

*Communications in
Applied
Mathematics and
Computational
Science*

EFFICIENT MULTIGRID SOLUTION
OF ELLIPTIC INTERFACE PROBLEMS
USING VISCOSITY-UPWINDED
LOCAL DISCONTINUOUS GALERKIN METHODS

ROBERT I. SAYE

vol. 14 no. 2 2019

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 $u : \Omega \rightarrow \mathbb{R}$ such that

$$\left\{ \begin{array}{ll} -\nabla \cdot (\mu_i \nabla u) = f_i & \text{in } \Omega_i, \\ \llbracket u \rrbracket = g_{ij} & \text{on } \Gamma_{ij}, \\ \llbracket (\mu \nabla u) \cdot \mathbf{n} \rrbracket = h_{ij} & \text{on } \Gamma_{ij}, \\ u = g_{\partial} & \text{on } \Gamma_D, \\ (\mu \nabla u) \cdot \mathbf{n} = h_{\partial} & \text{on } \Gamma_N, \end{array} \right. \quad (1)$$

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, $\llbracket \cdot \rrbracket$ denotes the jump in a quantity across an interface and \mathbf{n} is to be understood from context — on $\partial\Omega$, \mathbf{n} denotes the outward unit normal to the domain boundary, whereas for an interface Γ_{ij} , \mathbf{n} denotes the unit normal to Γ_{ij} , oriented consistently with the definition of the jump operator $\llbracket \cdot \rrbracket$. 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 function¹ $\mu_i : \Omega_i \rightarrow \mathbb{R}^+$. 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 $\mathbf{q} = \mu \nabla u$. On a typical mesh face, the numerical fluxes u^* and \mathbf{q}^* 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

$$\begin{cases} -\nabla^2 u_1 = f_1 & \text{in } \Omega_1, \\ -\epsilon \nabla^2 u_\epsilon = f_\epsilon & \text{in } \Omega_\epsilon, \\ u_1 - u_\epsilon = g & \text{on } \Gamma, \\ \mathbf{n} \cdot \nabla u_1 - \epsilon \mathbf{n} \cdot \nabla u_\epsilon = h & \text{on } \Gamma, \end{cases}$$

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

¹Comments 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 $\mathcal{O}(1)$ 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 $\mathbf{n} \cdot \nabla u_1 \approx h$ 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^* 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 \mathbf{q}^* 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 $\mathbf{q} \cdot \mathbf{n} \approx h$ 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^* should bias

towards the phase with (locally) largest viscosity, (ii) \mathbf{q}^* 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,

²The 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 $\mathbf{q} = \mu \nabla u$ and rewrite the system as

$$\begin{cases} \mathbf{q} = \mu_i \nabla u & \text{in } \Omega_i, & \llbracket u \rrbracket = g_{ij} & \text{on } \Gamma_{ij}, & u = g_\partial & \text{on } \Gamma_D, \\ -\nabla \cdot \mathbf{q} = f_i & \text{in } \Omega_i, & \llbracket \mathbf{q} \cdot \mathbf{n} \rrbracket = h_{ij} & \text{on } \Gamma_{ij}, & \mathbf{q} \cdot \mathbf{n} = h_\partial & \text{on } \Gamma_N. \end{cases} \quad (2)$$

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 $\mathcal{E} = \bigcup_i E_i$ 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 $\mathcal{Q}_p(E)$ to be the space of tensor-product polynomials of degree p on the element E . For example, \mathcal{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

$$\begin{aligned} V_h(\mathcal{E}) &= \{v : \Omega \rightarrow \mathbb{R} \mid v|_E \in \mathcal{Q}_p(E) \text{ for every } E \in \mathcal{E}\}, \\ V_h^d(\mathcal{E}) &= \{\boldsymbol{\omega} : \Omega \rightarrow \mathbb{R}^d \mid \boldsymbol{\omega}|_E \in [\mathcal{Q}_p(E)]^d \text{ for every } E \in \mathcal{E}\}. \end{aligned}$$

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 \mathbf{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 *intrapphase 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 \mathbf{n} ; in this work, intraphase faces are always flat and lie in a particular coordinate plane so that \mathbf{n} is defined to point from “left-to-right”, e.g., for vertical faces in 2D, $\mathbf{n} = \hat{\mathbf{x}}$ and, for horizontal faces, $\mathbf{n} = \hat{\mathbf{y}}$. 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 $\llbracket \cdot \rrbracket$ denotes the jump of a quantity across an interface or face and is defined consistent with its orientation; in particular, $\llbracket u \rrbracket := u^- - u^+$ where $u^\pm(x) = \lim_{\epsilon \rightarrow 0^+} u(x \pm \epsilon \mathbf{n})$ denotes the left u^- and right u^+ trace values. Last, for an element $E \in \mathcal{E}$, 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

$$\int_E \boldsymbol{\eta} \cdot \boldsymbol{\omega} = \int_E \nabla u \cdot \boldsymbol{\omega} + \int_{\partial E} (u_{\chi(E)}^* - u) \boldsymbol{\omega} \cdot \mathbf{n} \quad (3)$$

holds for every element $E \in \mathcal{E}$ and every test function $\boldsymbol{\omega} \in V_h^d$. Here, u_χ^* 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):

$$u_\chi^* := \begin{cases} u^- & \text{on any intraphase face,} \\ \lambda u^- + (1 - \lambda)(u^+ + g_{\chi i}) & \text{on } \Gamma_{\chi i} \text{ if } \chi < i, \\ \lambda(u^- - g_{i\chi}) + (1 - \lambda)u^+ & \text{on } \Gamma_{i\chi} \text{ if } \chi > i, \\ u^- & \text{on } \Gamma_N, \\ g_\partial & \text{on } \Gamma_D. \end{cases} \quad (4)$$

(See Figure 1 for a schematic illustration.) Note that the flux is multivalued on interphase faces — on these faces, the interfacial jump condition $\llbracket u \rrbracket = g_{ij}$ 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

