappa(VA)$ should be reasonably close to unity and bounded as $h \to 0$.

4. One-dimensional analysis

In this section, we examine the role of weighted interfacial fluxes on multigrid performance for a one-dimensional, two-phase, constant-coefficient elliptic interface problem. Although only in one spatial dimension, the observed behavior in accuracy, conditioning, and convergence rates is reflective of what also occurs in two- and higher-dimensional problems with more complex interface geometry.

Throughout this section, let $\Omega = (0, 1)$ be the unit interval divided into a middle interior phase $\Omega_1 = \left(\frac{1}{4}, \frac{3}{4}\right)$ and an exterior phase $\Omega_2 = \left(0, \frac{1}{4}\right) \cup \left(\frac{3}{4}, 1\right)$, and let $\Gamma = \left\{\frac{1}{4}, \frac{3}{4}\right\}$ denote the interface between Ω_1 and Ω_2. We consider the elliptic interface problem with Dirichlet boundary conditions

$$\begin{cases} -\mu_i \nabla^2 u = f_i & \text{in } \Omega_i, \\ [\![u]\!] = g & \text{on } \Gamma, \\ [\![\mu \nabla u \cdot \boldsymbol{n}]\!] = h & \text{on } \Gamma, \\ u = g_\partial & \text{on } \partial\Omega, \end{cases} \tag{10}$$

where μ_1 shall in the following have small ($\ll 1$), unit, and large ($\gg 1$) values, while μ_2 is always held fixed at $\mu_2 = 1$. The discretization employs the LDG schemes of Section 2 with the following characteristics:

- a mesh consisting of $n = 1/h$ equal-sized elements, with n divisible by four so as to ensure the interface is situated between elements,
- polynomial degree[6] $p = 3$,

[6] For simplicity of presentation, results in one dimension are shown solely for $p = 3$; similar behavior is observed for other tested values of p between 1 and 10.

- boundary faces on $\partial\Omega$ carry a penalty parameter $\tau_D = \mu_2(p+1)h^{-1}$,
- following the motivation in Section 1.1, interfacial faces on Γ carry a penalty parameter $\tau_{12} = \min(\mu_1, \mu_2)(p+1)h^{-1}$, and
- all other faces have zero penalty parameter.

4.1. ***General behavior and smoothing performance.*** In the following set of tests we fix the number of elements $n = 16$ and consider a smooth test problem in which the source data f, g, h, and g_∂ in (10) are generated by the exact solution

$$u(x) = \begin{cases} \sin 4\pi(x-0.1) & \text{in } \Omega_1, \\ \cos 4\pi(x-0.1) & \text{in } \Omega_2. \end{cases} \tag{11}$$

In the following two test cases, the interior phase's viscosity coefficient is set equal to $\mu_1 = 10^{-5}$.

First, we consider a case in which the weighting on interfacial numerical fluxes is chosen suboptimally. Specifically, for the two interfacial faces located at $\Gamma = \{\frac{1}{4}, \frac{3}{4}\}$, the convex combinations in (4) and (6) use equal weighting with $\lambda = 0.5$, reflecting a central flux. This choice results in an extremely inaccurate discrete solution, poor multigrid performance, and poor conditioning of the V-cycle preconditioned linear system, as examined in Figure 2. In particular, Figure 2, top row, illustrates the piecewise-cubic discrete solution and its error, showing a pronounced numerical boundary layer.[7] The condition number of the preconditioned system VA is approximately 5200; inspection of the spectrum of VA, consisting of $n(p+1) = 64$ eigenvalues (see Figure 2, center left) shows that the smallest eigenvalue $\lambda_{\min} \approx 8.5 \times 10^{-4}$ is the main contributor to the poor condition number; the corresponding piecewise-cubic eigenfunction is essentially identical (up to normalization) to the error profile shown in Figure 2, top right. Thus, in this particular example, the mode which contributes to poor accuracy happens to be the same mode which multigrid most ineffectively handles.

To examine multigrid performance, one possible approach is to assess whether the associated relaxation method exhibits ideal smoothing properties. Let S denote the action of three iterations of the block Gauss–Seidel relaxation method, such that Su approximately solves $Ax = b$ with initial guess u and right-hand side equal to zero. S should have at least three desirable properties: (i) all of its eigenvalues should have absolute value not greater than one, (ii) modes which are spatially high-frequency should be damped quickly (i.e., eigenvalue close to zero), and (iii) modes which are damped slowly (i.e., eigenvalues with magnitude close to 1) should be spatially low-frequency so that they can be effectively handled by coarser grids. For

[7] The boundary layer is more pronounced on one of the interfaces owing to the asymmetry introduced by the one-sided intraphase face fluxes; if these are switched in direction, the boundary layer moves to the right interface.

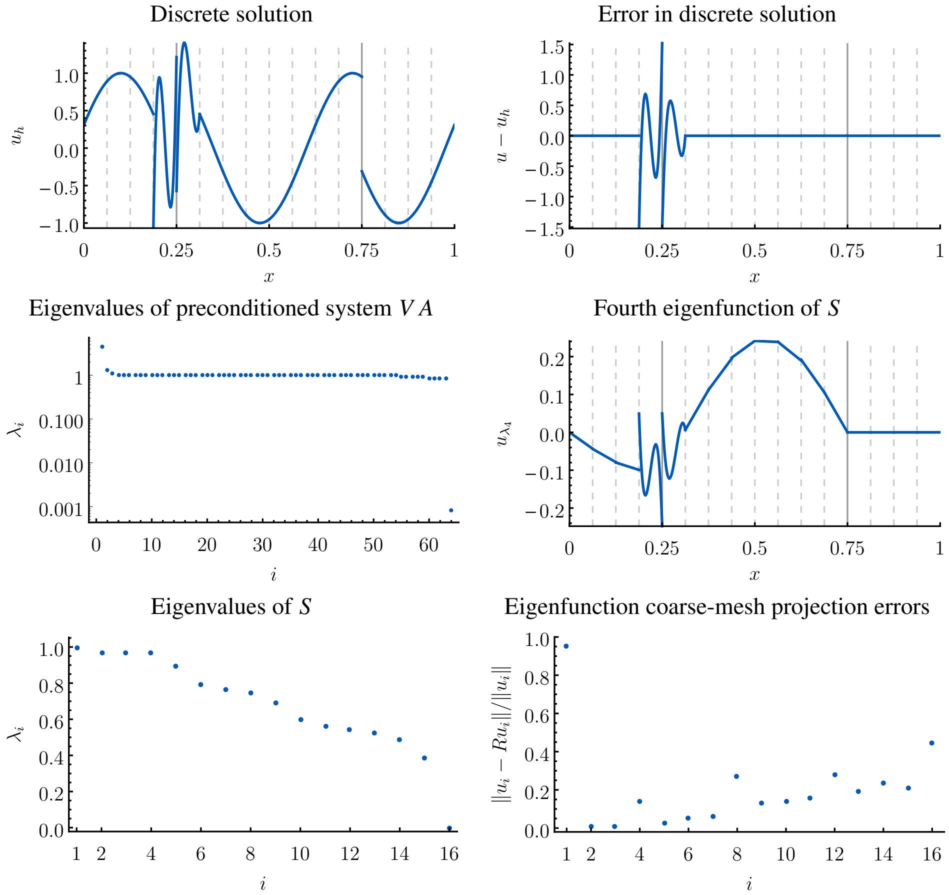


Figure 2. Solution accuracy and characteristics of multigrid performance for a two-phase, constant-coefficient elliptic interface problem in one dimension, wherein $\Omega_1 = \left(\frac{1}{4}, \frac{3}{4}\right)$ has viscosity coefficient $\mu_1 = 10^{-5}$ and $\Omega_2 = \left(0, \frac{1}{4}\right) \cup \left(\frac{3}{4}, 1\right)$ has coefficient $\mu_2 = 1$, using the suboptimal choice of central flux weighting on interphase faces. In the plots of the top row and center right, the piecewise-cubic polynomial functions are graphed in addition to dashed/solid vertical lines indicating the boundaries between the $n = 16$ elements; solid lines indicate the interface.

the present test problem, the largest[8] n eigenvalues of S are displayed in Figure 2, bottom left, and show that all eigenvalues are real and lie in the unit interval. To examine whether an eigenfunction is spatially low-frequency, a simple method is to test how similar the function is to its projection onto a coarse mesh. This can be accomplished by examining the relative error in $u \in V_h$ versus $R_h^{2h} u \in V_{2h}$ where

[8] In one dimension, experiments indicate that at most n eigenvalues of S are nonzero, while the remaining np eigenvalues are exactly zero. This is in part attributed to the elementwise block action of the smoother.

R_h^{2h} is the restriction operator used by multigrid. Ideally, the eigenfunctions of S with large eigenvalue should have very small relative errors in their coarse-mesh projections; however, the results in Figure 2, bottom right, show that the fourth, eighth, and especially first mode are outliers in this regard. In fact, the eigenfunction of S with largest eigenvalue is the same one encountered before, identical in profile to Figure 2, top right. This function is clearly not smooth. As another example, mode 4, i.e., the eigenfunction of S with fourth-largest eigenvalue, is shown in Figure 2, center right. Once more we see a high-frequency mode, which is not effectively damped by the smoother.

In summary, we see that an unwise choice of weighting in the numerical flux results in discrete solutions of unacceptable accuracy as well as poor multigrid performance. In this case, the poor multigrid performance can be attributed to an ineffective smoother wherein particular high-frequency modes are damped very slowly.

