Higher-order accurate diffuse-domain methods for partial differential equations with Dirichlet boundary conditions in complex, evolving geometries

Fei Yu[a], Zhenlin Guo[a], John Lowengrub[a,b]

[a]*Department of Mathematics, University of California, Irvine, CA, 92697, United States*
[b]*Department of Mathematics and Department of Biomedical Engineering, University of California, Irvine, CA, 92697, United States*

Abstract

The diffuse-domain, or smoothed boundary, method is an attractive approach for solving partial differential equations in complex geometries because of its simplicity and flexibility. In this method the complex geometry is embedded into a larger, regular domain. The original PDE is reformulated using a smoothed characteristic function of the complex domain and source terms are introduced to approximate the boundary conditions. The reformulated equation, which is independent of the dimension and domain geometry, can be solved by standard numerical methods and the same solver can be used for any domain geometry. A challenge is making the method higher-order accurate. For Dirichlet boundary conditions, which we focus on here, current implementations demonstrate a wide range in their accuracy but generally the methods yield at best first order accuracy in ϵ, the parameter that characterizes the width of the region over which the characteristic function is smoothed. Typically, $\epsilon \propto h$, the grid size. Here, we analyze the diffuse-domain PDEs using matched asymptotic expansions and explain the observed behaviors. Our analysis also identifies simple modifications to the diffuse-domain PDEs that yield higher-order accuracy in ϵ, e.g., $O(\epsilon^2)$ in the L^2 norm and $O(\epsilon^p)$ with $1.5 \leq p \leq 2$ in the L^∞ norm. Our analytic results are confirmed numerically in stationary and moving domains where the level set method is used to capture the dynamics of the domain boundary and to construct the smoothed characteristic function.

1. Introduction

Many problems in the physical, biological and engineering sciences involve solving partial differential equations (PDEs) in complex geometries that change their size and shape in time. There are two main approaches for solving such problems. In one approach, interface-fitted meshes are used. Examples include finite element methods (e.g., [1]), boundary integral (e.g., [2]) and boundary element (e.g., [3]) methods. Because of the challenges associated with generating interface-fitted meshes for complex geometries, especially in three-dimensions, and because in many applications the complex geometry evolves in time, which would require a new mesh to be generated at each time step, embedded domain methods have been developed as an alternative approach.

In this second approach, the complex domain is embedded into a larger, regular domain and the boundary conditions are approximated by a variety of different techniques. Examples include the adaptive fast multipole accelerated Poisson solver (e.g., [4]), which combines boundary and volume integral methods in the larger domain, fictitious domain methods (e.g., [5, 6, 7, 8]) where Lagrange multipliers are applied in order to enforce the boundary conditions, immersed boundary (e.g., [9, 10, 11, 12]), front-tracking (e.g., [13, 14, 15]) and arbitrary Lagrangian-Eulerian methods (e.g., [16, 17, 18, 19]) utilize separate surface and volume meshes where force distributions are interpolated from the surface to the volume meshes, in a neighborhood of the domain boundary, to approximate the boundary conditions. In addition, a number of specialized methods have been designed to achieve better than first order accuracy in the L^∞ norm. These include the immersed interface (e.g., [20, 21, 22, 23][24, 25]), ghost fluid (e.g., [26, 27, 28, 29]), cut-cell methods (e.g., [30, 31, 32, 33]) and Voronoi interface (e.g., [34, 35]) methods. These methods modify difference stencils near the

2

domain boundary to account for the boundary conditions. Further, the extended finite element method (e.g., [36, 37, 38]) approximates the boundary conditions by enriching the space of test functions. A related approach is the virtual node method [39, 40], which can achieve 2nd order accuracy in L^∞ and involves adding (virtual) degrees of freedom along the interface together with Lagrange multipliers to enforce the boundary conditions. Another related approach is the active penalty method [41], which can achieve higher order accuracy in the L^∞ norm by extending the boundary function off the interface such that the extension matches the physical function and its normal derivatives at the interface. The smoothness of the solution and the number of matched normal derivatives constrains the accuracy of the method. A disadvantage of most of the methods described above is that either modifications of standard finite element and finite difference software packages are needed, or extra auxiliary equations have to be solved.

Diffuse-domain methods (DDM), also known as smoothed boundary methods, have emerged as an attractive alternative approach because they are easy to implement and the formulation does not depend on the dimension of the problem or the geometry of the domain. In the DDM, the complex geometry is embedded into a larger, regular domain and a phase-field function is used to provide a smooth approximation to the characteristic function of the complex domain. A parameter ϵ, usually proportional to the grid size, is introduced that characterizes the width of the diffuse interfacial region and typically controls the accuracy of the approximation. The original PDE is reformulated with additional source terms that enforce the boundary conditions. When ϵ is small, the DDM is most efficient when combined with adaptive mesh refinement to enable the use of small grid cells to resolve the narrow ($\approx O(\epsilon)$) transition layer and large grid cells in the extended, non-physical part of the regular domain. While DDMs are not always as accurate as some of the specialized methods described above and involve the solution of equations with non-constant coefficients and introduce additional length scales, DDMs have the advantage that they can be easily formulated for a wide range of equations and solved using standard uni-

3

form or adaptive discretizations combined with iterative matrix solvers, which are normally contained in standard PDE software packages. Further, the same solver can be used for any domain geometry.

The DDM was first used in [42, 43, 44] to solve diffusion equations with Neumann (no-flux) boundary conditions. The DDM was extended to simulate PDEs on surfaces in [45], to PDEs with Robin and Dirichlet boundary conditions in [46] and to cases in which bulk and surface equations are coupled [47]. Later, in [48] and [49] alternate derivations of diffuse-domain methods for such problems were presented. Diffuse-domain methods have been applied to a wide variety of problems that arise, for example, in biology (e.g., [42, 50, 51, 52, 53, 54, 55, 56, 57, 58]), in fluid dynamics (e.g., [59, 60, 61, 62, 63]) and in materials science (e.g., [64, 55, 65, 66, 67]), just to name a few, and have been implemented using a wide spectrum of methods including in-house finite-difference, finite-element and pseudo-spectral algorithms and software packages such as Matlab, AMDiS [68], MRAG [69], and BSAM [70].

Using rigorous mathematical theory [71, 72, 73, 74], matched asymptotic expansions and numerical simulations (e.g., [45, 46, 47, 48, 49]), the DDMs have been shown to converge to the original PDE and boundary conditions as the diffuse interface parameter ϵ tends to zero. Further, in [75] a matched asymptotic analysis for general DDMs with Neumann and Robin boundary conditions showed that for certain choices of the source terms, the DDMs were second-order accurate in ϵ and in the grid size h in both the L^2 and L^∞ norms, taking $\epsilon \propto h$; see the recent paper [74] for a rigorous proof.

For Dirichlet boundary conditions it is more challenging to obtain higher-order accurate diffuse domain methods. Current implementations of DDMs demonstrate a wide range in their accuracy but generally the methods yield at best first-order accuracy in ϵ [71, 48, 73, 74, 49]. For example, some schemes, including one presented in [46], achieve sub-first order accuracy: $O(\epsilon \ln \epsilon)$ in both the L^2 and the L^∞ norms. While others, including some presented in [46, 48, 49], achieve first-order accuracy: $O(\epsilon)$ in both the L^2 and L^∞ norms. Further, in [49] it was shown using asymptotic analysis that for several DDM

4

approximations of the diffusion equation with Dirichlet boundary conditions, that while the order of accuracy was $O(\epsilon)$ in L^2 the error could be significantly reduced by special choices of the DDM parameters. However, because DDMs for Dirichlet boundary conditions derive from introducing penalty terms in the equation, which are inherently low-order accurate, new modifications must be introduced to achieve higher-order accuracy.