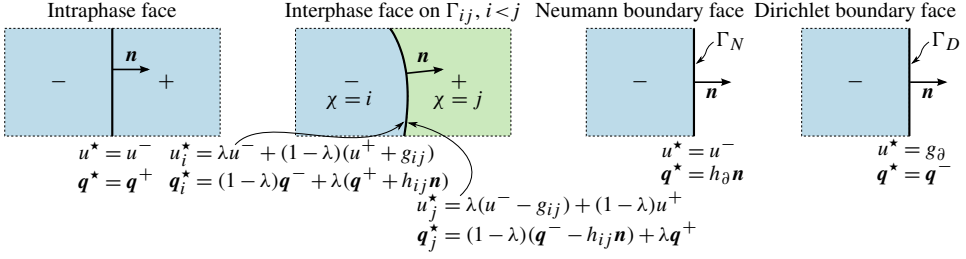


Figure 1. Schematic of the numerical flux functions u^* and q^* defined by (4) and (6). Except for interphase faces, the flux is single-valued; on interphase faces, the flux is multivalued so as to incorporate the interfacial jump conditions $\llbracket u \rrbracket = g_{ij}$ and $\llbracket \mathbf{q} \cdot \mathbf{n} \rrbracket = h_{ij}$ on $\Gamma_{ij}, i < j$. 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, $u^\pm(x) = \lim_{\epsilon \rightarrow 0^+} u(x \pm \epsilon \mathbf{n})$.

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^* equals g_{∂} on all Dirichlet boundary faces, and equals the boundary trace on all Neumann boundary faces.

Second, we define a discrete approximation to $\mathbf{q} \in V_h^d$, which is essentially $\boldsymbol{\eta}$ multiplied by the local ellipticity coefficient μ . To do so, we define \mathbf{q} as the L^2 projection of the function $\mu \boldsymbol{\eta}$, i.e., $\mathbf{q} \in V_h^d$ is the unique piecewise polynomial function such that

$$\int_E \mathbf{q} \cdot \boldsymbol{\omega} = \int_E \mu \boldsymbol{\eta} \cdot \boldsymbol{\omega} \quad (5)$$

holds for every element $E \in \mathcal{E}$ 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, \mathbf{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 \mathbf{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 \mathbf{q} is vector-valued; however, only the normal component of the flux is used.) Given $\mathbf{q} \in V_h^d$, define $w \in V_h$ as the discrete divergence of \mathbf{q} such that

$$\int_E wv = - \int_E \mathbf{q} \cdot \nabla v + \int_{\partial E} v \mathbf{q}_{\chi(E)}^* \cdot \mathbf{n}$$

holds for every test function $v \in V_h$ and every element $E \in \mathcal{E}$ with phase $\chi(E)$. Here, the numerical flux is defined by (see also Figure 1)

$$\mathbf{q}_\chi^\star := \begin{cases} \mathbf{q}^+ & \text{on any intraphase face,} \\ (1-\lambda)\mathbf{q}^- + \lambda(\mathbf{q}^+ + h_{\chi i}\mathbf{n}) & \text{on } \Gamma_{\chi i} \text{ if } \chi < i, \\ (1-\lambda)(\mathbf{q}^- - h_{i\chi}\mathbf{n}) + \lambda\mathbf{q}^+ & \text{on } \Gamma_{i\chi} \text{ if } \chi > i, \\ h_\partial\mathbf{n} & \text{on } \Gamma_N, \\ \mathbf{q}^- & \text{on } \Gamma_D. \end{cases} \quad (6)$$

As in the numerical flux for u^\star , the interfacial jump condition $[\![\mathbf{q} \cdot \mathbf{n}]\!] = h_{ij}$ 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 $E_g : V_h \rightarrow V_h$ be the operator such that, for each $u \in V_h$,

$$\int_{\Omega} E_g(u)v = \sum_i \int_{\Gamma_i} \tau_i [\![u]\!] [v] + \sum_{i < j} \int_{\Gamma_{ij}} \tau_{ij} ([\![u]\!] - g_{ij}) [v] + \int_{\Gamma_D} \tau_D (u^- - g_\partial) v^- \quad (7)$$

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 \mathbf{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., $\tau = \mathcal{O}(h^{-1})$ as $h \rightarrow 0$. 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 \mathbf{q} , (iii) computes the discrete divergence of \mathbf{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 derivation³ and instead state the final result wherein the auxiliary variable \mathbf{q} is eliminated: u solves the linear problem

$$\left(\sum_{i=1}^d G_i^T M_\mu G_i \right) u + M E_0 u = M \mathbb{P}_{V_h}(f) + J_h(h_{ij}, h_\partial) + J_g(g_{ij}, g_\partial) \quad (8)$$

where:

- $G = (G_1, \dots, G_d) : V_h \rightarrow V_h^d$ 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 $M^{-1} M_\mu u$.

³The 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 $\mu\boldsymbol{\eta}$.

⁴A 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 $D = -\text{adj}(G)$ 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, quadrees 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_{2h}^h transfers a piecewise polynomial function on a coarse mesh to a piecewise polynomial function on the fine mesh. In the present setting, I_{2h}^h is naturally defined by injection: $(I_{2h}^h u)|_{E_f} = u|_{E_c}$, 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_h^{2h} is defined to be the adjoint of the interpolation operator. Equivalently, for a piecewise polynomial function u on a fine mesh, $R_h^{2h} u$ is defined as the L^2 projection of u onto the coarse mesh. It is related to the interpolation operator via $R_h^{2h} = M_{2h}^{-1} (I_{2h}^h)^T M_h$ where M_h and M_{2h} are the mass matrices on the fine and coarse meshes, respectively, and $(I_{2h}^h)^T$ is the transpose of the interpolation operator matrix.
- *Coarsening of a general operator.* Given an operator $A : V_h \rightarrow V_h$ defined on a fine mesh, its coarsened counterpart on a coarse mesh is defined variationally, such that $\mathcal{C}(A) : V_{2h} \rightarrow V_{2h}$ satisfies

$$(\mathcal{C}(A)u, v)_{V_{2h}} = (AI_{2h}^h u, I_{2h}^h v)_{V_h}$$

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, $\mathcal{C}(A) = R_h^{2h} A I_{2h}^h$.