Next, we examine precisely the same problem, except now the weighting on interfacial numerical fluxes is chosen to bias as motivated in Section 1.1. In particular, we set the convex combination in (4) and (6) to be such that $\lambda = 0$. Figure 3 presents a similar analysis as was shown in Figure 2, and demonstrates significantly improved behavior. The discrete solution is now four orders of magnitude more accurate, and the condition number of the V-cycle preconditioned system is approximately 1.4 (compared to the value of 5200 in the previous case). Comparison of Figure 3, bottom right, with Figure 2, bottom right, shows that the smoother performance has also markedly improved. Indeed, the first four modes of the smoother (those with largest eigenvalues) are illustrated in Figure 3, center left, all of which are relatively smooth modes for this problem having a mesh of $n = 16$ elements.

4.2. *Optimal choice of weighting.* To investigate the role of weighted interfacial fluxes across a range of ellipticity coefficient jump ratios, we examine two metrics as a function of λ: (i) the maximum-norm error in the discrete solution, and (ii) the condition number of the V-cycle preconditioned system. Using the same two-phase, constant-coefficient elliptic interface problem of the previous section, and the same exact solution given in (11), Figure 4 shows results for five different values of μ_1, specifically 10^{-8}, 10^{-4}, 1, 10^4, and 10^8. In the graphs, the convex combination is varied from one-sided in one direction, $\lambda = 0$, to central, $\lambda = 0.5$, to one-sided in the other direction, $\lambda = 1$. Specifically, λ takes on values 10^{-k} and $1 - 10^{-k}$ for $k = \infty, 10.5, 10, 9.5, 9, \dots, 1.5, 1$ along with the central value $\lambda = 0.5$; in the plots, these are shown on a quasilogarithmic scale. A number of conclusions can be drawn from Figure 4:

- Optimal errors and conditioning can be attained provided the weighted fluxes are sufficiently biased in the appropriate direction. If the viscosity ratio is less than one, λ should be chosen closer to zero, which corresponds to the

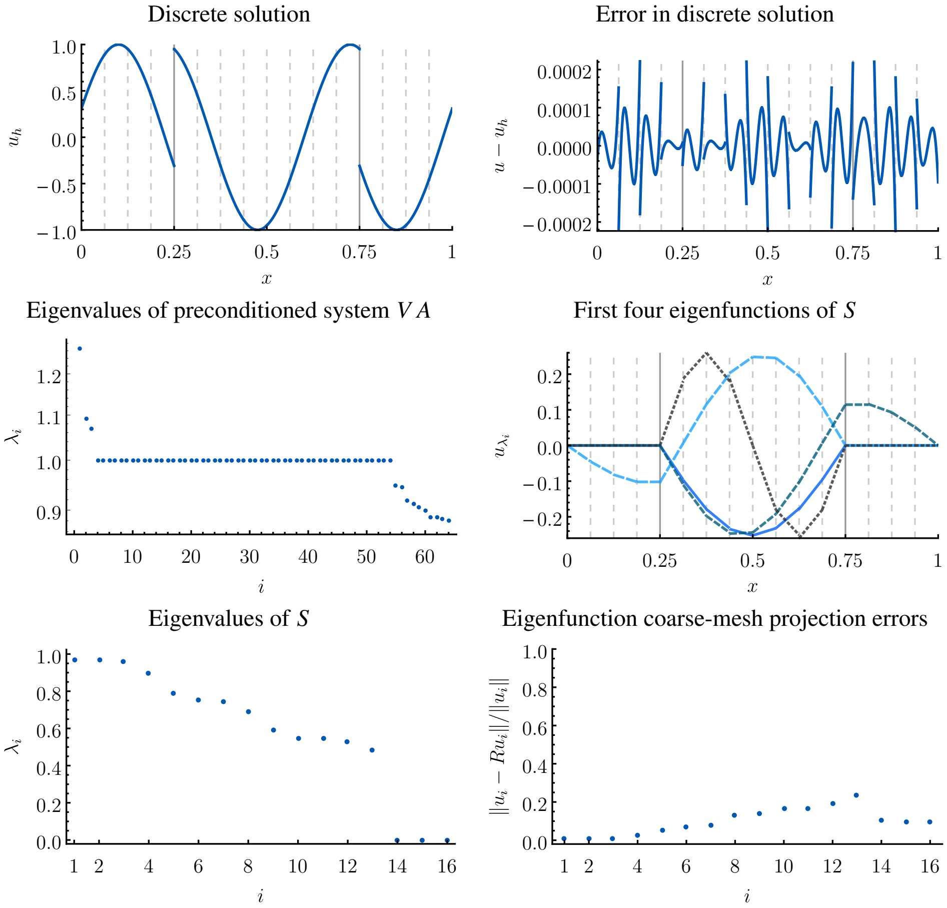


Figure 3. Solution accuracy and characteristics of multigrid performance for a two-phase, constant-coefficient elliptic interface problem in one dimension, wherein $\Omega_1 = \left(\frac{1}{4}, \frac{3}{4}\right)$ has viscosity coefficient $\mu_1 = 10^{-5}$ and $\Omega_2 = \left(0, \frac{1}{4}\right) \cup \left(\frac{3}{4}, 1\right)$ has coefficient $\mu_2 = 1$, using an ideal choice of weighting for interfacial numerical fluxes. In the plots of the top row and center right, the piecewise-cubic polynomial functions are graphed in addition to dashed/solid vertical lines indicating the boundaries between the $n = 16$ elements; solid lines indicate the interface.

numerical flux for $u^\star$ biasing towards phase Ω_2 and $\boldsymbol{q}^\star$ biasing toward Ω_1 (see (4) and (6)). If the viscosity ratio is greater than one, λ should be chosen closer to unity, thereby biasing in the opposite direction. In both cases, the direction of weighting is consistent with the motivation given in Section 1.1, i.e., the numerical flux for $u^\star$ should bias towards the more viscous phase, and that for $\boldsymbol{q}^\star$ should bias towards the less viscous phase.

- The condition number of the V-cycled preconditioned operator is minimized when $\lambda \lessapprox \sqrt{\mu_1/\mu_2}$ if $\mu_1/\mu_2 < 1$, or $\lambda \gtrapprox 1 - \sqrt{\mu_2/\mu_1}$ if $\mu_1/\mu_2 > 1$.

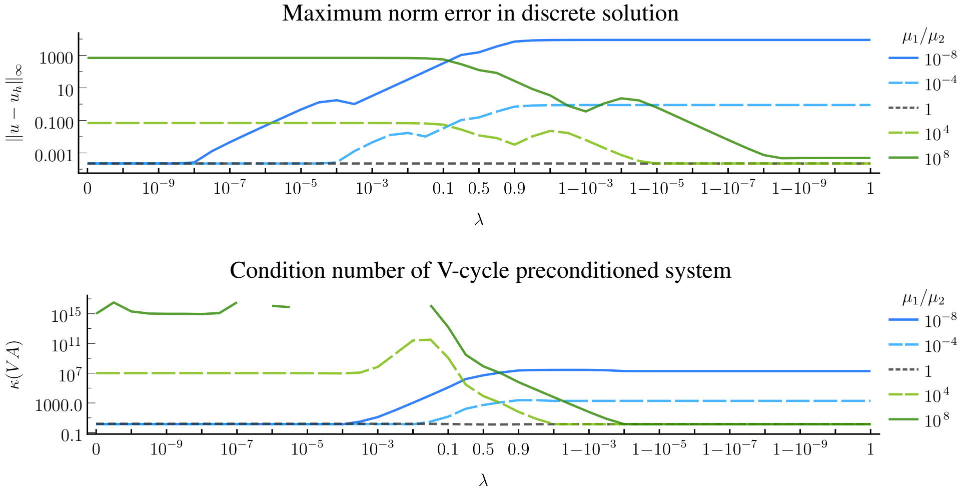


Figure 4. Solution accuracy and multigrid performance as a function of interfacial flux weighting for a two-phase, constant-coefficient elliptic interface problem in one dimension, wherein $\Omega_1 = \big(\frac{1}{4}, \frac{3}{4}\big)$ has viscosity coefficient μ_1 and $\Omega_2 = \big(0, \frac{1}{4}\big) \cup \big(\frac{3}{4}, 1\big)$ has coefficient μ_2, with the viscosity ratio as indicated. Here, the λ parameter is varied corresponding to a one-sided flux ($\lambda = 0$), to a central flux ($\lambda = 0.5$), and to a one-sided flux in the opposite direction ($\lambda = 1$); note the quasilogarithmic scale of λ values on the horizontal axis.

- The maximum norm error in the discrete solution is minimized when $\lambda \lessapprox \mu_1/\mu_2$ if $\mu_1/\mu_2 < 1$, or $\lambda \gtrapprox 1 - \mu_2/\mu_1$ if $\mu_1/\mu_2 > 1$. In fact, the results for $\mu_1 \in \{10^4, 10^8\}$ show that one can do slightly better by biasing slightly more by requiring $\lambda \gtrapprox 1 - C\mu_2/\mu_1$ where $C \approx 0.1$.
- In all cases, a purely one-sided weighting strategy matches the best possible solution error and condition number, i.e., if $\mu_1/\mu_2 < 1$, then set $\lambda = 0$, and if $\mu_1/\mu_2 > 1$, then set $\lambda = 1$.