In this paper, we analyze, using matched asymptotic expansions, several DDMs for approximating PDEs with Dirichlet boundary conditions. Our analysis explains the wide variation in the observed accuracy of the schemes and identifies simple modifications of the DDMs that can yield higher-order accuracy in ϵ. Similar to the active penalty method [41], our modifications are designed to enforce continuity of the normal derivative of the physical solution and its extension off the interface. However, we match the normal derivative only at the leading order in the asymptotic expansion in ϵ, which simplifies the method. Our new schemes are $O(\epsilon^2)$ in the L^2 norm and $O(\epsilon^p)$ in the L^∞ norm, with p ranging from 1.5 to 2 depending on the range of ϵ and h considered. For example, in the limit, $\epsilon \to 0$ and $\epsilon/h \to 0$ we obtain $p \to 1.5$. Surprisingly, the schemes can be more accurate when $\epsilon \propto h$. We find this can occur because the truncation/discretization error $u_{h,\epsilon} - u_\epsilon$ and the analytic error $u_\epsilon - u$ can have similar magnitudes but opposite signs so that there can be cancellation in the total error $u_{h,\epsilon} - u$. As h decreases, the error eventually becomes dominated by the analytic error.

Our formulation is dimension-independent, easy to implement and independent of computational methodology. Here we use the level-set method [76, 77] to generate the phase-field, or smoothed characteristic, function, and adaptive finite-difference discretizations with a mass-conserving multigrid method on block-structured adaptive meshes (BSAM) [70] to obtain highly-efficient schemes. We present examples in $1D$, $2D$ and $3D$ in stationary and moving domains and confirm our analytic results.

The outline of the paper is as follows. In Sec. 2, we analyze DDM approximations to the Poisson equation with Dirichlet boundary conditions and present

1D numerical results that confirm our analysis. In Sec. 3, we extend our analysis and simulations to time-dependent equations, focusing on the heat equation with external forcing. We consider both stationary and moving domains. We discuss different approaches to model the moving domains (phase-field versus level-set) and present 1D numerical simulations using the method. 2D and 3D numerical results are provided in Secs. 4 and 5, respectively. Finally, in Sec. 7, we present conclusions and discuss future work. In Appendixes A and B, we provide additional numerical results and theoretical details.

2. Diffuse domain method to the Poisson equation with Dirichlet boundary conditions

Consider the Poisson equation with Dirichlet boundary condition on a complex (non-regular) domain D:

$$\Delta u = f \quad in \quad D, \tag{1}$$

$$u = g \quad on \quad \partial D. \tag{2}$$

To approximate Eqs. (1) and (2), we consider three DDM approximations defined in [46] and posed on a larger, regular domain Ω:

$$\textbf{DDM1}: \quad \nabla \cdot (\phi \nabla u_\epsilon) - \frac{1}{\epsilon^3}(1-\phi)(u_\epsilon - g) = \phi f, \tag{3}$$

$$\textbf{DDM2}: \quad \phi \Delta u_\epsilon - \frac{1}{\epsilon^2}(1-\phi)(u_\epsilon - g) = \phi f, \tag{4}$$

$$\textbf{DDM3}: \quad \nabla \cdot (\phi \nabla u_\epsilon) - \frac{|\nabla \phi|}{\epsilon^2}(u_\epsilon - g) = \phi f, \tag{5}$$

where the source terms $\frac{1}{\epsilon^3}(1-\phi)(u_\epsilon - g)$, $\frac{1}{\epsilon^2}(1-\phi)(u_\epsilon - g)$ and $\frac{|\nabla \phi|}{\epsilon^2}(u_\epsilon - g)$ represent different choices for enforcing the Dirichlet boundary condition $u = g$. The function ϕ approximates the characteristic function of D, e.g.

$$\phi(\boldsymbol{x}, t) = \frac{1}{2}(1 - \tanh\left(\frac{3r(\boldsymbol{x}, t)}{\epsilon}\right)), \tag{6}$$

here $r(\boldsymbol{x}, t)$ is the signed-distance function to ∂D, which is taken to be negative inside D, and ϵ is the interface thickness. The interface thickness ϵ should be

smaller than any physically relevant length scale. In particular, $\epsilon << 1/|\kappa|$ where κ is the total curvature of the interface and $\epsilon < ||u||_{\partial D,\infty}/||\nabla u||_{\partial D,\infty}$ where the norms denote the L^∞ norm on the boundary ∂D. In addition, we require $\epsilon < \delta$, where $\delta = \delta(\kappa)$ is the width of a layer around the boundary where the functions g and f can be smoothly extended outside D constant in the normal direction. The distance over which g and f need to be extended, and the distance between ∂D and $\partial\Omega$, depends on the speed at which the approximate characteristic function ϕ asymptotes. Generally, this distance should be at least 3ϵ so that ϕ is sufficiently close to zero. Accordingly, we take $\delta = 3\epsilon$ and find no significant improvement in the errors when larger values of δ are used. To be safe, we took the distance between ∂D and $\partial\Omega$ to be $\approx 10\epsilon$ and find this works well for all the numerical simulations in this paper, although we likely could have used a smaller value.

Note that for DDM3, since the phase field function ϕ and its gradient vanish rapidly outside D, in order to prevent the equation from being ill-posed, we use the following modified gradient instead in the numerical calculation [75],

$$|\nabla\hat{\phi}| = \tau + (1-\tau)|\nabla\phi|, \tag{7}$$

where $\tau = 10^{-15}$. Note that no such regularization is required for DDMs 1 and 2 because the $1 - \phi \approx 1$ terms dominate outside D. In all the DDMs we consider, we assume that the boundary function g is defined in a small $O(\epsilon)$ neighborhood of ∂D using an extension off ∂D that is constant in the normal direction and that f is similarly extended smoothly out of the domain D. This can always be done if ϵ is sufficiently small as described above.

2.1. Numerical results for DDM1-3

We present representative numerical results in 1D for the DDMs given in Sec. 2. We suppose the original Poisson equation is defined in $D = [-1.111, 1.111]$ with Dirichlet boundary conditions at $x = \pm1.111$, which are not grid points. We take the forcing function $f = 1$ and boundary condition $g = 1.111^2/2$ so

7

that the exact solution to Eqs. (1)-(2) is $u = x^2/2$. The results from other test cases can be found in Appendix A.

We solve the DDMs in a larger domain $\Omega = [-2, 2]$ with the same Dirichlet boundary condition as the original Poisson equation. Note that the choice for boundary condition for the outer domain Ω should not affect the inner domain D. Therefore the distance between ∂D and $\partial \Omega$ should be large compared to ϵ. The equations are discretized on a uniform grid with the standard second-order central difference scheme (adaptive mesh refinement is used later in 2D, see Sec. 4). The discrete system is solved using the Thomas method [78].