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, $D = -\text{adj}(G) = -M^{-1}G^T M$ 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 $-\mathcal{C}(D)\mathcal{C}(G) + \tau\mathcal{C}(E)$. 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 \mathcal{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 $\Theta_\mu : V_h \rightarrow V_h$ 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

$$-\mathcal{C}(D)\mathcal{C}(\Theta_\mu)\mathcal{C}(G) + \frac{1}{2}\mathcal{C}(E_\tau). \quad (9)$$

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 $\mathcal{C}(\Theta_\mu)$ 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 factor⁵ 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, $\tilde{E}_h := M_h E_h$ 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:

⁵In 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.

$$\begin{aligned}
M_{2h} &:= (I_{2h}^h)^T M_h I_{2h}^h \\
M_{\mu,2h} &:= (I_{2h}^h)^T M_{\mu,h} I_{2h}^h \\
G_{2h} &:= M_{2h}^{-1} (I_{2h}^h)^T M_h G_h I_{2h}^h \\
\tilde{E}_{2h} &:= \frac{1}{2} (I_{2h}^h)^T \tilde{E}_h I_{2h}^h \\
A_{2h} &:= G_{2h}^T M_{\mu,2h} G_{2h} + \tilde{E}_{2h}
\end{aligned}$$

Algorithm 1. Construction of coarse-mesh operators, given fine-mesh operators M_h , $M_{\mu,h}$, G_h , and \tilde{E}_h .

if \mathcal{E}_h is the bottom level **then**

Solve $A_h x_h = b_h$ with bottom solver

else

Apply smoother ν times

$$r_{2h} := (I_{2h}^h)^T (b_h - A_h x_h)$$

$$x_{2h} := V(\mathcal{E}_{2h}, 0, r_{2h})$$

$$x_h \leftarrow x_h + I_{2h}^h x_{2h}$$

Apply smoother (in reverse ordering) ν times

return x_h

Algorithm 2. Multigrid V-cycle $V(\mathcal{E}_h, x_h, b_h)$ on a mesh \mathcal{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 $h \rightarrow 0$.

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 $\Omega_1 = (\frac{1}{4}, \frac{3}{4})$ and an exterior phase $\Omega_2 = (0, \frac{1}{4}) \cup (\frac{3}{4}, 1)$, and let $\Gamma = \{\frac{1}{4}, \frac{3}{4}\}$ denote the interface between Ω_1 and Ω_2 . We consider the elliptic interface problem with Dirichlet boundary conditions

$$\left\{ \begin{array}{ll} -\mu_i \nabla^2 u = f_i & \text{in } \Omega_i, \\ \llbracket u \rrbracket = g & \text{on } \Gamma, \\ \llbracket \mu \nabla u \cdot \mathbf{n} \rrbracket = h & \text{on } \Gamma, \\ u = g_\partial & \text{on } \partial\Omega, \end{array} \right. \quad (10)$$

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 degree⁶ $p = 3$,

⁶For 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_θ in (10) are generated by the exact solution

$$u(x) = \begin{cases} \sin 4\pi(x - 0.1) & \text{in } \Omega_1, \\ \cos 4\pi(x - 0.1) & \text{in } \Omega_2. \end{cases} \quad (11)$$

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

⁷The 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.

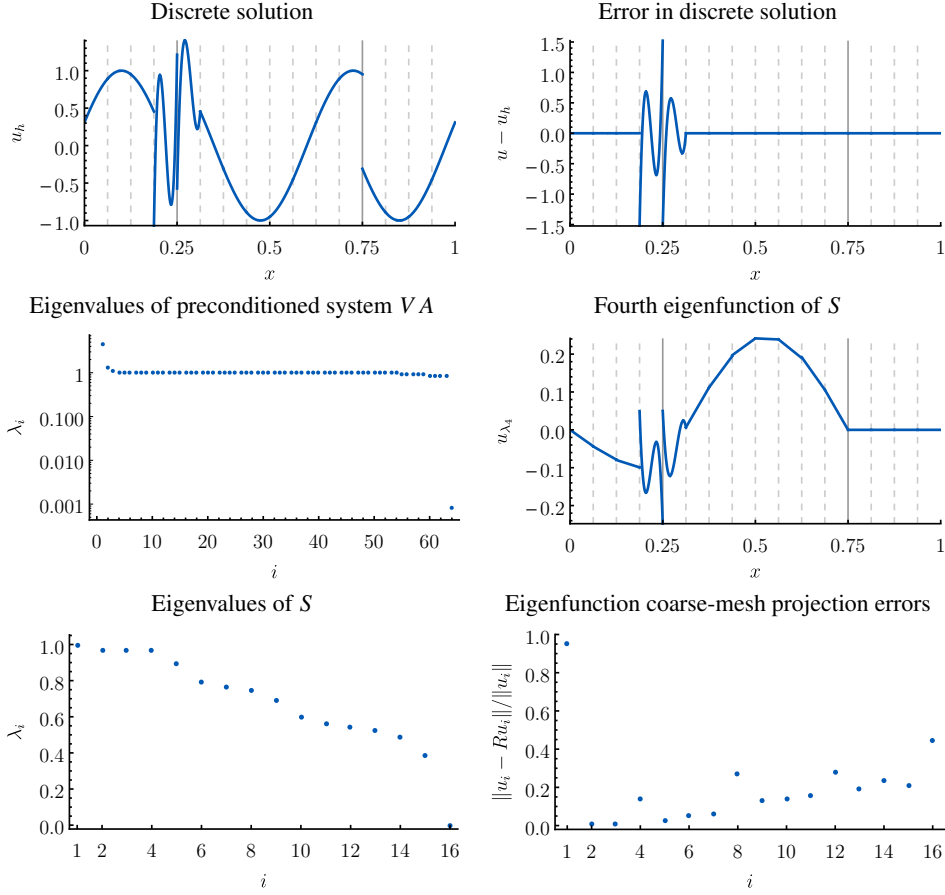


Figure 2. Solution accuracy and characteristics of multigrid performance for a two-phase, constant-coefficient elliptic interface problem in one dimension, wherein $\Omega_1 = (\frac{1}{4}, \frac{3}{4})$ has viscosity coefficient $\mu_1 = 10^{-5}$ and $\Omega_2 = (0, \frac{1}{4}) \cup (\frac{3}{4}, 1)$ 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 largest⁸ 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