These observations closely match the strategy of harmonic weighting used in a variety of prior work, as surveyed in Section 1.2. In particular, harmonic weighting chooses λ in (4) and (6) such that

$$\lambda = \frac{\mu^-}{\mu^- + \mu^+}, \tag{12}$$

where $\mu^\pm$ denotes the trace values on either side of an interphase mesh face. Other possibilities suggested in prior work include the weaker biasing choice of $\lambda \approx \sqrt{\mu^-}/\big(\sqrt{\mu^-} + \sqrt{\mu^+}\big)$ [19]; however, as shown above, and although this attains near-optimal preconditioned condition numbers, markedly better solution errors can be obtained with stronger biasing. According to our results (and also those observed in Figure 7 below), one can attain marginally better results by biasing the weights stronger than a harmonic weighting. To this end, one could choose a

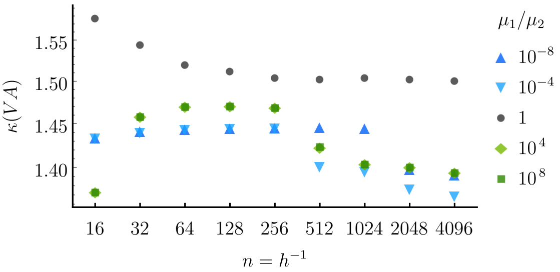


Figure 5. Multigrid performance under the action of mesh refinement for a two-phase, constant-coefficient elliptic interface problem in one dimension, wherein $\Omega_1 = \big(\frac{1}{4}, \frac{3}{4}\big)$ has viscosity coefficient μ_1 and $\Omega_2 = \big(0, \frac{1}{4}\big) \cup \big(\frac{3}{4}, 1\big)$ has coefficient μ_2, with the viscosity ratio as indicated. Here, n denotes the number of equal-sized elements that mesh the unit interval domain. In these experiments, the viscosity-upwinded weighting strategy is chosen.

strategy of

$$\lambda = \frac{(\mu^-)^\alpha}{(\mu^-)^\alpha + (\mu^+)^\alpha} \tag{13}$$

where $\alpha > 1$ is a user-chosen parameter controlling the biasing strength. The limit $\alpha \to \infty$ corresponds to pure one-sided biasing, denoted in this work as *viscosity-upwinded weighting*,

$$\lambda = \begin{cases} 0 & \text{if } \mu^- < \mu^+, \\ 0.5 & \text{if } \mu^- = \mu^+, \\ 1 & \text{if } \mu^- > \mu^+. \end{cases} \tag{14}$$

In applications involving variable ellipticity coefficient, wherein the ratio may change between less-than-unity to greater-than-unity along an interface, it may be beneficial to smoothly vary λ from less than half to greater than half. If so, a finite value of α could be more appropriate, e.g., $\alpha = 2$. This possibility is not investigated here; instead, for the results presented in this work, the viscosity-upwinded weighting has been uniformly effective, and so the strategy in (14) is hereon adopted throughout, unless otherwise stated.

4.3. ***Multigrid performance under mesh refinement.*** In the last set of results for the one-dimensional test problem, we examine multigrid performance under the action of mesh refinement. Figure 5 shows the condition number of the V-cycled preconditioned system (using the viscosity-upwinded weighting strategy) for different values of the viscosity ratio, specifically 10^{-8}, 10^{-4}, 1, 10^4, and 10^8, as a function of the grid size, ranging from $n = 16$ to 4096 elements. Note that, in all cases, the condition number remains in the interval $\kappa \in (1.35, 1.60)$, which for practical purposes can essentially be considered a well-conditioned system, independent of h. Thus, as used in a multigrid preconditioned conjugate gradient method, for example, we expect a bounded number of iterations for a fixed reduction in residual norm, and this is indeed observed in experiments.

5. Two- and three-dimensional results

In this section, we assess the efficacy of viscosity-upwinded weighted fluxes for a variety of elliptic interface problems in two and three dimensions, ranging from two-phase constant-coefficient problems, to multiphase problems with variable viscosity coefficients, to a set of challenging "bubbly" geometry problems.

5.1. ***Rectangular interface.*** First, we consider 2D and 3D analogues of the two-phase, constant-coefficient problem given in the previous section. Let $\Omega = (0, 1)^d$ be divided into an interior phase $\Omega_1 = \left(\frac{1}{4}, \frac{3}{4}\right)^d$ and exterior phase $\Omega_2 = \Omega \setminus \overline{\Omega_1}$. The elliptic interface problem given in (10) is considered, with source data f, g, h, and g_∂ generated by the exact solution

$$u(\mathbf{x}) = \begin{cases} \prod_{i=1}^d \sin 4\pi(x_i - 0.1) & \text{in } \Omega_1, \\ \prod_{i=1}^d \cos 4\pi(x_i - 0.1) & \text{in } \Omega_2, \end{cases}$$

where $x_1 = x$, $x_2 = y$, and $x_3 = z$. A uniform Cartesian grid mesh with n elements in each direction is employed, with polynomial degree $p = 3$ in 2D (i.e., a piecewise-bicubic polynomial space), and $p = 2$ in 3D (i.e., piecewise-triquadratic); boundary faces carry a penalty parameter $\tau_D = \mu_2(p + 1)h^{-1}$, interfacial faces $\tau_{12} = \min(\mu_1, \mu_2)(p + 1)h^{-1}$, and all other faces zero penalty parameter.

Fixing $\mu_2 = 1$ and $\mu_1 = 10^{-5}$, Figure 6 illustrates the differences between a suboptimal choice of central interfacial flux along with a more optimal, viscosity-upwinded strategy. In the case of an interfacial central flux, the condition number of the V-cycle preconditioned system is approximately 508 and the discrete solution error shown in Figure 6, top center, exhibits significant numerical boundary layers. Inspection of the spectrum of the Gauss–Seidel relaxation operator S reveals that, although all eigenvalues lie in the unit interval, the eigenfunction with largest eigenvalue (approximately 0.991) is identical in profile to the function shown in Figure 6, top center, which is clearly not spatially smooth. Another example is shown in Figure 6, top right, which displays the eigenfunction of S having ninth-largest eigenvalue (approximately 0.965), which is also nonsmooth. As in Section 4.1, we see that the multigrid relaxation method exhibits slowly damped modes that are spatially high-frequency in profile, thereby preventing ideal multigrid behavior. However, with a viscosity-upwinded interfacial flux strategy, the preconditioned system has condition number approximately 1.55; Figure 6, bottom center, shows a significant improvement in the accuracy of the discrete solution; and the corresponding eigenfunction of S (now with eigenvalue 0.944) is vividly

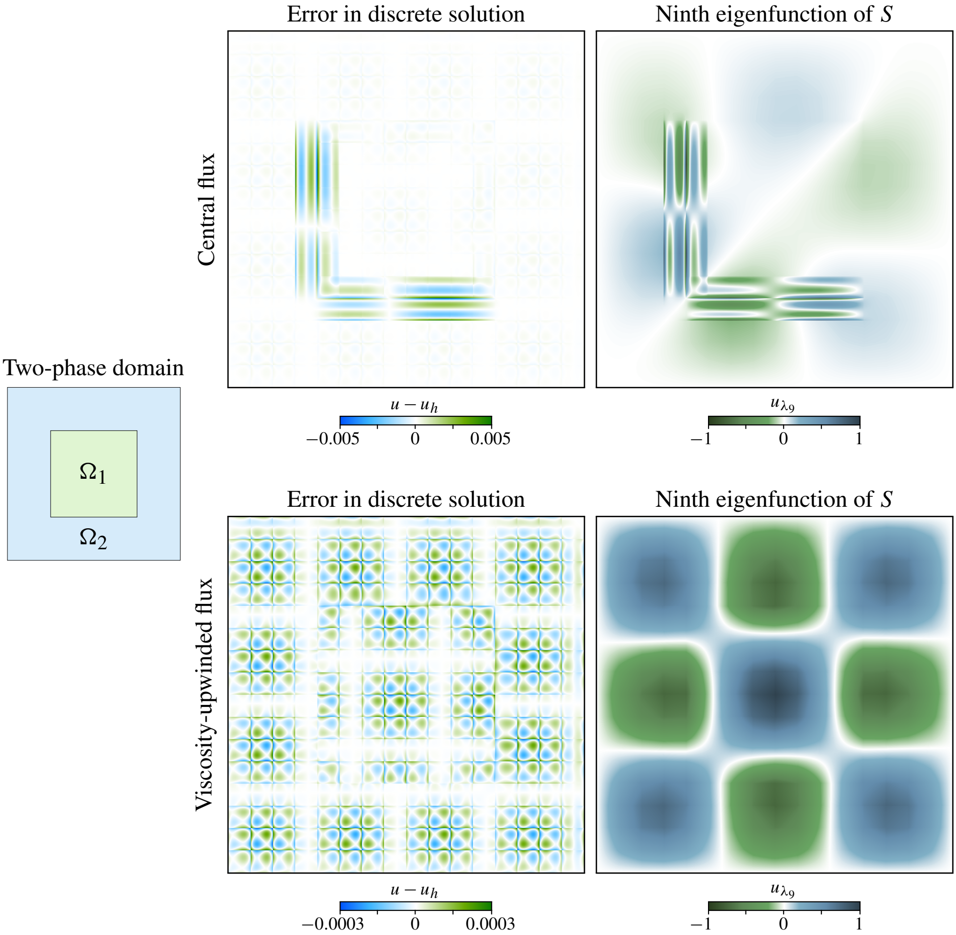


Figure 6. Solution accuracy and example eigenfunctions of the multigrid relaxation operator for a two-phase, constant-coefficient elliptic interface problem in two dimensions, wherein $\Omega_1 = \left(\frac{1}{4}, \frac{3}{4}\right)^2$ has viscosity coefficient 10^{-5} and the exterior phase $\Omega_2 = (0, 1)^2 \setminus \overline{\Omega_1}$ has coefficient 1. Top row: using a suboptimal central-flux weighting strategy. Bottom row: using the more optimal, viscosity-upwinded weighting strategy.

smoother. Similar conclusions hold for the two flux weighting strategies when the viscosity ratio is reversed.