The numerical solutions of DDM1-3, together with the exact solution, are shown in Fig. 1(a), a close-up of the solutions with $x \in [-0.1, 0.1]$ is shown as an inset. Here, we have taken $\epsilon = 0.0125$ and $h = \epsilon/4$. We next calculate and compare the error between the simulated DDM solutions $u_{h,\epsilon}$ and the analytic solution u of the original PDE. We present the errors in both the L^2 and the L^∞ norms, defined as

$$E_\epsilon^{(2)} = \frac{\|\phi(u - u_{h,\epsilon})\|_{L^2(\Omega)}}{\|\phi u\|_{L^2(\Omega)}}, \tag{8}$$

where $\|\cdot\|_{L^2}$ is the discrete L^2 norm:

$$\|u\|_{L^2} = \sqrt{1/N \sum_{i=1}^{N} u_i^2}, \tag{9}$$

where N is the number of grid points in Ω. The error in the L^∞ norm is defined as [20]

$$E_\epsilon^{(\infty)} = \frac{\|(u - u_{h,\epsilon})\|_{L^\infty(D)}}{\|u\|_{L^\infty(D)}}, \tag{10}$$

and $\|u\|_{L^\infty(D)} = \max_{1 \leq i \leq M} |u_i|$ and M is the number of grid points in D . Note this is not L^∞ in Ω. The convergence rate in ϵ is calculated as

$$\boldsymbol{k} = \frac{\log(\frac{E_{\epsilon_i}}{E_{\epsilon_{i-1}}})}{\log(\frac{\epsilon_i}{\epsilon_{i-1}})}. \tag{11}$$

The results are presented in Tabs. 1 and 2. We observe that DDM2 is 1st order accurate, which is consistent with findings in [48, 49], while the convergence

8

rates of DDM1 and DDM3 are less than 1 in both L^2 and L^∞ norms (see also Tabs. 7-12 in Appendix A for other test cases).

(a)   (b)

Figure 1: (a). Numerical solutions of DDM1–3 from Eqs. (3)-(5); (b). The corresponding modified diffuse domain methods mDDM1-3 from Eqs. (77)-(79); with $f = 1$, $g = 1.111^2/2$, $\epsilon = 0.0125$ and $h = \epsilon/4$.

	DDM1				DDM3			
ϵ	$E^{(2)}$	$\boldsymbol{k}$	$E^{(\infty)}$	$\boldsymbol{k}$	$E^{(2)}$	$\boldsymbol{k}$	$E^{(\infty)}$	$\boldsymbol{k}$
2.00E-01	4.88E-01	0.00	2.12E-01	0.00	3.82E-02	0.00	4.73E-02	0.00
1.00E-01	1.12E-01	2.12	5.26E-02	2.01	9.45E-02	-1.31	4.42E-02	0.10
5.00E-02	9.88E-03	3.51	7.96E-03	2.72	8.03E-02	0.24	3.66E-02	0.27
2.50E-02	1.31E-02	-0.41	5.89E-03	0.44	5.39E-02	0.58	2.44E-02	0.59
1.25E-02	1.41E-02	-0.11	6.31E-03	-0.10	3.33E-02	0.69	1.50E-02	0.70
6.25E-03	1.04E-02	0.44	4.66E-03	0.44	1.97E-02	0.76	8.84E-03	0.76
3.13E-03	6.80E-03	0.62	3.04E-03	0.62	1.14E-02	0.79	5.09E-03	0.80
1.56E-03	4.16E-03	0.71	1.86E-03	0.71	6.42E-03	0.82	2.87E-03	0.82

Table 1: The L^2 and L^∞ errors in DDM1 and DDM3 with $h = \epsilon/4$.

ϵ	DDM2			
	$E^{(2)}$	k	$E^{(\infty)}$	k
2.00E-01	3.42E-01	0.00	1.65E-01	0.00
1.00E-01	1.70E-01	1.01	7.95E-02	1.05
5.00E-02	8.45E-02	1.01	3.88E-02	1.04
2.50E-02	4.17E-02	1.02	1.91E-02	1.02
1.25E-02	2.07E-02	1.01	9.51E-03	1.00
6.25E-03	1.03E-02	1.00	4.82E-03	0.98
3.13E-03	5.16E-03	1.00	2.40E-03	1.00
1.56E-03	2.58E-03	1.00	1.20E-03	1.01

Table 2: The L^2 and L^∞ errors in DDM2 with $h = \epsilon/4$.

To understand this behavior, the scaled errors near the boundary $x = 1.111$ are shown in Fig. 2. In Fig. 2(a), we plot $\frac{u_\epsilon - u}{\epsilon}$ versus the stretched inner variable $z = \frac{r}{\epsilon}$ ($r = x - 1.111$) using $\epsilon = 0.00625$ and $N = 2560$, which is obtained from the grid size $h = \frac{\epsilon}{4}$. When $z << 0$ (inside the domain), $\frac{u_\epsilon - u}{\epsilon}$ tends to a constant, $C(\epsilon)$. In Fig. 2(b), the constants $C(\epsilon)$ from DDM1 and DDM3 are plotted as a function of $\ln(\epsilon)$. Linear fits give that $C(\epsilon) \approx -0.186 \ln(\epsilon) - 0.475$ for DDM1 and $C(\epsilon) \approx -0.188 \ln(\epsilon) - 0.143$ for DDM3, which suggests that the orders of accuracy of DDM1 and DDM3 are $\epsilon \ln(\epsilon)$ as $\epsilon \to 0$. In Fig. 2(c), we plot the limit value $C(\epsilon)$ from DDM2 as a function of ϵ. The plots suggest that $C(\epsilon) \to -0.45 \neq 0$, which implies DDM2 is 1st order accurate.

(a)

(b)   (c)

Figure 2: Comparisons of the behavior of DDM1, DDM2 and DDM3 near ∂D. (a): $\frac{u_\epsilon - u}{\epsilon}$ with $\epsilon = 0.00625$, (b): $C(\epsilon)$ for DDM1 and DDM3 (see text), (c): $C(\epsilon)$ for DDM2.

2.2. Matched asymptotic analysis of DDMs

In this section, we perform a matched asymptotic analysis in 1D for DDM1-3 in order to explain the numerical results in Sec. 2.1. This analysis can be easily extended to higher dimensions. Without loss of generality, we also assume the boundary is located at $x = 0$ and we take the signed distance (near the boundary) to be $r = x$, again $x < 0$ denotes the interior of D. We consider the expansion of diffuse-domain variables in powers of the interface thickness ϵ in regions close to and far from the interface, which are known as inner ($\hat{u}$) and

11

outer ($\bar{u}$) expansions, respectively. On each side of the interface, there exists an outer expansion, here labeled $\bar{u}_1(x; \epsilon)$ where $x < 0$ and $\phi = 1$, and $\bar{u}_2(x; \epsilon)$ where $x > 0$ and $\phi = 0$ (see Fig. 3 for an illustration). Clearly for all three DDMs, we have

$$\bar{u}_2 = g. \tag{12}$$

Note that there could be multiple layers near the boundary and hence multiple inner expansions may be required to match the two outer expansions.

2.2.1. Analysis of DDM2

We first present the matched asymptotic analysis of DDM2. In 1D, DDM2 is,

$$\phi \frac{d^2}{dx^2} u_\epsilon - \frac{1}{\epsilon^2}(1 - \phi)(u_\epsilon - g) = \phi f, \tag{13}$$

We assume the outer expansion $\bar{u}_1$ satisfies,

$$\bar{u}_1(x; \epsilon) = \bar{u}_1^{(0)}(x) + \epsilon \bar{u}_1^{(1)}(x) + ... \tag{14}$$

Plugging into Eq. (13) and assuming that neither f nor g depends on ϵ, we have,

$$\frac{d^2}{dx^2} \bar{u}_1^{(0)} = f, \tag{15}$$

$$\frac{d^2}{dx^2} \bar{u}_1^{(k)} = 0, k = 1, 2, ... \tag{16}$$

Now, if $\bar{u}_1^{(0)} = g$ on ∂D so that $\bar{u}_1^{(0)}$ is the unique solution to Eqs. (1) and (2), then DDM2 reduces to the Poisson equation with Dirichlet boundary condition at leading order. The L^2 convergence rate is then determined by the next leading order term, $\bar{u}_1^{(1)}$. In order to determine the boundary conditions for $\bar{u}_1^{(0)}$ and $\bar{u}_1^{(1)}$, we need to analyze the behavior of DDM2 (Eq. (13)) near ∂D using inner expansions and then match the inner and outer expansions in a region of overlap.