⁸In 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

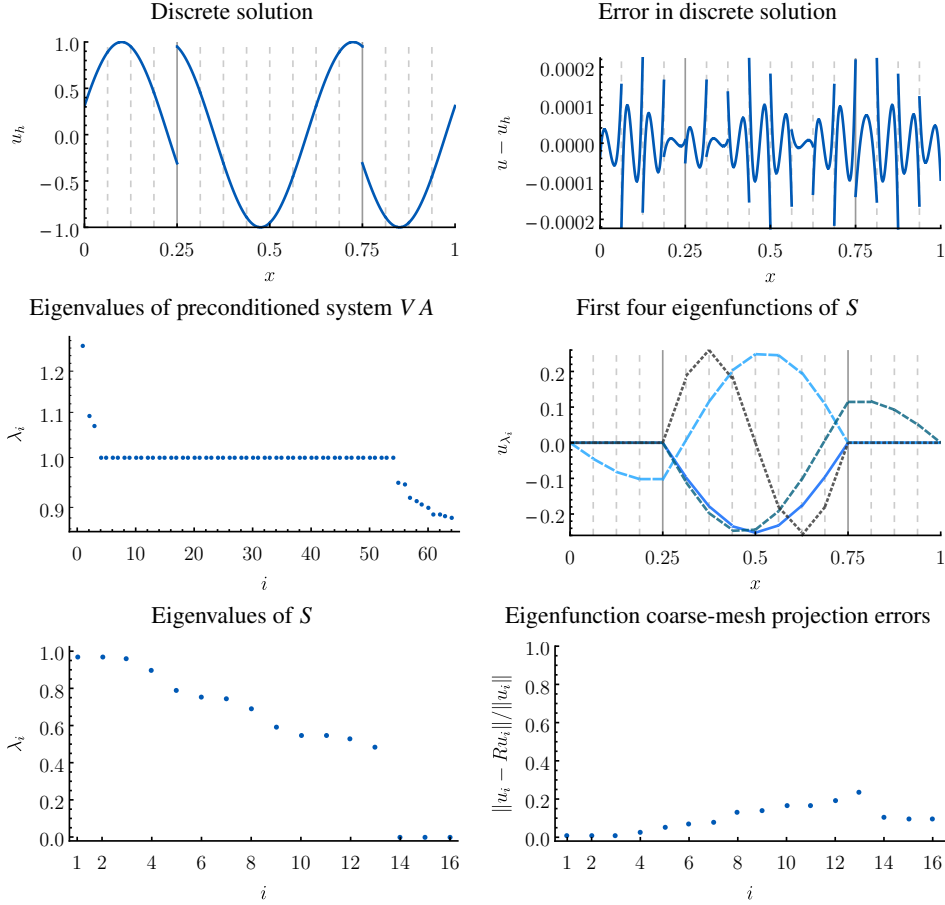


Figure 3. Solution accuracy and characteristics of multigrid performance for a two-phase, constant-coefficient elliptic interface problem in one dimension, wherein $\Omega_1 = (\frac{1}{4}, \frac{3}{4})$ has viscosity coefficient $\mu_1 = 10^{-5}$ and $\Omega_2 = (0, \frac{1}{4}) \cup (\frac{3}{4}, 1)$ 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^* biasing towards phase Ω_2 and q^* 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^* should bias towards the more viscous phase, and that for q^* should bias towards the less viscous phase.

- The condition number of the V-cycled preconditioned operator is minimized when $\lambda \lesssim \sqrt{\mu_1/\mu_2}$ if $\mu_1/\mu_2 < 1$, or $\lambda \gtrsim 1 - \sqrt{\mu_2/\mu_1}$ if $\mu_1/\mu_2 > 1$.

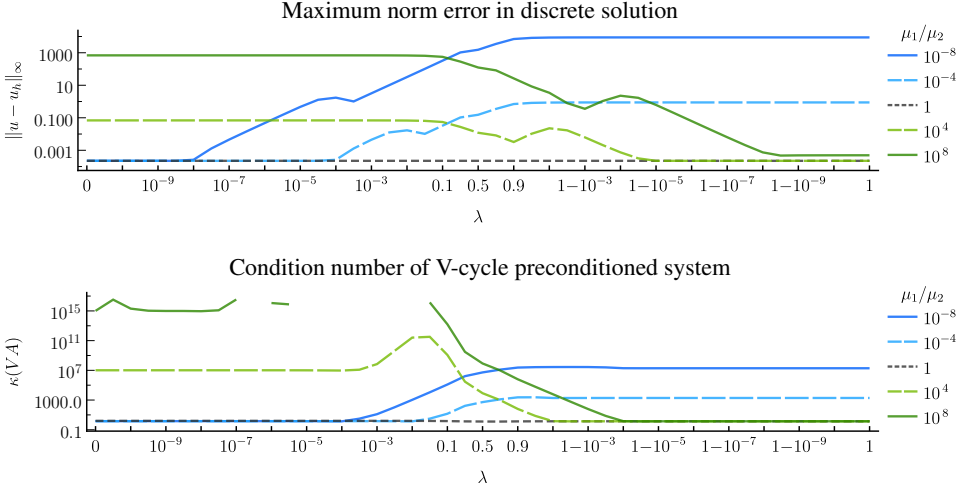


Figure 4. Solution accuracy and multigrid performance as a function of interfacial flux weighting for a two-phase, constant-coefficient elliptic interface problem in one dimension, wherein $\Omega_1 = (\frac{1}{4}, \frac{3}{4})$ has viscosity coefficient μ_1 and $\Omega_2 = (0, \frac{1}{4}) \cup (\frac{3}{4}, 1)$ 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 ($\lambda = 0.5$), 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 $\lambda \lesssim \mu_1/\mu_2$ if $\mu_1/\mu_2 < 1$, or $\lambda \gtrsim 1 - \mu_2/\mu_1$ 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 $\lambda \gtrsim 1 - C\mu_2/\mu_1$ 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

$$\lambda = \frac{\mu^-}{\mu^- + \mu^+}, \quad (12)$$

where μ^\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 $\lambda \approx \sqrt{\mu^-}/(\sqrt{\mu^-} + \sqrt{\mu^+})$ [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