To confirm that a viscosity-upwinded weighting strategy is ideal across a range of viscosity ratios, the experiments of Section 4.2 are repeated here for the 2D and 3D cases, corresponding to fixed 16 × 16 and 16 × 16 × 16 meshes, respectively. In particular, for five different values of $\mu_1 \in \{10^{-8}, 10^{-4}, 1, 10^4, 10^8\}$, the convex combination for the numerical flux of interfacial faces is varied from one-sided in one direction ($\lambda = 0$) to one-sided in the opposite direction ($\lambda = 1$). Figure 7 illustrates the behavior of the maximum norm error in the discrete solution and the

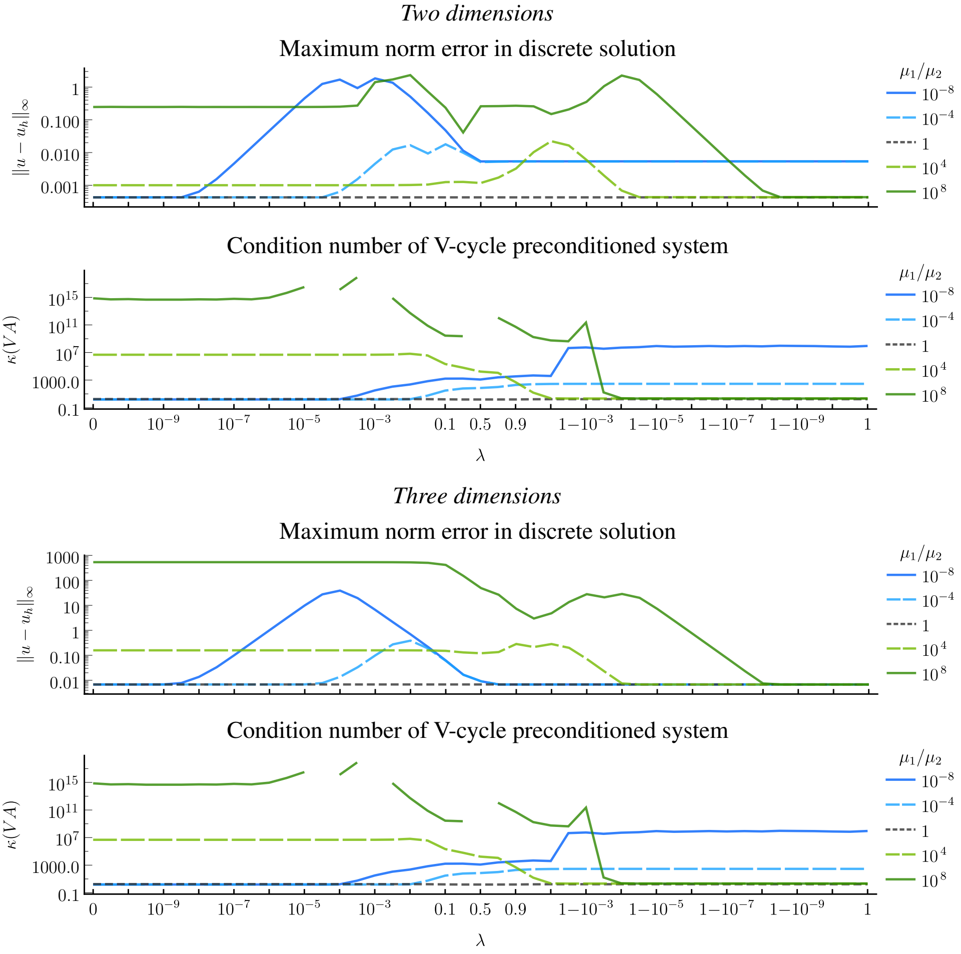


Figure 7. Solution accuracy and multigrid performance as a function of interfacial flux weighting for a two-phase, constant-coefficient elliptic interface problem in 2D and 3D, wherein $\Omega_1 = \left(\frac{1}{4}, \frac{3}{4}\right)^d$ has viscosity coefficient μ_1 and $\Omega_2 = (0, 1)^d \setminus \overline{\Omega_1}$ has coefficient μ_2, with the viscosity ratio as indicated. Here, the λ parameter is varied corresponding to a one-sided flux ($\lambda = 0$), to a central flux ($\lambda = 0.5$), to a one-sided flux in the opposite direction ($\lambda = 1$); note the quasilogarithmic scale of λ values on the horizontal axis.

condition number[9] of the V-cycle preconditioned system as a function of λ. Similar

[9] In higher dimensions, it can be computationally expensive to calculate an exact two-norm condition number of the preconditioned operator VA, especially for highly resolved meshes. In this paper, the condition number for 2D and 3D problems is approximated via eigenvalue estimation methods derived from the Lanczos iteration of the preconditioned conjugate gradient (PCG) algorithm [42]; in essence, these techniques compute the spectrum of the linear system's projection onto the underlying Krylov subspace. To apply these estimators, a randomly generated right-hand side vector is given to PCG which, with high probability, samples both large and small eigenmodes. Experiments indicate the condition number estimate is highly accurate (at least two digits) for

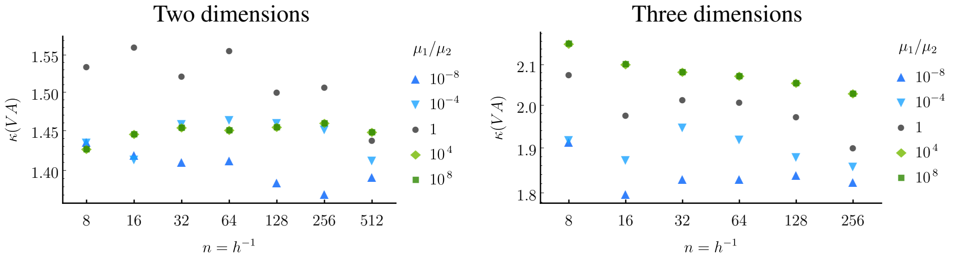


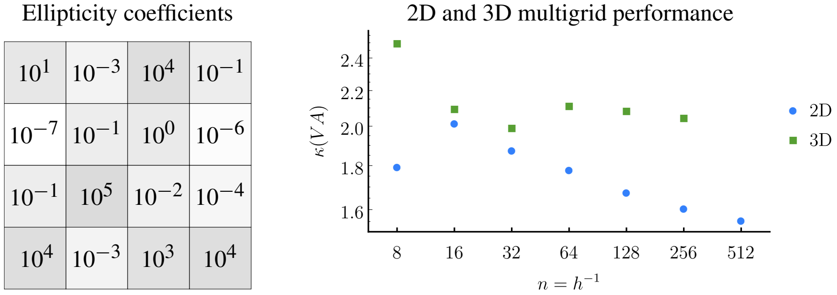
Figure 8. Multigrid performance under the action of mesh refinement for a two-phase, constant-coefficient elliptic interface problem in two and three dimensions, wherein $\Omega_1 = \left(\frac{1}{4}, \frac{3}{4}\right)^d$ has viscosity coefficient μ_1 and $\Omega_2 = (0, 1)^d \setminus \overline{\Omega_1}$ has coefficient μ_2, with the viscosity ratio as indicated. Here, n denotes the number of elements in the corresponding uniform Cartesian $n \times n(\times n)$ mesh.

to the conclusions found in the one-dimensional case, we see that the solution error is minimized when $\lambda \lessapprox 0.1(\mu_1/\mu_2)$ if $\mu_1/\mu_2 < 1$, or $\lambda \gtrapprox 1 - 0.1(\mu_2/\mu_1)$ if $\mu_1/\mu_2 > 1$; meanwhile, the condition number is minimized when $\lambda \lessapprox \sqrt{\mu_1/\mu_2}$ if $\mu_1/\mu_2 < 1$, or $\lambda \gtrapprox 1 - \sqrt{\mu_2/\mu_1}$ if $\mu_1/\mu_2 > 1$.

In the remainder of this article, we cease examination of the influence of λ on accuracy and conditioning. Instead, the viscosity-upwinded interfacial flux strategy is automatically applied, and attention is focused solely on multigrid performance under the action of mesh refinement. Figure 8 shows the condition number of the V-cycled preconditioned system for the elliptic interface problem with the rectangular interface geometry currently under consideration. In two dimensions, the mesh is refined from 8×8 to 512×512, while in three dimensions, the mesh is refined from $8 \times 8 \times 8$ to $256 \times 256 \times 256$ (representing a maximum of almost half a billion degrees of freedom in the solution u). In 2D, the condition number remains in the interval (1.35, 1.60), while in 3D it remains in the interval (1.8, 2.2), independent of h, for all viscosity ratios.

5.2. ***Multiphase checkerboard.*** In the next example, we consider a multiphase elliptic interface problem exhibiting a checkerboard pattern of different viscosity coefficients, as shown in Figure 9, left. The largest jump in viscosity ratio across any one interface is 10^8, and the largest ratio across all phases is 10^{12}. Boundary faces in phase i carry the penalty parameter $\tau_D = \mu_i(p+1)h^{-1}$ and interphase faces on Γ_{ij} carry a penalty parameter $\tau_{ij} = 2\min(\mu_i, \mu_j)(p+1)h^{-1}$. Figure 9, right, shows the condition number of the multigrid preconditioned system in 2D and

reasonably conditioned systems, and becomes inaccurate only for badly conditioned systems with $\kappa \gg 10^4$. However in these ill-conditioned cases the precise value of κ is not of concern. Regarding the results in this work, if PCG fails to converge within 1000 iterations, the last estimate of the condition number is taken; if the system is so severely ill-conditioned that PCG reports the matrix is not symmetric positive definite, the condition estimate is set to ∞ and does not appear in the graphs.