To get the inner equations, we first rewrite DDM2 by substituting ϕ with Eq. (6) and dividing by ϕ:

$$\frac{d^2}{dx^2} u_\epsilon - \frac{e^{6x/\epsilon}}{\epsilon^2}(u_\epsilon - g) = f. \tag{17}$$

12

$$\begin{array}{ccc} \text{Outer} & \text{Inner } K_1 & \text{Outer} \\[4pt] & \hat{u}_1 & \\ \phi = 1 & x \sim \mathcal{O}(\epsilon) & \phi = 0 \\ \bar{u}_1 & 1 - \phi \sim \mathcal{O}(1) & \bar{u}_2 = g \\[4pt] & z_1 = x/\epsilon & \\ & d/dx = \epsilon^{-1} d/dz_1 & \end{array}$$

Figure 3: Inner layer K_1

Consider Eq. (17) in a region K_1 (see Fig. 3), where $x \sim O(\epsilon)$ and $e^{6x/\epsilon} \sim O(1)$ and we introduce a stretched variable,

$$z_1 = \frac{x}{\epsilon}. \tag{18}$$

In a local coordinate system near ∂D, the derivatives become

$$\frac{d}{dx} = \frac{1}{\epsilon} \frac{d}{dz_1}, \tag{19}$$

$$\frac{d^2}{dx^2} = \frac{1}{\epsilon^2} \frac{d^2}{dz_1^2}, \tag{20}$$

The inner expansion associated with the stretched variable z_1 is

$$\hat{u}_1(z_1; \epsilon) = \hat{u}_1^{(0)}(z_1) + \epsilon \hat{u}_1^{(1)}(z_1) + \dots \tag{21}$$

To obtain the matching conditions for each outer solution $\bar{u}_i$ $(i = 1, 2)$, we assume that there is a region of overlap where both the inner and the outer expansions are valid. In this region, if we evaluate the outer expansion in the inner coordinates, this must match the limits of the inner solutions away from the interface, that is,

$$\hat{u}_1(z_1; \epsilon) \simeq \bar{u}_1(\epsilon z_1; \epsilon), \text{ as } z_1 \to -\infty \text{ and } \epsilon z_1 \to 0^-, \tag{22}$$

$$\hat{u}_1(z_1; \epsilon) \simeq \bar{u}_2(\epsilon z_1; \epsilon) = g, \text{ as } z_1 \to +\infty \text{ and } \epsilon z_1 \to 0^+. \tag{23}$$

Here a single inner expansion $\hat{u}_1$ is able to match both outer expansions up to $O(\epsilon)$ for DDM2 as shown below. As we see later in Sec. 2.2.2, this is not the case

13

for DDM1 and DDM3 and an additional layer needs to be introduced. Combining Eqs. (12), (14), (21)-(23), we have the following asymptotic matching conditions up to $O(\epsilon)$,

$$\hat{u}_1^{(0)}(z_1) = \bar{u}_1^{(0)}(0), \text{ as } z_1 \to -\infty, \tag{24}$$

$$\hat{u}_1^{(1)}(z_1) = \bar{u}_1^{(1)}(0) + z_1 \frac{d}{dx}\bar{u}_1^{(0)}(0), \text{ as } z_1 \to -\infty, \tag{25}$$

and

$$\hat{u}_1^{(0)}(z_1) = g, \text{ as } z_1 \to +\infty, \tag{26}$$

$$\hat{u}_1^{(1)}(z_1) = 0, \text{ as } z_1 \to +\infty. \tag{27}$$

Plugging Eq. (21) into Eq. (17) and using derivatives in the local coordinate system (Eqs. (19) and (20)), we obtain the following equations for the inner expansion $\hat{u}_1$:

At $O(\epsilon^{-2})$,
$$\frac{d^2}{dz_1^2}\hat{u}_1^{(0)} - e^{6z_1}(\hat{u}_1^{(0)} - g) = 0, \tag{28}$$

At $O(\epsilon^{-1})$,
$$\frac{d^2}{dz_1^2}\hat{u}_1^{(1)} - e^{6z_1}\hat{u}_1^{(1)} = 0. \tag{29}$$

Note that the solution to the following homogeneous ordinary differential equation,

$$y'' - e^{6x}y = 0, \tag{30}$$

is given by

$$y = C_1 I_0(\frac{e^{3x}}{3}) + C_2 K_0(\frac{e^{3x}}{3}), \tag{31}$$

where I_0 and K_0 are the modified Bessel functions of the first and second kind, respectively, and C_1 and C_2 are constants. Clearly, $\hat{u}_1^{(0)} = g$ is a solution to Eq. (28). Therefore,

$$\hat{u}_1^{(0)}(z_1) = g + C_1 I_0(\frac{e^{3z_1}}{3}) + C_2 K_0(\frac{e^{3z_1}}{3}), \tag{32}$$

$$\hat{u}_1^{(1)}(z_1) = C_3 I_0(\frac{e^{3z_1}}{3}) + C_4 K_0(\frac{e^{3z_1}}{3}), \tag{33}$$

14

where C_i's are constants. The modified Bessel functions satisfy:

$$\lim_{z_1 \to -\infty} I_0\left(\frac{e^{3z_1}}{3}\right) = 1, \tag{34}$$

$$\lim_{z_1 \to +\infty} I_0\left(\frac{e^{3z_1}}{3}\right) = +\infty, \tag{35}$$

$$\lim_{z_1 \to -\infty} K_0\left(\frac{e^{3z_1}}{3}\right) \sim -\ln\left(\frac{1}{2}\frac{e^{3z_1}}{3}\right) - \gamma \sim -3z_1 + \ln 6 - \gamma, \tag{36}$$

$$\lim_{z_1 \to +\infty} K_0\left(\frac{e^{3z_1}}{3}\right) = 0, \tag{37}$$

where $\gamma \approx 0.5772$ is the Euler–Mascheroni constant. From Eq. (36) and the matching condition (24) we conclude that $C_2 = 0$. Using Eq. (35) and the matching conditions as $z_1 \to +\infty$ (Eqs. (26) and (27)), we have $C_1 = C_3 = 0$. Putting everything together and using (25), we find

$$\hat{u}_1^{(0)} = g, \tag{38}$$

$$\hat{u}_1^{(1)} = -\frac{A}{3}K_0\left(\frac{e^{3z_1}}{3}\right) \sim Az_1 + \frac{A}{3}(-\ln 6 + \gamma), \text{ as } z_1 \to -\infty, \tag{39}$$

where $A = \frac{d}{dx}\bar{u}^{(0)}(0)$ is the derivative of the exact solution at the boundary. By the matching conditions (Eqs. (24) and (25)), the two leading terms of the outer solution $\bar{u}_1$ satisfies

$$\bar{u}_1^{(0)}(0) = g, \tag{40}$$

$$\bar{u}_1^{(1)}(0) = \frac{A}{3}(-\ln 6 + \gamma). \tag{41}$$

Therefore, $\bar{u}_1^{(0)} = u$, which is the exact solution of the Poisson equation with Dirichlet boundary condition (Eqs. (1) and (2)), and DDM2 is first order accurate in ϵ in the L^2 norm since $\bar{u}_1 - u \sim O(\epsilon)$ if $A \neq 0$. As for the L^∞ norm, we also need to consider the error at the boundary , e.g., $\hat{u}_1(0) - g \approx \epsilon\hat{u}_1^{(1)}(0) \sim O(\epsilon)$ for DDM2. Thus, DDM2 is first order accurate in ϵ in the L^∞ norm as well if $A \neq 0$. If $A = 0$, then the asymptotic analysis suggests better than first-order accuracy. In fact when $A = 0$, DDM2 is $O\left(\epsilon^2\right)$ accurate in both L^2 and L^∞ as shown in Appendix A.