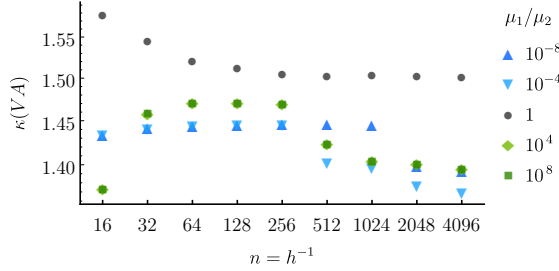


Figure 5. Multigrid performance under the action of mesh refinement for a two-phase, constant-coefficient elliptic interface problem in one dimension, wherein $\Omega_1 = (\frac{1}{4}, \frac{3}{4})$ has viscosity coefficient μ_1 and $\Omega_2 = (0, \frac{1}{4}) \cup (\frac{3}{4}, 1)$ has coefficient 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

$$\lambda = \frac{(\mu^-)^\alpha}{(\mu^-)^\alpha + (\mu^+)^\alpha} \quad (13)$$

where $\alpha > 1$ is a user-chosen parameter controlling the biasing strength. The limit $\alpha \rightarrow \infty$ corresponds to pure one-sided biasing, denoted in this work as *viscosity-upwinded weighting*,

$$\lambda = \begin{cases} 0 & \text{if } \mu^- < \mu^+, \\ 0.5 & \text{if } \mu^- = \mu^+, \\ 1 & \text{if } \mu^- > \mu^+. \end{cases} \quad (14)$$

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 $\Omega_1 = (\frac{1}{4}, \frac{3}{4})^d$ 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

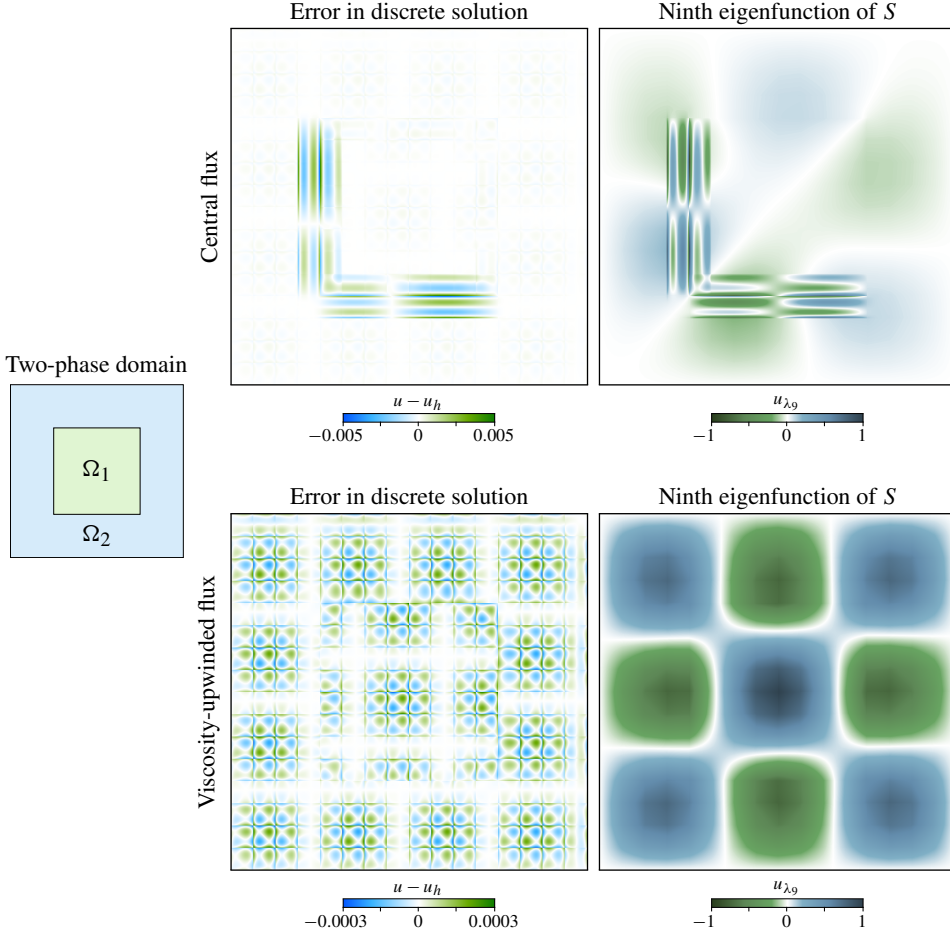


Figure 6. Solution accuracy and example eigenfunctions of the multigrid relaxation operator for a two-phase, constant-coefficient elliptic interface problem in two dimensions, wherein $\Omega_1 = (\frac{1}{4}, \frac{3}{4})^2$ has viscosity coefficient 10^{-5} and the exterior phase $\Omega_2 = (0, 1)^2 \setminus \overline{\Omega_1}$ 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 $16 \times 16 \times 16$ 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