Ellipticity coefficients

10^1	10^{-3}	10^4	10^{-1}
10^{-7}	10^{-1}	10^0	10^{-6}
10^{-1}	10^5	10^{-2}	10^{-4}
10^4	10^{-3}	10^3	10^4

2D and 3D multigrid performance

Figure 9. Left: a multiphase domain divided into a 4×4 array of subdomains with viscosity coefficient μ_i as indicated. Right: multigrid performance under the action of mesh refinement for the corresponding multiphase, constant-coefficient elliptic interface problem in two dimensions and a 3D analogue. Here, n denotes the number of elements in the uniform Cartesian $n \times n(\times n)$ mesh.

an analogous 3D case of the checkerboard problem. Bounded condition numbers as $h \to 0$ are observed in all cases.

5.3. ***Variable ellipticity coefficient.*** In the results presented so far, only constant-coefficient elliptic interface problems have been investigated. In the following problem, we consider a variable-coefficient problem wherein the jump in μ across the interface varies in space over several orders of magnitude. Specifically, in two dimensions, the domain is the unit square divided into an interior channel $\Omega_2 = (0, 1) \times \left(\frac{1}{4}, \frac{3}{4}\right)$ and an exterior phase $\Omega_1 = (0, 1)^2 \setminus \overline{\Omega_2}$, such that $\mu = \mu(x, y)$ is given by

$$\mu = \begin{cases} 10^{-4+8\sin\pi x/2} & \text{in } \Omega_1, \\ 10^{4-8\sin\pi x/2} & \text{in } \Omega_2. \end{cases} \tag{15}$$

Figure 10, left, illustrates the domain and μ on a base-10 logarithmic scale; note that the maximum viscosity jump is eight orders in magnitude. In 3D, an analogous configuration is chosen consisting of the unit cube divided into an interior channel $\Omega_2 = (0, 1) \times (0, 1) \times \left(\frac{1}{4}, \frac{3}{4}\right)$ and exterior phase $\Omega_1 = (0, 1)^3 \setminus \overline{\Omega_2}$, with $\mu = \mu(x, y, z)$ given by

$$\mu = \begin{cases} 10^{-4+8\sin\pi x/2\sin\pi y/2} & \text{in } \Omega_1, \\ 10^{4-8\sin\pi x/2\sin\pi y/2} & \text{in } \Omega_2. \end{cases}$$

In these examples, the viscosity-upwinded weighting strategy switches between the two phases Ω_1 and Ω_2 depending on the location on the interface. In addition, interphase faces carry a penalty parameter $\tau_{12} = 2\min(\mu_1, \mu_2)(p+1)h^{-1}$, which also varies from face to face. Figure 10, right, shows the condition number of the multigrid-preconditioned system in 2D and 3D. We see that optimal behavior is obtained, i.e., κ remains bounded as $h \to 0$.

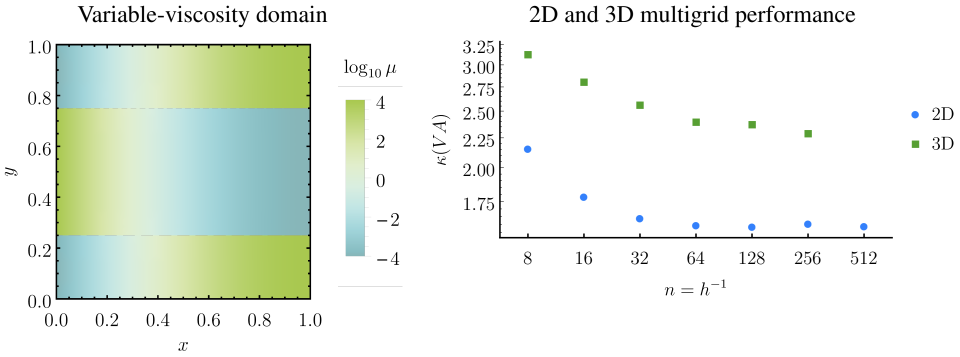


Figure 10. Left: a two-phase domain with variable ellipticity coefficient given by (15); the interface separating the two phases is $\Gamma = \{(x, y) : y = \frac{1}{4} \text{ or } y = \frac{3}{4}\}$. Right: multigrid performance under the action of mesh refinement for the corresponding two-phase, variable-coefficient elliptic interface problem and its associated 3D analogue. Here, n denotes the number of elements in the uniform Cartesian $n \times n(\times n)$ mesh.

5.4. ***Spherical geometry.*** In the remaining set of examples, we consider curved interface geometry and make use of a recently developed discontinuous Galerkin framework for computing high-order accurate multiphase multiphysics using implicitly defined meshes [44; 45]. Briefly, an implicitly defined mesh uses one or more level set functions, describing the domain geometry and interface, to cut through the cells of a background quadtree or octree; tiny cut cells are then merged with neighboring cells to create a mesh in which the shapes of interfacial or boundary elements are defined implicitly by the level set functions. In particular, the mesh is interface-conforming and sharply represents its implicitly defined geometry. For the elements and faces of the mesh whose geometry is implicitly defined, high-order accurate quadrature rules are computed using the schemes detailed in [43; 46]; these quadrature schemes are then used in the LDG methods for computing mass matrices, discrete gradient operators, L^2 projections, and so forth. For details on the implicit mesh DG framework, see [44; 45]; for illustrations of the associated multiphase interface-preserving h-multigrid hierarchy, see [29].

In the first example of curved geometry, we consider a circle and sphere of radius 0.3, i.e., $\Omega_1 = \{x : \|x\| < 0.3\}$ and $\Omega_2 = (0, 1)^d \setminus \overline{\Omega_1}$. An example of the implicitly defined mesh for a background Cartesian 16×16 grid is shown in Figure 11, left; note that away from the interface, the mesh consists of standard rectangular elements, whereas near the interface, the cell merging procedure results in a nonconforming mesh, with some mesh faces shared between more than two neighboring elements. Because of the increased complexity of the mesh topology, as compared to a standard Cartesian grid, with a larger variety of mesh face sizes, multigrid performance using implicitly defined meshes benefits from a slightly increased penalty stabilization in the LDG schemes. As such, in the remainder of these examples, penalty parameters

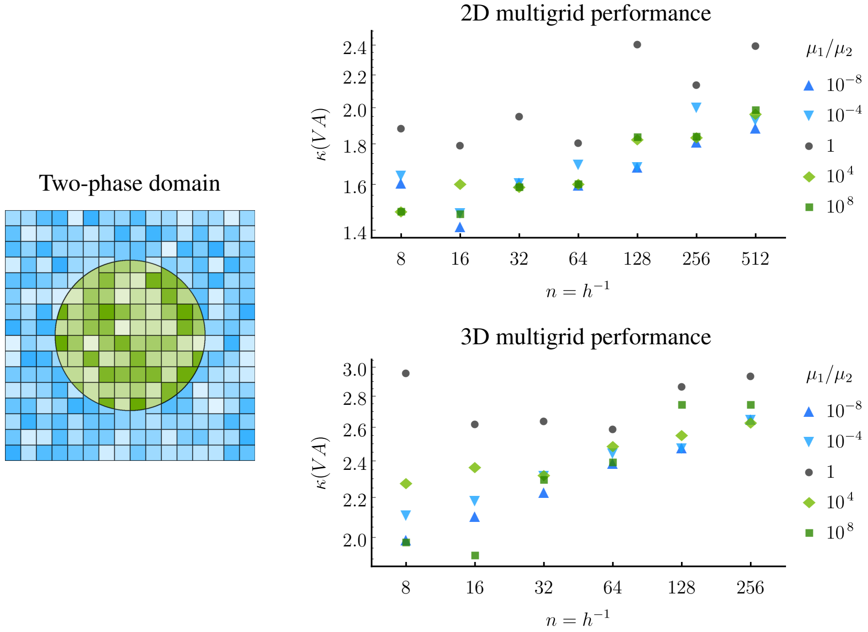


Figure 11. Left: an implicitly defined mesh corresponding to the two-phase circular interface test problem of Section 5.4 generated with a background 16 × 16 grid. Right: multigrid performance under the action of mesh refinement for the associated two-phase, constant-coefficient elliptic interface problem. Here, n denotes the number of cells in the uniform Cartesian $n \times n(\times n)$ grid underlying the employed implicitly defined meshes.

for interfacial faces are set to $\tau_{12} = 2^d \min(\mu_1, \mu_2)(p+1)h^{-1}$ in d dimensions and penalty parameters for boundary faces remain equal to $\mu^-(p+1)h^{-1}$ (where μ^{-1} is the coefficient of the element attached to the boundary face), while all remaining faces carry a penalty parameter of $\tau = \mu_F(p+1)h^{-1}$ (where μ_F represents the coefficient value local to the face in question), where h is the cell size of the background Cartesian grid.[10] These penalty parameters were chosen experimentally so as to approximately optimize for both solution accuracy and multigrid performance.

Using the spherical interface geometry, Figure 11 displays results for a variety of coefficient ratios (in this example, each phase has constant ellipticity coefficient). As compared to the simpler meshes used in previous test problems, the condition number of the V-cycle preconditioned system is slightly larger. This is attributed to both the increased mesh complexity, as well as the influence of the chosen nodal polynomial basis on curved, implicitly defined elements (see [44] for details). We also observe a minor trend upwards in condition number as the mesh is refined from

[10] Nonuniform quadtree and octrees can also be used with implicitly defined meshes, and corresponding multigrid algorithms have been devised [44; 29]; however, adaptive mesh refinement is not considered in the present work.