To confirm our analysis, we compute $\bar{u}_1^{(1)}(0) \approx C(\epsilon)$, where $C(\epsilon) \approx (u_\epsilon - u)/\epsilon$ as $z_1 << 0$ is shown in Fig. 2(c). Using $A = 1.111$, both the asymptotic analysis

15

and the numerical results agree and give $\bar{u}_1^{(1)} \approx -0.450$ (see Tab. 19 in Appendix C for other cases).

2.2.2. Analysis of DDM1 and DDM3

We now extend the matched asymptotic analysis in Sec.2.2.1 to DDM1 and DDM3. In 1D, DDM1 reads,

$$\frac{d}{dx}(\phi\frac{d}{dx}u_\epsilon) - \frac{1}{\epsilon^3}(1-\phi)(u_\epsilon - g) = \phi f, \tag{42}$$

Substituting ϕ from Eq. (6), we have,

$$\frac{d^2}{dx^2}u_\epsilon - \frac{6}{\epsilon}\frac{1}{1+e^{-6x/\epsilon}}\frac{d}{dx}u_\epsilon - \frac{e^{6x/\epsilon}}{\epsilon^3}(u_\epsilon - g) = f. \tag{43}$$

We assume that

$$\bar{u}_1(x;\epsilon) = \bar{u}_1^{(0)}(x) + \epsilon\bar{u}_1^{(1)}(x;\ln\epsilon) + o(\epsilon). \tag{44}$$

Note that $\ln\epsilon$ is included in the outer expansion, which we will explain later. Plugging Eq. (44) into the DDM1 (Eq. (42)), again we have,

$$\frac{d^2}{dx^2}\bar{u}_1^{(0)} = f, \tag{45}$$

$$\frac{d^2}{dx^2}\bar{u}_1^{(k)} = 0, k = 1, 2, ... \tag{46}$$

As for inner expansion, we first consider DDM1 (Eq. (43)) in K_1, where $x \sim O(\epsilon)$ and $e^{6x/\epsilon} \sim O(1)$. We introduce the same stretched variable,

$$z_1 = \frac{x}{\epsilon}, \tag{47}$$

and the inner expansion,

$$\hat{u}_1(z_1;\epsilon) = \hat{u}_1^{(0)}(z_1) + \epsilon\hat{u}_1^{(1)}(z_1) + ..., \tag{48}$$

Outer	Inner K_2	Inner K_1	Outer

$$\phi = 1 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \phi = 0$$

$$\bar{u}_1 \xrightarrow{\;-\infty \leftarrow z_2\;} \hat{u}_2 \xrightarrow{\;z_2 \to +\infty\;} \qquad \hat{u}_1 \xrightarrow{\;z_1 \to +\infty\;} \bar{u}_2 = g$$

$$x \sim (\epsilon \ln \epsilon)/6 + \mathcal{O}(\epsilon) \qquad x \sim \mathcal{O}(\epsilon)$$

$$1 - \phi \sim \mathcal{O}(\epsilon) \qquad 1 - \phi \sim \mathcal{O}(1)$$

$$z_2 = x/\epsilon - \ln \epsilon/6 \qquad z_1 = x/\epsilon$$

Figure 4: Inner layers K_1 and K_2

Plugging into Eq. (43), we have the following equations,

At $O(\epsilon^{-3})$,
$$e^{6z_1}(\hat{u}_1^{(0)} - g) = 0, \tag{49}$$

At $O(\epsilon^{-2})$,
$$\frac{d^2}{dz_1^2}\hat{u}_1^{(0)} - \frac{6}{1 + e^{-6z_1}}\frac{d}{dz_1}\hat{u}_1^{(0)} - e^{6z_1}\hat{u}_1^{(1)} = 0, \tag{50}$$

At $O(\epsilon^{-1})$,
$$\frac{d^2}{dz_1^2}\hat{u}_1^{(1)} - \frac{6}{1 + e^{-6z_1}}\frac{d}{dz_1}\hat{u}_1^{(1)} - e^{6z_1}\hat{u}_1^{(2)} = 0, \tag{51}$$

At $= O(1)$,
$$\frac{d^2}{dz_1^2}\hat{u}_1^{(2)} - \frac{6}{1 + e^{-6z_1}}\frac{d}{dz_1}\hat{u}_1^{(2)} - e^{6z_1}\hat{u}_1^{(3)} = f, \tag{52}$$

$$\tag{53}$$

It follows that

$$\hat{u}_1^{(0)} = g, \tag{54}$$

$$\hat{u}_1^{(1)} = \hat{u}_1^{(2)} = 0, \tag{55}$$

$$\hat{u}_1^{(3)} = -fe^{-6z_1}. \tag{56}$$

Thus,

$$\hat{u}_1(z_1; \epsilon) = g - \epsilon^3 f e^{-6z_1} + o(\epsilon^3). \tag{57}$$

Clearly, $\hat{u}_1$ satisfies two of the matching conditions as $z_1 \to +\infty$ (Eqs. (26) and (27)). However, $\lim_{z_1 \to -\infty} \hat{u}_1(z_1; \epsilon) = -\infty$, which implies $\hat{u}_1$ does not satisfy the other two matching conditions (Eqs. (24) and (25)) and hence there exists another layer, namely K_2, and another inner solution $\hat{u}_2$ (see Fig. 4). Accordingly, we introduce the layer K_2 where $x \sim \alpha\epsilon \ln(\epsilon) + O(\epsilon)$. Here we choose $\alpha = 1/6$ and consider the two leading terms in the inner expansions, other choices of α

17

will result different layers, but their inner expansions are the same up to $O(\epsilon)$. We consider a new stretched variable,

$$z_2 = \frac{x - \epsilon(\ln \epsilon)/6}{\epsilon} = \frac{x}{\epsilon} - \frac{\ln \epsilon}{6} \tag{58}$$

Similar to Eq. (22), we derive the corresponding matching conditions in K_2, that is

$$\bar{u}_1(\epsilon z_2 + \frac{\epsilon \ln \epsilon}{6}) \simeq \hat{u}_2(z_2), \text{ as } z_2 \to -\infty \text{ and } \epsilon z_2 \to 0^-. \tag{59}$$

It follows that

$$\lim_{z_2 \to -\infty} \hat{u}_2^{(0)}(z_2) = \bar{u}_1^{(0)}(0), \tag{60}$$

$$\lim_{z_2 \to -\infty} \hat{u}_2^{(1)}(z_2) = \bar{u}_1^{(1)}(0) + (z_2 + \frac{\ln \epsilon}{6}) \frac{d}{dx} \bar{u}_1^{(0)}(0). \tag{61}$$

As $z_2 \to +\infty$, we need to match the inner solutions $\hat{u}_2(z_2)$ with $\hat{u}_1(z_1)$ from K_1. This is described below.

Plugging in the inner expansion into DDM1 (Eq. (43)) and using that $e^{6x/\epsilon} = \epsilon e^{6z_2}$, we obtain:

$$\text{At } O(\epsilon^{-2}), \qquad \frac{d^2}{dz_2^2} \hat{u}_2^{(0)} - e^{6z_2}(\hat{u}_2^{(0)} - g) = 0, \tag{62}$$

$$\text{At } O(\epsilon^{-1}), \qquad \frac{d^2}{dz_2^2} \hat{u}_2^{(1)} - 6e^{6z_2} \frac{d}{dz_2} \hat{u}_2^{(0)} - e^{6z_2} \hat{u}_2^{(1)} = 0. \tag{63}$$

The general solution to Eq. (62) is

$$\hat{u}_2^{(0)}(z_2) = g + C_1 I_0(\frac{e^{3z_2}}{3}) + C_2 K_0(\frac{e^{3z_2}}{3}), \tag{64}$$

where I_0 and K_0 are the modified Bessel functions of the first and second kind, respectively, and C_1, C_2 are constants.