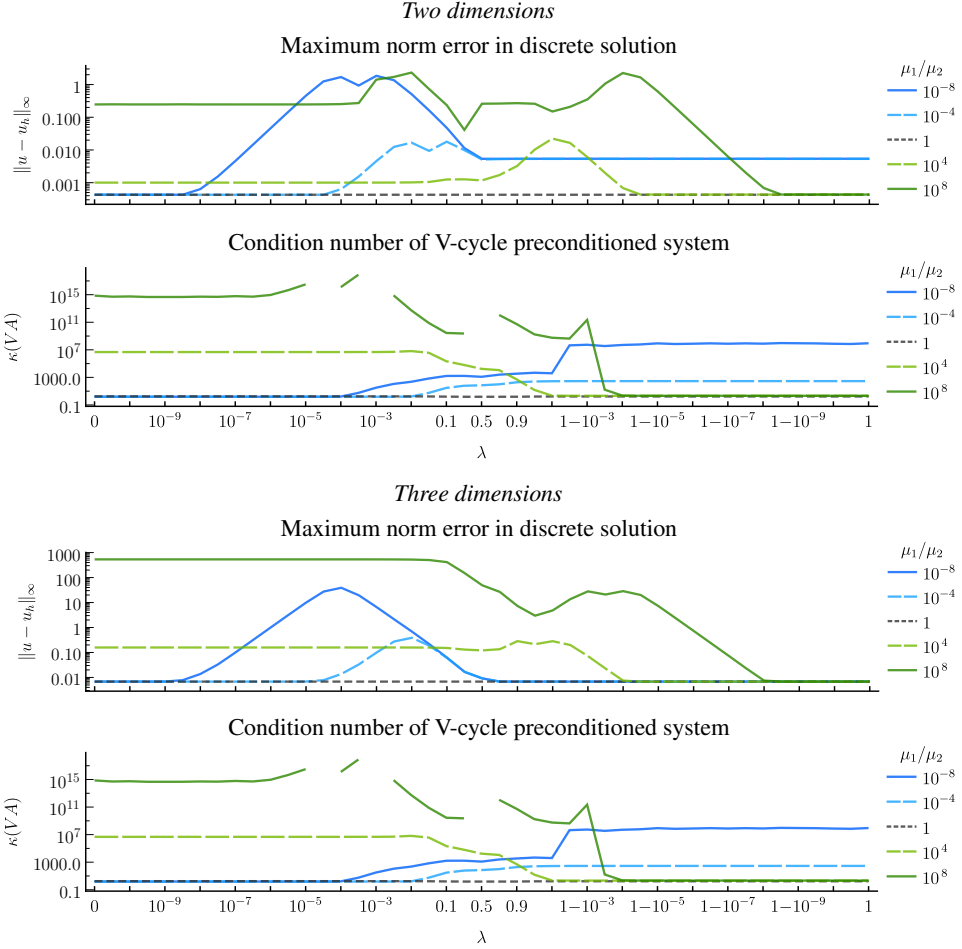


Figure 7. Solution accuracy and multigrid performance as a function of interfacial flux weighting for a two-phase, constant-coefficient elliptic interface problem in 2D and 3D, wherein $\Omega_1 = (\frac{1}{4}, \frac{3}{4})^d$ has viscosity coefficient μ_1 and $\Omega_2 = (0, 1)^d \setminus \overline{\Omega_1}$ 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 number⁹ of the V-cycle preconditioned system as a function of λ . Similar

⁹In 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

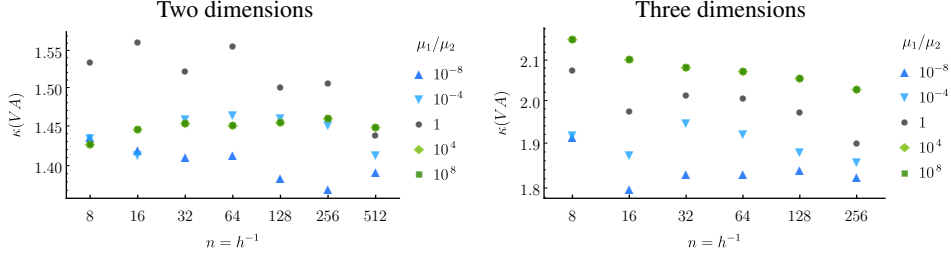


Figure 8. Multigrid performance under the action of mesh refinement for a two-phase, constant-coefficient elliptic interface problem in two and three dimensions, wherein $\Omega_1 = (\frac{1}{4}, \frac{3}{4})^d$ has viscosity coefficient μ_1 and $\Omega_2 = (0, 1)^d \setminus \overline{\Omega_1}$ has 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 $\lambda \lesssim 0.1(\mu_1/\mu_2)$ if $\mu_1/\mu_2 < 1$, or $\lambda \gtrsim 1 - 0.1(\mu_2/\mu_1)$ if $\mu_1/\mu_2 > 1$; meanwhile, the condition number is minimized when $\lambda \lesssim \sqrt{\mu_1/\mu_2}$ if $\mu_1/\mu_2 < 1$, or $\lambda \gtrsim 1 - \sqrt{\mu_2/\mu_1}$ 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.

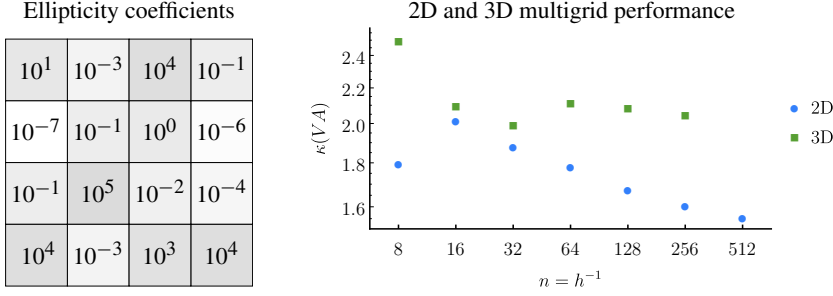


Figure 9. Left: a multiphase domain divided into a 4×4 array of subdomains with viscosity coefficient μ_i as indicated. Right: multigrid performance under the action of mesh refinement for the corresponding multiphase, constant-coefficient elliptic interface problem in two dimensions and a 3D analogue. 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 $h \rightarrow 0$ 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 $\Omega_2 = (0, 1) \times (\frac{1}{4}, \frac{3}{4})$ and an exterior phase $\Omega_1 = (0, 1)^2 \setminus \overline{\Omega_2}$, such that $\mu = \mu(x, y)$ is given by

$$\mu = \begin{cases} 10^{-4+8 \sin \pi x/2} & \text{in } \Omega_1, \\ 10^{4-8 \sin \pi x/2} & \text{in } \Omega_2. \end{cases} \quad (15)$$

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 $\Omega_2 = (0, 1) \times (0, 1) \times (\frac{1}{4}, \frac{3}{4})$ 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 $h \rightarrow 0$.