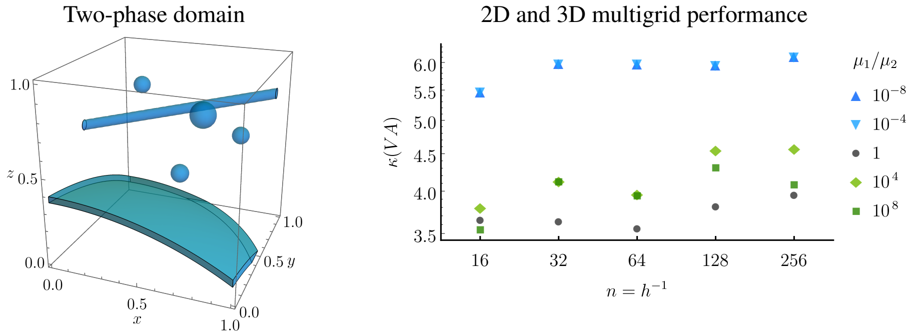
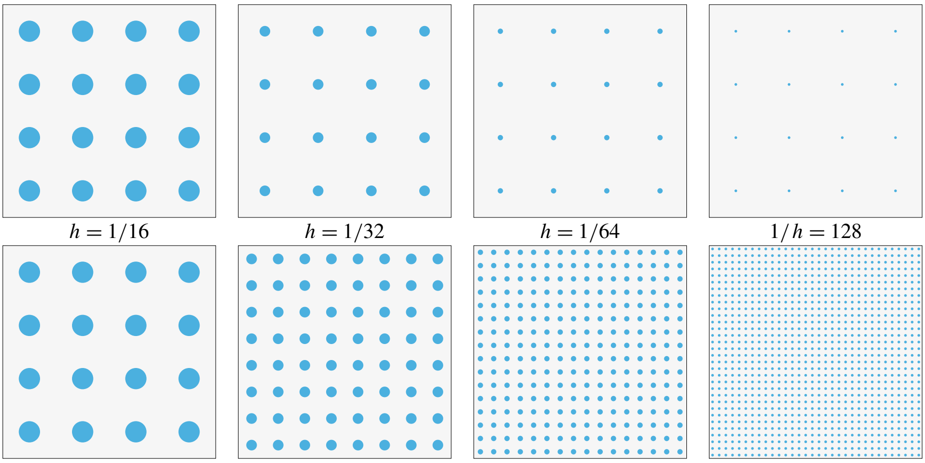


Figure 12. Left: a two-phase domain divided into the indicated shaded phase Ω_1, consisting of a thin sheet, a long thin filament, and three droplets; the secondary phase Ω_2 fills the exterior region. Right: multigrid performance under the action of mesh refinement for the geometry depicted in the left. Here, n denotes the number of cells in the uniform Cartesian $n \times n \times n$ grid underlying the employed implicitly defined meshes.

the coarsest level; this trend starts to plateau for the finest meshes and is expected to plateau for ultrarefined meshes. For the presented results, the condition numbers of the preconditioned system remain in the interval (1.4, 2.4) in 2D and (1.8, 3.0) in 3D, across 16 orders of ellipticity coefficient ratio.

5.5. ***Thin sheets, filaments, and droplets.*** To examine multigrid performance in the case of more challenging interface geometry, in the next test problem we consider a 3D example exhibiting a thin sheet, a thin filament, and three small, dispersed phase components, as illustrated in Figure 12, left. Here, the shaded phase (Ω_1) is composed using multiple level set functions describing a spherical shell, a cylinder, and three spheres, and Ω_2 denotes the exterior phase. Figure 12, right, shows the condition number of the multigrid preconditioned system for a variety of viscosity ratios. For this test geometry, the condition number is in some cases about two times bigger than witnessed in previous three-dimensional problems; the increased conditioning is attributed to the more challenging geometry; however, bounded condition numbers are still attained as $h \to 0$.

5.6. ***Bubbly geometry.*** In the last set of examples, we consider two kinds of problems involving "bubbly" interface geometry. These are representative of the kind of challenging multimaterial problems in which small interfacial features are in some sense never resolved by the mesh, e.g., dispersed gas bubbles in a liquid with diameter only a few mesh elements. The first problem considers a lattice of 4×4 droplets (in 2D) or $4 \times 4 \times 4$ droplets (in 3D), each of radius $0.8h$, where h is the cell size of the background Cartesian grid. The second problem considers a lattice of $k \times k(\times k)$ droplets of the same size, where $k = 1/(4h) = n/4$; in particular, the number of droplets increases as the mesh is refined. Figure 13, top and bottom,



$h = 1/16$ $h = 1/32$ $h = 1/64$ $1/h = 128$

Figure 13. Illustration of the “bubbly” geometry of the elliptic interface problems considered in Section 5.6 where $h = 1/n$ is the cell size of the background uniform Cartesian grid used to define the corresponding implicitly defined meshes. Top: fixed lattice of vanishing droplets. Bottom: growing lattice of vanishing droplets.

illustrates the two-dimensional case for these two problems. To emphasize, the droplet curvature length scale is tied to the element size of the mesh in such a way that the resulting elliptic interface problem contains ever-decreasing small geometric features — as such, we do not necessarily expect a multigrid method to attain optimal efficiency, i.e., bounded $\kappa(VA)$ as $h \to 0$, as the relaxation operator may not exhibit the usual scale-separated smoothing behavior across the full grid hierarchy.

Figure 14 shows the multigrid preconditioned condition numbers $\kappa(VA)$ for both bubbly problems in 2D and 3D. A number of observations can be made:

- For viscosity ratio less than one, i.e., when the droplets are less viscous than the surrounding medium, well-behaved multigrid performance is obtained in all tested cases, with condition numbers bounded as $h \to 0$ and close to unity.
- For unit viscosity ratio, we see a small upwards trend in the condition number in all cases.
- For viscosity ratio greater than one, a stronger upwards trend in condition number is seen. In particular, the problem with an ever-increasing lattice of vanishingly small droplets exhibits larger condition numbers than the case of a fixed-size lattice; the former case generally has better conditioning in 3D than in 2D.
- In 2D, on the finest and second-finest meshes, some of the test cases with viscosity ratio 10^4 or 10^8 failed to converge by 1000 PCG iterations. Experiments indicate the condition number of A for these cases exceeds 10^{15}; it is

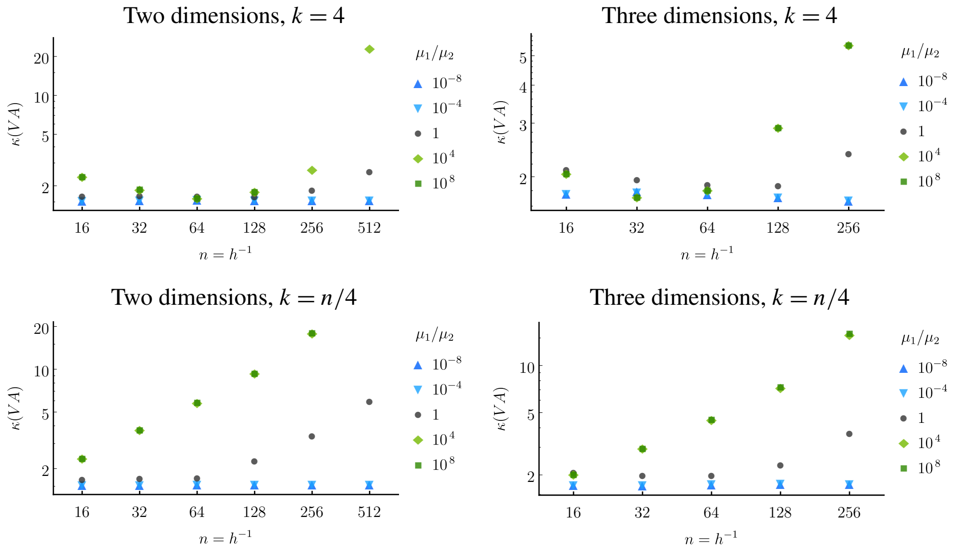


Figure 14. Multigrid performance under the action of mesh refinement for the two-phase elliptic interface problems with "bubbly" geometry shown in Figure 13, consisting of a $k \times k(\times k)$ lattice of droplets of vanishing size. In the top row, the lattice size is fixed at $k = 4$; in the bottom row, the number of droplets increases as the mesh is refined, with $k \propto n = 1/h$. The droplets have viscosity coefficient μ_1 whereas the surrounding medium has viscosity μ_2. Here, n denotes the number of cells in the uniform Cartesian $n \times n(\times n)$ grid underlying the employed implicitly defined meshes.

not surprising therefore that multigrid fails to precondition stably when $\kappa(A)$ exceeds limits of double-precision arithmetic (as used in this work).

A possible explanation for the above observed behavior is as follows. When the viscosity ratio is less one, the droplets are less viscous than the surrounding medium — one may think of gas bubbles in water. As intuited in the motivation of Section 1.1, this case reduces to a Dirichlet problem for the gas bubbles and a Neumann problem for the surrounding medium; as seen in the results, the condition number remains bounded as $h \to 0$. In effect, the liquid medium solves a Neumann problem for the bulk domain, and transmits Dirichlet boundary conditions to individual droplets. On the other hand, when the viscosity ratio is much greater than one, the droplets are more viscous than the surrounding medium — one may think of liquid droplets surrounded by gas. In this circumstance, the individual liquid droplets (nearly) solve a Poisson problem with a (nearly) pure Neumann boundary condition, whose solution is therefore (almost) defined up to an arbitrary constant. Thus, each droplet solves a Poisson problem that is nearly decoupled from all others. In actuality, each droplet's constant is uniquely defined by the solution across the entire domain. Thus, in effect, the liquid droplets are very weakly coupled to each other via the surrounding gas phase medium. In an incompressible fluid flow problem, this could be interpreted through the physical intuition that each droplet's viscous stress is essentially unaffected by distant droplets, owing to the fact the gas phase has weak viscous forces.