Then we need to match $\hat{u}_2^{(0)}(z_2)$ with the leading order of the inner solution in K_1 (Eq. (54)) as $z_2 \to +\infty$, that is

$$\lim_{z_2 \to +\infty} \hat{u}_2^{(0)}(z_2) = g. \tag{65}$$

It follows that $C_1 = 0$. From Eq. (60) and (36), we obtain $C_2 = 0$. Thus,

$$\hat{u}_2^{(0)}(z_2) = g, \tag{66}$$

18

which implies

$$\bar{u}_1^{(0)}(0) = \lim_{z_2 \to -\infty} \hat{u}_2^{(0)}(z_2) = g. \tag{67}$$

Hence, DDM1 recovers the Poisson equation with Dirichlet boundary condition at leading order and $\bar{u}_1^{(0)}$ is the exact solution to Eqs. (1) and (2).

Plugging Eq. (66) into Eq. (63), we have

$$\hat{u}_2^{(1)}(z_2) = C_3 I_0(\frac{e^{3z_2}}{3}) + C_4 K_0(\frac{e^{3z_2}}{3}), \tag{68}$$

where C_3 and C_4 are constants. $\hat{u}_2^{(1)}$ needs to match $\hat{u}_1^{(1)} = 0$ in K_1 as $z_2 \to +\infty$, that is

$$\lim_{z_2 \to +\infty} \hat{u}_2^{(1)}(z_2) = 0, \tag{69}$$

Hence, $C_3 = 0$. On the other side where $z_2 \to -\infty$, by the matching condition Eq. (61), we obtain

$$\lim_{z_2 \to -\infty} \hat{u}_2^{(1)}(z_2) = \bar{u}_1^{(1)}(0) + A(z_2 + \frac{\ln \epsilon}{6}), \tag{70}$$

where $A = \frac{d}{dx}\bar{u}_1^{(0)}(0)$ is the derivative of exact solution at the boundary $x = 0$. Therefore, taking the limit of Eq. (68) as $z_2 \to -\infty$ and equating the result to Eq. (70), it follows that

$$\bar{u}_1^{(1)}(0) = -\frac{A}{6}\ln \epsilon + \frac{A}{3}(-\ln 6 + \gamma) \sim O(\ln \epsilon) \tag{71}$$

Hence, the convergence rate of DDM1 in the L^2 norm is $\epsilon \ln(\epsilon)$ if $A \neq 0$. Although the error at the boundary is $\hat{u}_1(0) - g \sim O(\epsilon^3)$ as seen from Eq. (57)), the L^∞ error is dominated by $\epsilon \ln(\epsilon)$.

As for DDM3, Replacing $\frac{1-\phi}{\epsilon}$ with $\frac{|\nabla \phi|}{\epsilon^2}$ and conducting an analogous analysis, we obtain

$$\bar{u}_1^{(0)}(0) = g, \tag{72}$$

$$\bar{u}_1^{(1)}(0) = -\frac{A}{6}\ln \epsilon + \frac{A}{3}(-\ln(\frac{1}{2}\sqrt{\frac{2}{3}}) + \gamma) \sim O(\ln \epsilon). \tag{73}$$

Thus, DDM3 is also of $O(\epsilon \ln \epsilon)$ in both L^2 and L^∞ norms if $A \neq 0$. If $A = 0$, our analysis suggests that DDM1 and DDM3 are better than first-order accurate. In fact, we find the schemes are $O\left(\epsilon^2 (\ln \epsilon)^2\right)$ accurate (see Appendix A).

19

To validate our analysis, we plot $C(\epsilon) \approx (u_\epsilon - u)/\epsilon$ versus $\ln(\epsilon)$ as in Fig. 2(b) and compute the slope through a linear fit using the numerical results in Sec. 2.1. For both DDM1 and DDM3, our asymptotic analysis suggests that the slope is $-A/6 \approx -0.185$ using $A = 1.111$. Numerically, we obtain the slopes -0.186 and -0.188 from DDM1 and DDM3, respectively, which agrees well with the theory (see Tab. 20 in Appendix C for other cases).

2.3. Development and analysis of higher-order, modified DDMs

In order to achieve higher order accuracy, we need to guarantee that the first order term in the outer expansion ($\bar{u}_1$) vanishes, that is, $\bar{u}_1^{(1)} = 0$. From the matching condition, Eq. (25), we observe that if $\hat{u}_1^{(1)}(z_1)$ behaves as Az_1 as $z_1 \to -\infty$, where A is the derivative of the exact solution at the boundary, then $\bar{u}_1^{(1)}(0)$ must be 0 and hence $\bar{u}_1^{(1)}(x) = 0$ since $d^2\bar{u}_1^{(1)}/dx^2 = 0$ from Eqs. (16) and (46). Therefore, if we can modify the original DDMs in a way such the inner solution $\hat{u}_1(z_1)$ satisfies,

$$\lim_{z_1 \to \pm\infty} \hat{u}_1^{(0)}(z_1) = g, \tag{74}$$

$$\lim_{z_1 \to +\infty} \hat{u}_1^{(1)}(z_1) = 0, \tag{75}$$

$$\lim_{z_1 \to -\infty} \hat{u}_1^{(1)}(z_1) \sim Az_1, \tag{76}$$

we should be able to achieve higher-order accuracy based on the asymptotic analysis. Accordingly, this suggests that to achieve higher-order accuracy we may modify the DDMs (referred to as the mDDMs) in the following way:

$$\textbf{mDDM1}: \quad \nabla \cdot (\phi \nabla u_\epsilon) - \frac{1}{\epsilon^3}(1 - \phi)(u_\epsilon - g - r\boldsymbol{n} \cdot \nabla u_\epsilon) = \phi f, \tag{77}$$

$$\textbf{mDDM2}: \quad \phi \Delta u_\epsilon - \frac{1}{\epsilon^2}(1 - \phi)(u_\epsilon - g - r\boldsymbol{n} \cdot \nabla u_\epsilon) = \phi f, \tag{78}$$

$$\textbf{mDDM3}: \quad \nabla \cdot (\phi \nabla u_\epsilon) - \frac{|\nabla \phi|}{\epsilon^2}(u_\epsilon - g - r\boldsymbol{n} \cdot \nabla u_\epsilon) = \phi f, \tag{79}$$

where $\boldsymbol{n} = -\frac{\nabla \phi}{|\nabla \phi|}$ is the normal vector and the new term $r\boldsymbol{n} \cdot \nabla u_\epsilon$ is designed to cancel the contribution of the derivative of the exact solution (e.g., the A terms) in the outer solutions $\bar{u}_1^{(1)}$. This is similar to, but simpler than, the active penalty method in [41], which introduces penalty terms to match the

20

solution and normal derivatives across the physical boundary. The modified boundary condition $g + r\mathbf{n} \cdot \nabla u_\epsilon$ is simply a linear approximation of the solution near the boundary in the normal direction to the boundary. Next, we analyze these methods in the following two sections to determine their actual order of accuracy.