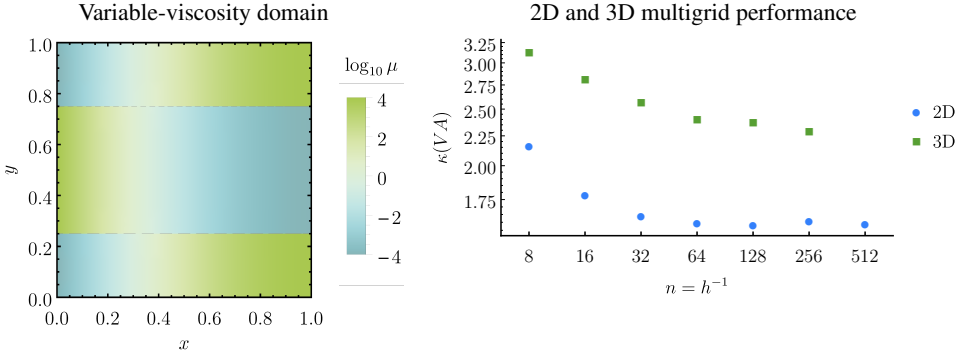


Figure 10. Left: a two-phase domain with variable ellipticity coefficient given by (15); the interface separating the two phases is $\Gamma = \{(x, y) : y = \frac{1}{4} \text{ or } y = \frac{3}{4}\}$. Right: multigrid performance under the action of mesh refinement for the corresponding two-phase, variable-coefficient 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

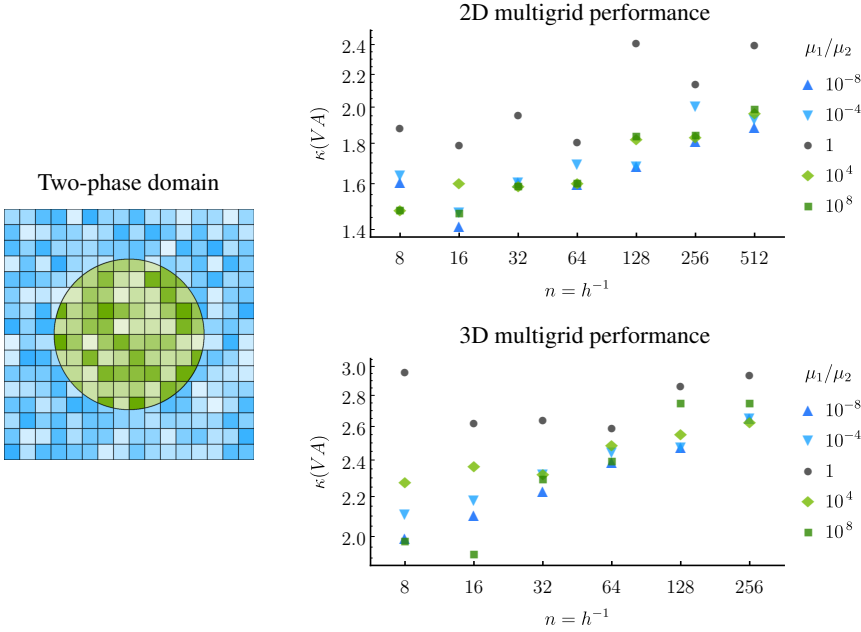


Figure 11. Left: an implicitly defined mesh corresponding to the two-phase circular interface test problem of Section 5.4 generated with a background 16×16 grid. Right: multigrid performance under the action of mesh refinement for the associated two-phase, constant-coefficient elliptic interface problem. 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

¹⁰Nonuniform 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.

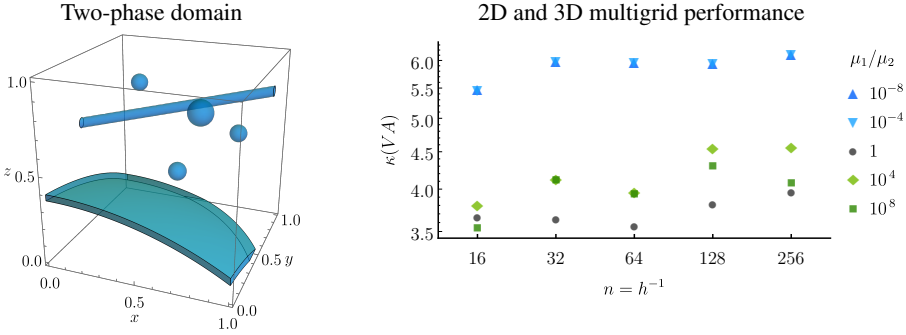


Figure 12. Left: a two-phase domain divided into the indicated shaded phase Ω_1 , consisting of a thin sheet, a long thin filament, and three droplets; the secondary phase Ω_2 fills the exterior region. Right: multigrid performance under the action of mesh refinement for the geometry depicted in the 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 $h \rightarrow 0$.