A different perspective comes from the multigrid mesh hierarchy. Owing to the property that elements are never agglomerated across interfaces, as the hierarchy coarsens, larger and larger elements for the surrounding medium Ω_2 are created, having relatively smaller and smaller punctured discs. If the viscosity ratio is much less than one, the surrounding medium (nearly) solves a Neumann Poisson problem that is largely decoupled from the Poisson problem on each droplet; according to the presented results, in this circumstance, multigrid performance through the coarsening mesh hierarchy is unaffected by tiny punctures in the mesh. If the viscosity ratio is much greater than one, the surrounding medium (nearly) solves a Dirichlet Poisson problem, whose Dirichlet boundary conditions are determined by the (nearly) Neumann problem on each individual puncture. On coarse grids, neighboring droplets are agglomerated into a single element having many connected components (see, e.g., Figure 4.8 in [29]). Here it is apparent the multigrid method does not get a chance to effectively solve the Poisson problem on each individual droplet; the geometry is simply too complex for a fixed-degree piecewise polynomial solution to accurately solve.

As seen, elliptic interface problems with vanishingly small geometry can pose difficulties for efficient multigrid performance, depending on the configuration of ellipticity coefficients. A variety of techniques could be used to tackle these problems — one possibility may be to prevent neighboring droplets from being agglomerated together (which comes at the cost of increased degrees of freedom), or to design solvers that identify specific geometric components and exclude them from normal treatment, e.g., by using a deflated conjugate gradient algorithm; see, e.g., [40; 49]. These possibilities in combination with viscosity-upwinded LDG operator-coarsening multigrid schemes could be pursued in future work.

6. Concluding remarks

In this paper, we discussed the design of local discontinuous Galerkin methods for multiphase elliptic interface problems, with a central focus on obtaining good multigrid solver performance through an apt choice of weighting in the numerical fluxes for interfacial mesh faces. In particular, across interfaces exhibiting jumps in viscosity of several orders in magnitude, a simple physical argument showed that the more viscous phase sees a predominantly Neumann-like boundary condition on the interface, whereas the less viscous phase sees a predominantly Dirichlet-like boundary condition. As such, one may expect better discretization characteristics or multigrid relaxation/smoothing behavior if the numerical fluxes are biased appropriately. This was indeed observed here — findings support the commonly used strategy of harmonic weighting, but also show that results can be improved further by using a viscosity-upwinded strategy, wherein the numerical fluxes for the unknown u and its flux $\boldsymbol{q} = \mu \nabla u$ are biased entirely so as to obtain one-sided fluxes.

The test problems presented in this study examined simple constant-coefficient elliptic interface problems as well as problems with variable viscosity, multiphase checkerboarding, and intricate curved geometry. In particular, viscosity coefficient ratios ranged across 16 orders in magnitude. The primary metric used to test multigrid efficacy consisted of the two-norm condition number of the associated multigrid V-cycle preconditioned system, $\kappa(VA)$; with the exception of a challenging elliptic interface problem involving bubbly geometry, the results showed that, using viscosity-upwinded numerical fluxes, $\kappa(VA)$ is unit order in magnitude and bounded as $h \to 0$. The exception to the result concerns the very challenging case of a lattice of vanishingly small droplets; see the discussion in Section 5.6. We note that this metric examining $\kappa(VA)$ is relatively stringent — for example, in establishing convergence results for the conjugate gradient method, κ leads to an upper bound on the number of iterations needed to reduce the residual by a given factor; the number of iterations which (preconditioned) conjugate gradient may actually take could be fewer and depends on the clustering of the spectrum of VA. Our results predominantly examined the case of $p = 3$ in two dimensions and $p = 2$ in three dimensions. Experiments show that optimal multigrid behavior as $h \to 0$ is seen with other polynomial degrees as well, with tested values ranging from $p = 1$ up to $p = 9$. In all cases, the derived LDG schemes for elliptic interface problems are optimal order accurate, showing $p + 1$ convergence rates in the maximum norm.

A variety of aspects could be studied in future work. Extension of viscosity-upwinded LDG schemes to matrix-valued diffusion coefficients is one possibility; here, the work of Ern et al. [28] suggests that in this case, one could upwind based on the normal component of the viscosity tensor, e.g., apply (12), (13), or (14) to $\boldsymbol{n} \cdot \mu^{\pm} \cdot \boldsymbol{n}$, where $\boldsymbol{n}$ is the normal to the interface. Meanwhile, although the focus was not on minimizing $\kappa(VA)$ as best as possible, a few remarks can be made in this regard. In this work a one-sided intraphase flux is used, which leads to a more compact stencil for the final discrete Laplacian operator; according to some tests, a central intraphase flux can lead to 10–20% better condition numbers, but at the cost of increased stencil size. In addition, for the presented three-dimensional results, a domain decomposition MPI implementation with a processor-block damped Gauss–Seidel relaxation method was used; owing to the damping used, the resulting condition number is about 25% larger than what could be obtained if no damping was used. One could also investigate different damping strategies or relaxation methods; for example, polynomial relaxation algorithms or additive Schwarz smoothers, which have been shown effective for other DG schemes involving agglomeration procedures [8].

Acknowledgements

This research was supported by the Applied Mathematics Program of the U.S. Department of Energy (DOE) Office of Advanced Scientific Computing Research under contract number DE-AC02-05CH11231, and by the High Performance Computing for Manufacturing (HPC4Mfg) program sponsored by the U.S. DOE Advanced Manufacturing Office of the Energy Efficiency and Renewable Energy Office. Some computations used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. DOE Office of Science User Facility operated under contract number DE-AC02-05CH11231.

References

[1] M. Adams, M. Brezina, J. Hu, and R. Tuminaro, *Parallel multigrid smoothing: polynomial versus Gauss–Seidel*, J. Comput. Phys. **188** (2003), no. 2, 593–610. MR Zbl

[2] B. Aksoylu, I. G. Graham, H. Klie, and R. Scheichl, *Towards a rigorously justified algebraic preconditioner for high-contrast diffusion problems*, Comput. Vis. Sci. **11** (2008), no. 4–6, 319–331. MR

[3] B. Aksoylu and Z. Yeter, *Robust multigrid preconditioners for cell-centered finite volume discretization of the high-contrast diffusion equation*, Comput. Vis. Sci. **13** (2010), no. 5, 229–245. MR Zbl

[4] R. E. Alcouffe, A. Brandt, J. E. Dendy, Jr., and J. W. Painter, *The multigrid method for the diffusion equation with strongly discontinuous coefficients*, SIAM J. Sci. Statist. Comput. **2** (1981), no. 4, 430–454. MR Zbl

[5] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, *A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations*, J. Comput. Phys. **142** (1998), no. 1, 1–46. MR Zbl

[6] C. Annavarapu, M. Hautefeuille, and J. E. Dolbow, *A robust Nitsche's formulation for interface problems*, Comput. Methods Appl. Mech. Engrg. **225–228** (2012), 44–54. MR Zbl

[7] P. F. Antonietti, P. Houston, G. Pennesi, and E. Süli, *An agglomeration-based massively parallel non-overlapping additive Schwarz preconditioner for high-order discontinuous Galerkin methods on polytopic grids*, preprint, 2019. arXiv

[8] P. F. Antonietti and G. Pennesi, *V-cycle multigrid algorithms for discontinuous Galerkin methods on non-nested polytopic meshes*, J. Sci. Comput. **78** (2019), no. 1, 625–652. MR Zbl

[9] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM J. Numer. Anal. **39** (2002), no. 5, 1749–1779. MR Zbl

[10] N. Barrau, R. Becker, E. Dubach, and R. Luce, *A robust variant of NXFEM for the interface problem*, C. R. Math. Acad. Sci. Paris **350** (2012), no. 15–16, 789–792. MR Zbl

[11] P. Bastian, M. Blatt, and R. Scheichl, *Algebraic multigrid for discontinuous Galerkin discretizations of heterogeneous elliptic problems*, Numer. Linear Algebra Appl. **19** (2012), no. 2, 367–388. MR Zbl

[12] P. Bastian, E. H. Müller, S. Müthing, and M. Piatkowski, *Matrix-free multigrid block-preconditioners for higher order discontinuous Galerkin discretisations*, J. Comput. Phys. **394** (2019), 417–439. MR

[13] J. B. Bell, C. N. Dawson, and G. R. Shubin, *An unsplit, higher order godunov method for scalar conservation laws in multiple dimensions*, J. Comput. Phys. **74** (1988), no. 1, 1–24. Zbl

[14] M. Blatt, *A parallel algebraic multigrid method for elliptic problems with highly discontinuous coefficients*, Ph.D. thesis, Universität Heidelberg, 2010. Zbl

[15] E. Burman and A. Ern, *An unfitted hybrid high-order method for elliptic interface problems*, SIAM J. Numer. Anal. **56** (2018), no. 3, 1525–1546. MR Zbl

[16] E. Burman, J. Guzmán, M. A. Sánchez, and M. Sarkis, *Robust flux error estimation of an unfitted Nitsche method for high-contrast interface problems*, IMA J. Numer. Anal. **38** (2018), no. 2, 646–668. MR Zbl

[17] E. Burman and P. Zunino, *A domain decomposition method based on weighted interior penalties for advection-diffusion-reaction problems*, SIAM J. Numer. Anal. **44** (2006), no. 4, 1612–1638. MR Zbl

[18] ______, *Numerical approximation of large contrast problems with the unfitted Nitsche method*, Frontiers in numerical analysis – Durham 2010 (J. Blowey and M. Jensen, eds.), Lect. Notes Comput. Sci. Eng., no. 85, Springer, 2012, pp. 227–282. MR Zbl