2.3.1. Asymptotic analysis of mDDM1 and mDDM3

Similar to Sec. 2.2, we assume $x = 0$ is the boundary and $r = x$. Consider mDDM1 in 1D:

$$\frac{d}{dx}(\phi \frac{d}{dx} u_\epsilon) - \frac{1}{\epsilon^3}(1 - \phi)(u_\epsilon - g - x\frac{d}{dx} u_\epsilon) = \phi f, \qquad (80)$$

Substituting ϕ with Eq. (6), we obtain

$$\frac{d^2}{dx^2} u_\epsilon - \frac{6}{\epsilon} \frac{1}{1 + e^{-6x/\epsilon}} \frac{d}{dx} u_\epsilon - \frac{e^{6x/\epsilon}}{\epsilon^3}(u_\epsilon - g - x\frac{d}{dx} u_\epsilon) = f. \qquad (81)$$

After examining possible ways of performing matched asymptotic expansions for Eq. (81), the simplest approach is to choose a new inner variable $z_3 = \frac{x}{\epsilon^{1.5}}$ and the corresponding inner solution $\hat{u}_3(z_3)$ in the region K_3 (see Fig. 5), where $x \sim \epsilon^{1.5}$ and $e^{6x/\epsilon} \approx 1$. In the local coordinate system, the derivatives become

$$\frac{d}{dx} = \frac{1}{\epsilon^{1.5}} \frac{d}{dz_3}, \qquad (82)$$

$$\frac{d^2}{dx^2} = \frac{1}{\epsilon^3} \frac{d^2}{dz_3^2}, \qquad (83)$$

Similar to Eqs. (22) and (23), we develop matching conditions for K_3, that is,

$$\hat{u}_3(z_3; \epsilon) \simeq \bar{u}_1(\epsilon^{1.5} z_3; \epsilon), \text{ as } z_3 \to -\infty, \quad \text{and} \quad \epsilon^{1.5} z_3 \to 0^- \qquad (84)$$

$$\hat{u}_3(z_3; \epsilon) \simeq \bar{u}_2(\epsilon^{1.5} z_3; \epsilon) = g, \text{ as } z_3 \to +\infty, \quad \text{and} \quad \epsilon^{1.5} z_3 \to 0^+. \qquad (85)$$

Since we use the scale $\epsilon^{1.5}$, it is natural to include half powers in the outer and inner expansions:

$$\bar{u}_1(x; \epsilon) = \bar{u}_1^{(0)}(x) + \epsilon^{0.5} \bar{u}_1^{(0.5)}(x) + \epsilon \bar{u}_1^{(1)}(x) + \epsilon^{1.5} \bar{u}_1^{(1.5)}(x)..., \qquad (86)$$

$$\hat{u}_3(z_3; \epsilon) = \hat{u}_3^{(0)}(z_3) + \epsilon^{0.5} \hat{u}_3^{(0.5)}(z_3) + \epsilon \hat{u}_3^{(1)}(z_3) + \epsilon^{1.5} \hat{u}_3^{(1.5)}(z_3).... \qquad (87)$$

$$\phi = 1 \quad \vdots \quad \hat{u}_3$$

$$\vdots \quad x \sim \mathcal{O}(\epsilon^{1.5}) \quad \vdots \quad \phi = 0$$

$$\bar{u}_1 \quad \vdots \quad 1 - \phi \sim 0.5 \quad \vdots \quad \bar{u}_2 = g$$

$$\vdots \quad z_3 = x/\epsilon^{1.5} \quad \vdots$$

$$\vdots \, d/dx = \epsilon^{-1.5} d/dz_3 \,\vdots$$

Figure 5: Inner layer K_3

Plugging into Eqs. (84) and (85), we obtain the matching conditions,

$$\hat{u}_3^{(0)}(z_3) = g, \text{ as } z_3 \to +\infty, \tag{88}$$

$$\hat{u}_3^{(k)}(z_3) = 0, k = 0.5, 1, 1.5, ..., \text{ as } z_3 \to +\infty, \tag{89}$$

and

$$\hat{u}_3^{(k)}(z_3) = \bar{u}_1^{(k)}(0), k = 0, 0.5, 1, \text{ as } z_3 \to -\infty, \tag{90}$$

$$\hat{u}_3^{(1.5)}(z_3) = \bar{u}_1^{(1.5)}(0) + z_3 \frac{d}{dx} \bar{u}_1^{(0)}(0), \text{ as } z_3 \to -\infty. \tag{91}$$

Plugging Eq. (87) into Eq. (81) we derive the following inner equation at the leading order $O(\epsilon^{-3})$:

$$\frac{d^2}{dz_3^2} \hat{u}_3^{(0)} - (\hat{u}_3^{(0)} - g - z_3 \frac{d}{dz_3} \hat{u}_3^{(0)}) = 0, \tag{92}$$

Clearly, $\hat{u}_3^{(0)} = g$ is a solution to Eq. (92). Note that the general solution to the following homogeneous ordinary differential equation,

$$y'' + xy' - y = 0, \tag{93}$$

involves a linear combination of parabolic cylinder functions $(D_{-2}(x))$ [79]:

$$y = e^{-x^2/4}(C_1 D_{-2}(x) + C_2 D_{-2}(-x)), \tag{94}$$

where C_1 and C_2 are constants. Thus,

$$\hat{u}_3^{(0)} = g + e^{-z_3^2/4}(C_1 D_{-2}(z_3) + C_2 D_{-2}(-z_3)), \tag{95}$$

22

where $D_{-2}(z_3) \sim z_3^2 e^{-z_3^2/4}$ as $z_3 \to +\infty$ and $D_{-2}(z_3) = -\sqrt{2\pi} z_3 e^{z_3^2/4}$ as $z_3 \to -\infty$. By the matching conditions (Eqs. (88) and (90)), we have $C_1 = C_2 = 0$ and

$$\hat{u}_3^{(0)} = g, \tag{96}$$

$$\bar{u}_1^{(0)}(0) = g. \tag{97}$$

Hence mDDM1 recovers the Poisson equation at the leading order, e.g., $\bar{u}_1^{(0)} = u$, the exact solution.

At the next order $O(\epsilon^{-2.5})$ in Eq. (81), we have,

$$\frac{d^2}{dz_3^2} \hat{u}_3^{(0.5)} - (\hat{u}_3^{(0.5)} - z_3 \frac{d}{dz_3} \hat{u}_3^{(0.5)}) = 0. \tag{98}$$

Thus,

$$\hat{u}_3^{(0.5)} = e^{-z_3^2/4}(C_1 D_{-2}(z_3) + C_2 D_{-2}(-z_3)). \tag{99}$$

Again, using the matching conditions (Eqs. (89) and (90)), we derive $\hat{u}_3^{(0.5)} = 0$. A similar argument at $O(\epsilon^{-2})$ in Eq. (81) gives $\hat{u}_3^{(1)} = 0$.

At $O(\epsilon^{-1.5})$ in Eq. (81), we obtain,

$$\frac{d^2}{dz_3^2} \hat{u}_3^{(1.5)} - (\hat{u}_3^{(1.5)} - z_3 \frac{d}{dz_3} \hat{u}_3^{(1.5)}) = 0. \tag{100}$$

Hence,

$$\hat{u}_3^{(1.5)} = e^{-z_3^2/4}(C_1 D_{-2}(z_3) + C_2 D_{-2}(-z_3)). \tag{101}$$

Applying the matching conditions (Eqs. (89) and (91)), we get

$$\hat{u}_3^{(1.5)} = -\frac{A e^{-z_3^2/4}}{\sqrt{2\pi}} D_{-2}(z_3) \sim A z_3 \text{ as } z_3 \to -\infty, \tag{102}$$

where $A = \frac{d}{dx} \bar{u}_1^{(0)}(0)$ is the derivative of the exact solution at the boundary. Plugging into Eq. (91), we conclude

$$\bar{u}_1^{(1.5)}(0) = 0. \tag{103}$$

Thus, the asymptotic analysis suggests that $\bar{u}_1(x) = \bar{u}_1^{(0)}(x) + O(\epsilon^2)$ and mDDM1 is 2nd order accurate in ϵ in the L^2 norm as desired.