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,

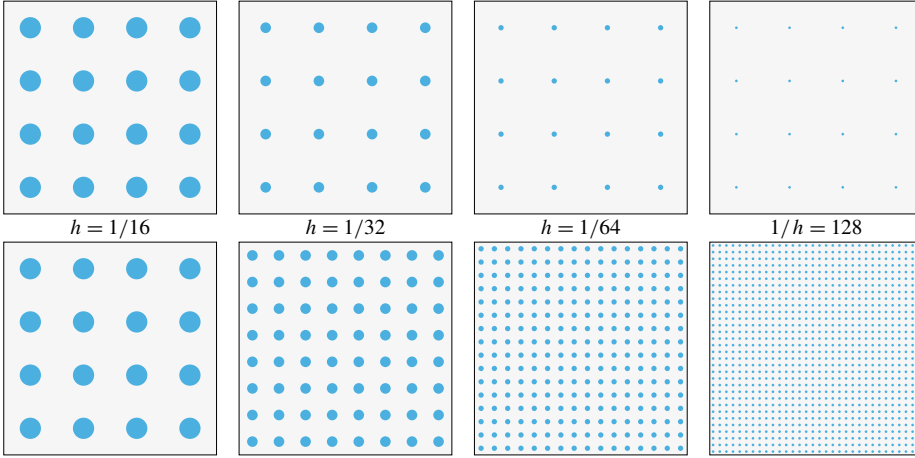


Figure 13. Illustration of the “bubbly” geometry of the elliptic interface problems considered in Section 5.6 where $h = 1/n$ is the cell size of the background uniform Cartesian grid used to define the corresponding implicitly defined meshes. Top: fixed lattice of vanishing droplets. Bottom: growing 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 $h \rightarrow 0$, 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 $h \rightarrow 0$ 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

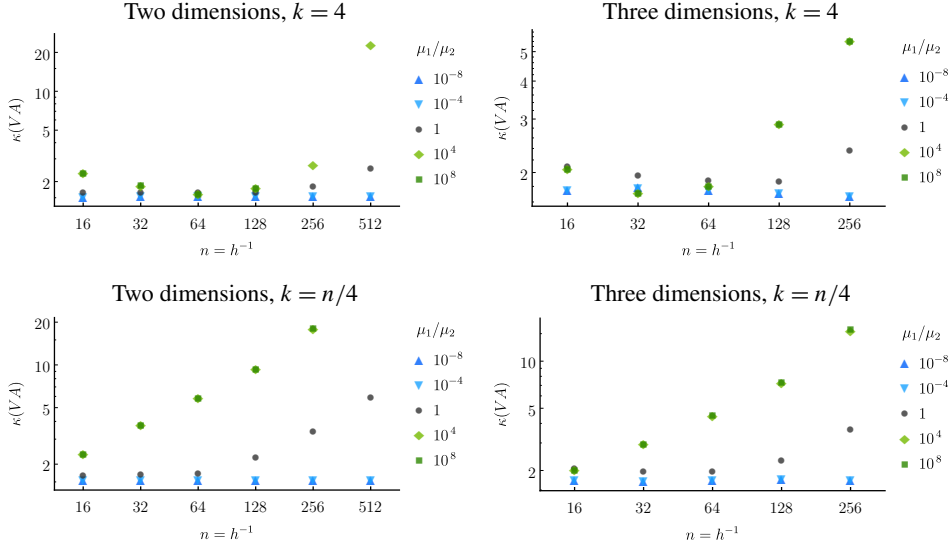


Figure 14. Multigrid performance under the action of mesh refinement for the two-phase elliptic interface problems with “bubbly” geometry shown in Figure 13, consisting of a $k \times k (\times k)$ lattice of droplets of vanishing size. In the top row, the lattice size is fixed at $k = 4$; in the bottom 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 $h \rightarrow 0$. 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 $\mathbf{q} = \mu \nabla u$ 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 $h \rightarrow 0$. 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 $h \rightarrow 0$ 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 $\mathbf{n} \cdot \mu^\pm \cdot \mathbf{n}$, where \mathbf{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

Communications in Applied Mathematics and Computational Science

msp.org/camcos

EDITORS

MANAGING EDITOR

John B. Bell
Lawrence Berkeley National Laboratory, USA
jbbell@lbl.gov

BOARD OF EDITORS

Marsha Berger	New York University berger@cs.nyu.edu	Ahmed Ghoniem	Massachusetts Inst. of Technology, USA ghoniem@mit.edu
Alexandre Chorin	University of California, Berkeley, USA chorin@math.berkeley.edu	Raz Kupferman	The Hebrew University, Israel raz@math.huji.ac.il
Phil Colella	Lawrence Berkeley Nat. Lab., USA pcollella@lbl.gov	Randall J. LeVeque	University of Washington, USA rjl@amath.washington.edu
Peter Constantin	University of Chicago, USA const@cs.uchicago.edu	Mitchell Luskin	University of Minnesota, USA luskin@umn.edu
Maksymilian Dryja	Warsaw University, Poland maksymilian.dryja@acn.waw.pl	Yvon Maday	Université Pierre et Marie Curie, France maday@ann.jussieu.fr
M. Gregory Forest	University of North Carolina, USA forest@amath.unc.edu	James Sethian	University of California, Berkeley, USA sethian@math.berkeley.edu
Leslie Greengard	New York University, USA greengard@cims.nyu.edu	Juan Luis Vázquez	Universidad Autónoma de Madrid, Spain juanluis.vazquez@uam.es
Rupert Klein	Freie Universität Berlin, Germany rupert.klein@pik-potsdam.de	Alfio Quarteroni	Politecnico di Milano, Italy alfio.quarteroni@polimi.it
Nigel Goldenfeld	University of Illinois, USA nigel@uiuc.edu	Eitan Tadmor	University of Maryland, USA etadmor@cscamm.umd.edu
		Denis Talay	INRIA, France denis.talay@inria.fr

PRODUCTION

production@msp.org

Silvio Levy, Scientific Editor


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/>

© 2019 Mathematical Sciences Publishers

Communications in Applied Mathematics and Computational Science

vol. 14

no. 2

2019

Simple second-order finite differences for elliptic PDEs with discontinuous coefficients and interfaces	121
CHUNG-NAN TZOU and SAMUEL N. STECHMANN	
2D force constraints in the method of regularized Stokeslets	149
ONDREJ MAXIAN and WANDA STRYCHALSKI	
Potential field formulation based on decomposition of the electric field for a nonlinear induction hardening model	175
TONG KANG, RAN WANG and HUAI ZHANG	
Computing the quasipotential for highly dissipative and chaotic SDEs an application to stochastic Lorenz'63	207
MARIA CAMERON and SHUO YANG	
Efficient multigrid solution of elliptic interface problems using viscosity-upwinded local discontinuous Galerkin methods	247
ROBERT I. SAYE	



1559-3940(2019)14:2;1-7