[19] Z. Cai, X. Ye, and S. Zhang, *Discontinuous Galerkin finite element methods for interface problems: a priori and a posteriori error estimations*, SIAM J. Numer. Anal. **49** (2011), no. 5, 1761–1787. MR Zbl

[20] T. Chen and J. Strain, *Piecewise-polynomial discretization and Krylov-accelerated multigrid for elliptic interface problems*, J. Comput. Phys. **227** (2008), no. 16, 7503–7542. MR Zbl

[21] B. Cockburn and B. Dong, *An analysis of the minimal dissipation local discontinuous Galerkin method for convection-diffusion problems*, J. Sci. Comput. **32** (2007), no. 2, 233–262. MR Zbl

[22] B. Cockburn and C.-W. Shu, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal. **35** (1998), no. 6, 2440–2463. MR Zbl

[23] R. K. Crockett, P. Colella, and D. T. 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), no. 7, 2451–2469. MR Zbl

[24] M. Dryja, *On discontinuous Galerkin methods for elliptic problems with discontinuous coefficients*, Comput. Methods Appl. Math. **3** (2003), no. 1, 76–85. MR Zbl

[25] M. Dryja and P. Krzyżanowski, *Additive Schwarz methods for DG discretization of elliptic problems with discontinuous coefficient*, Domain decomposition methods in science and engineering XXII (T. Dickopf, M. J. Gander, L. Halpern, R. Krause, and L. F. Pavarino, eds.), Lect. Notes Comput. Sci. Eng., no. 104, Springer, 2016, pp. 167–175. MR Zbl

[26] M. Dryja, M. V. Sarkis, and O. B. Widlund, *Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions*, Numer. Math. **72** (1996), no. 3, 313–348. MR Zbl

[27] A. Ern and J.-L. Guermond, *Quasi-optimal nonconforming approximation of elliptic PDEs with contrasted coefficients and minimal regularity*, preprint, 2019. arXiv

[28] A. Ern, A. F. Stephansen, and P. Zunino, *A discontinuous Galerkin method with weighted averages for advection-diffusion equations with locally small and anisotropic diffusivity*, IMA J. Numer. Anal. **29** (2009), no. 2, 235–256. MR Zbl

[29] D. Fortunato, C. H. Rycroft, and R. I. Saye, *Efficient operator-coarsening multigrid schemes for local discontinuous Galerkin methods*, SIAM J. Sci. Comput. **41** (2019), no. 6.

[30] T. Frachon and S. Zahedi, *A cut finite element method for incompressible two-phase Navier–Stokes flows*, J. Comput. Phys. **384** (2019), 77–98. MR

[31] J. Galvis and Y. Efendiev, *Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces*, Multiscale Model. Simul. **8** (2010), no. 5, 1621–1644. MR Zbl

[32] C. Gürkan and A. Massing, *A stabilized cut discontinuous Galerkin framework for elliptic boundary value and interface problems*, Comput. Methods Appl. Mech. Engrg. **348** (2019), 466–499. MR

[33] P. Hansbo, M. G. Larson, and S. Zahedi, *A cut finite element method for a Stokes interface problem*, Appl. Numer. Math. **85** (2014), 90–114. MR Zbl

[34] J. S. Hesthaven and T. Warburton, *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, Texts in Applied Mathematics, no. 54, Springer, 2008. MR Zbl

[35] P. Huang, H. Wu, and Y. Xiao, *An unfitted interface penalty finite element method for elliptic interface problems*, Comput. Methods Appl. Mech. Engrg. **323** (2017), 439–460. MR

[36] I. Klapper and T. Shaw, *A large jump asymptotic framework for solving elliptic and parabolic equations with interfaces and strong coefficient discontinuities*, Appl. Numer. Math. **57** (2007), no. 5–7, 657–671. MR Zbl

[37] M. Kumar and P. Joshi, *Some numerical techniques for solving elliptic interface problems*, Numer. Methods Partial Differential Equations **28** (2012), no. 1, 94–114. MR Zbl

[38] R. M. R. Lewis, *A guide to graph colouring: algorithms and applications*, Springer, 2016. MR Zbl

[39] T. Ludescher, S. Gross, and A. Reusken, *A multigrid method for unfitted finite element discretizations of elliptic interface problems*, preprint, 2018. arXiv

[40] S. P. MacLachlan, J. M. Tang, and C. Vuik, *Fast and robust solvers for pressure-correction in bubbly flow problems*, J. Comput. Phys. **227** (2008), no. 23, 9742–9761. MR Zbl

[41] J. Mandel and M. Brezina, *Balancing domain decomposition for problems with large jumps in coefficients*, Math. Comp. **65** (1996), no. 216, 1387–1401. MR Zbl

[42] Y. Saad, *Iterative methods for sparse linear systems*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, 2003. MR Zbl

[43] R. I. Saye, *High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles*, SIAM J. Sci. Comput. **37** (2015), no. 2, A993–A1019. MR Zbl

[44] ———, *Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid-structure interaction, and free surface flow, I*, J. Comput. Phys. **344** (2017), 647–682. MR Zbl

[45] ———, *Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid-structure interaction, and free surface flow, II*, J. Comput. Phys. **344** (2017), 683–723. MR Zbl

[46] ———, *Algoim: algorithms for implicitly defined geometry, level set methods, and Voronoi implicit interface methods*, 2019, software package.

[47] B. Schott, *Stabilized cut finite element methods for complex interface coupled flow problems*, Ph.D. thesis, Technische Universität München, 2017.

[48] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, *An adaptive level set approach for incompressible two-phase flows*, J. Comput. Phys. **148** (1999), no. 1, 81–124. MR Zbl

[49] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga, *Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods*, J. Sci. Comput. **39** (2009), no. 3, 340–370. MR Zbl

[50] E. Wadbro, S. Zahedi, G. Kreiss, and M. Berggren, *A uniformly well-conditioned, unfitted Nitsche method for interface problems*, BIT **53** (2013), no. 3, 791–820. MR Zbl

[51] W. L. Wan, *Interface preserving coarsening multigrid for elliptic problems with highly discontinuous coefficients*, Numer. Linear Algebra Appl. **7** (2000), no. 7–8, 727–741. MR Zbl

[52] H. Xiaoxiao, S. Fei, and D. Weibing, *Stabilized nonconforming nitsche's extended finite element method for stokes interface problems*, preprint, 2019. arXiv

[53] J. Xu and Y. Zhu, *Uniform convergent multigrid methods for elliptic problems with strongly discontinuous coefficients*, Math. Models Methods Appl. Sci. **18** (2008), no. 1, 77–105. MR Zbl

[54] Y. Zhu, *Domain decomposition preconditioners for elliptic equations with jump coefficients*, Numer. Linear Algebra Appl. **15** (2008), no. 2–3, 271–289. MR Zbl

[55] Z. Zou, W. Aquino, and I. Harari, *Nitsche's method for Helmholtz problems with embedded interfaces*, Internat. J. Numer. Methods Engrg. **110** (2017), no. 7, 618–636. MR Zbl

[56] P. Zunino, *Discontinuous Galerkin methods based on weighted interior penalties for second order PDEs with non-smooth coefficients*, J. Sci. Comput. **38** (2009), no. 1, 99–126. MR Zbl

Received July 11, 2019.

ROBERT I. SAYE: rsaye@lbl.gov
Mathematics Group, Lawrence Berkeley National Laboratory, Berkeley, CA, United States

Guidelines for Authors

Authors may submit manuscripts in PDF format on-line at the Submission page at msp.org/camcos.

Originality. Submission of a manuscript acknowledges that the manuscript is original and and is not, in whole or in part, published or under consideration for publication elsewhere. It is understood also that the manuscript will not be submitted elsewhere while under consideration for publication in this journal.

Language. Articles in CAMCoS are usually in English, but articles written in other languages are welcome.

Required items. A brief abstract of about 150 words or less must be included. It should be self-contained and not make any reference to the bibliography. If the article is not in English, two versions of the abstract must be included, one in the language of the article and one in English. Also required are keywords and subject classifications for the article, and, for each author, postal address, affiliation (if appropriate), and email address.

Format. Authors are encouraged to use LaTeX but submissions in other varieties of TeX, and exceptionally in other formats, are acceptable. Initial uploads should be in PDF format; after the refereeing process we will ask you to submit all source material.

References. Bibliographical references should be complete, including article titles and page ranges. All references in the bibliography should be cited in the text. The use of BibTeX is preferred but not required. Tags will be converted to the house format, however, for submission you may use the format of your choice. Links will be provided to all literature with known web locations and authors are encouraged to provide their own links in addition to those supplied in the editorial process.

Figures. Figures must be of publication quality. After acceptance, you will need to submit the original source files in vector graphics format for all diagrams in your manuscript: vector EPS or vector PDF files are the most useful.

Most drawing and graphing packages (Mathematica, Adobe Illustrator, Corel Draw, MATLAB, etc.) allow the user to save files in one of these formats. Make sure that what you are saving is vector graphics and not a bitmap. If you need help, please write to graphics@msp.org with details about how your graphics were generated.

White space. Forced line breaks or page breaks should not be inserted in the document. There is no point in your trying to optimize line and page breaks in the original manuscript. The manuscript will be reformatted to use the journal's preferred fonts and layout.

Proofs. Page proofs will be made available to authors (or to the designated corresponding author) at a Web site in PDF format. Failure to acknowledge the receipt of proofs or to return corrections within the requested deadline may cause publication to be postponed.

Communications in Applied Mathematics and Computational Science

vol. 14 no. 2 2019

1559-3940(2019)14:2;1-7