23

To analyze the error in the L^∞ norm, we need to consider the error at the boundary. At the boundary, $\hat{u}_3(0) - g \sim \epsilon^{1.5}\hat{u}_3^{(1.5)}(0) = -\frac{A\epsilon^{1.5}}{\sqrt{2\pi}}$. This suggests that mDDM1 is 1.5 order accurate in ϵ in the L^∞ norm.

This analysis can be easily extended to higher dimensions and to mDDM3 to obtain the same conclusions. The only difference with respect to mDDM3 is that $\frac{1-\phi}{\epsilon^3}$ is replaced with $\frac{|\nabla\phi|}{\epsilon^2}$, which gives $\hat{u}_3(0) - g \sim \epsilon^{1.5}\hat{u}_3^{(1.5)}(0) = -\frac{A\epsilon^{1.5}}{\sqrt{6\pi}}$. The errors in the L^2 norm and L^∞ norm are still suggested to be 2nd and 1.5 orders, respectively.

To confirm our analysis, we compute $\hat{u}_3^{(1.5)}(0)$ numerically in the following way. We plot the values of $(u_\epsilon - u)/\epsilon^{1.5}$ at the boundary ($z_3 = 0$) versus $\sqrt{\epsilon}$, where u_ϵ is the numerical solution, and we find the y-intercept of a quadratic fit. Since $A = 1.111$, our asymptotic analysis gives that the values of $\hat{u}_3^{(1.5)}(0)$ are -0.443 and -0.256 for mDDM1 and mDDM3, respectively. We obtain -0.442 and -0.257 from the numerical results of mDDM1 and mDDM3, respectively, which is consistent with our theory (see Tab. 21 in Appendix C for other cases).

2.3.2. Asymptotic analysis of mDDM2

Even though mDDM2 is obtained using the same modification as in mDDM1 and mDDM3, surprisingly as our analysis below suggests and numerical results confirm (see Sec. 2.4), mDDM2 is only first-order accurate in both L^2 and L^∞. In 1D, the mDDM2 is:

$$\phi\frac{d^2}{dx^2}u_\epsilon - \frac{1}{\epsilon^2}(1-\phi)(u_\epsilon - g - x\frac{d}{dx}u_\epsilon) = \phi f. \tag{104}$$

Substituting ϕ from Eq. (6), we obtain

$$\frac{d^2}{dx^2}u_\epsilon - \frac{e^{6x/\epsilon}}{\epsilon^2}(u_\epsilon - g - x\frac{d}{dx}u_\epsilon) = f. \tag{105}$$

Considering Eq. (105) in K_1, we use the inner variable $z_1 = \frac{x}{\epsilon}$ to derive the following inner equations for $\hat{u}_1$:

$$\text{At } O(\epsilon^{-2}), \qquad \frac{d^2}{dz_1^2}\hat{u}_1^{(0)} - e^{6z_1}(\hat{u}_1^{(0)} - g - z_1\frac{d}{dz_1}\hat{u}_1^{(0)}) = 0, \tag{106}$$

$$\text{At } O(\epsilon^{-1}), \qquad \frac{d^2}{dz_1^2}\hat{u}_1^{(1)} - e^{6z_1}(\hat{u}_1^{(1)} - z_1\frac{d}{dz_1}\hat{u}_1^{(1)}) = 0. \tag{107}$$

24

Clearly, $\hat{u}_1^{(0)} = g$ is a solution to Eq. (106), and the general solution to the homogeneous ordinary differential equation below,

$$y'' - e^{6x}(y - xy') = 0, \tag{108}$$

is given by,

$$y = C_1 x + C_2(-e^{e^{6x}(1-6x)/36} - x \int_0^x h(t)dt), \tag{109}$$

where $h(x) = e^{e^{6x}(1-6x)/36+6x}$ (see Appendix D for details). It follows that

$$\hat{u}_1^0(z_1) = g + C_1 z_1 + C_2(-e^{e^{6z_1}(1-6z_1)/36} - z_1 \int_0^{z_1} h(t)dt), \tag{110}$$

$$\hat{u}_1^1(z_1) = C_3 z_1 + C_4(-e^{e^{6z_1}(1-6z_1)/36} - z_1 \int_0^{z_1} h(t)dt), \tag{111}$$

where the C_i are constants and

$$\lim_{z_1 \to -\infty} e^{e^{6z_1}(1-6z_1)/36} = 1, \tag{112}$$

$$\lim_{z_1 \to +\infty} e^{e^{6z_1}(1-6z_1)/36} = 0, \tag{113}$$

$$\lim_{z_1 \to -\infty} \int_0^{z_1} h(t)dt \approx 0.17, \tag{114}$$

$$\lim_{z_1 \to +\infty} \int_0^{z_1} h(t)dt \approx 2.75. \tag{115}$$

Combining with the matching conditions as $z_1 \to \pm\infty$ (Eqs. (24)-(27)) and solving for the constants C_i, we obtain that

$$\hat{u}_1^{(0)}(z_1) = g, \tag{116}$$

$$\hat{u}_1^{(1)}(z_1) \approx \frac{A}{2.92}(-e^{e^{6z_1}(1-6z_1)/36} + z_1 \int_{z_1}^{\infty} h(t)dt), \tag{117}$$

where as before $A = \frac{d}{dx}\bar{u}_1^{(0)}(0)$. Therefore,

$$\lim_{z_1 \to -\infty} \hat{u}^{(1)}(z_1) \sim A z_1 - \frac{A}{2.92}, \tag{118}$$

which implies

$$\bar{u}_1^{(1)}(0) \approx -\frac{A}{2.92} \neq 0, \text{ if } A \neq 0. \tag{119}$$

25

This suggests mDDM2 is 1st order accurate in ϵ in the L^2 norm when $A \neq 0$. Plugging $z_1 = 0$ into Eq. (117) and we have

$$\hat{u}_1^{(1)}(0) \approx -\frac{Ae^{1/36}}{2.92}, \qquad (120)$$

which implies that mDDM2 is also 1st order accurate in the L^∞ norm.

To confirm our analysis, we compute $\bar{u}_1^{(1)}$ numerically, which is given by $(u_\epsilon - u)/\epsilon$ as $z << 0$, using the numerical solution u_ϵ in Sec. 2.4. In addition, we interpolate and plot the boundary values of $(u_\epsilon - u)/\epsilon$ at $z_1 = 0$ versus ϵ, and then calculate $\hat{u}^{(1)}(0)$ from the y-intercept of a linear fit. Since $A = 1.111$, our analysis indicates that $\bar{u}_1^{(1)} \approx -\frac{A}{2.92} \approx -0.381$ and $\hat{u}^{(1)}(0) \approx -\frac{Ae^{1/36}}{2.92} \approx -0.391$. The numerical results yield $\bar{u}_1^{(1)} \approx -0.380$ and $\hat{u}^{(1)}(0) \approx -0.391$, which agrees very well with the theory (see Tab. 22 in Appendix C for other cases).

2.4. Numerical results for mDDM1-3

We adopt the same problem setup and an analogous discretization as in Sec. 2.1. The results using $h = \epsilon/4$ are presented in Fig. 1(b). Clearly mDDM3 and mDDM1 are more accurate than DDM1-3 and mDDM2. Further, mDDM2 and DDM2 (Fig. 1(a)) display similar levels of accuracy as suggested by our analysis.

To test the order of accuracy in ϵ predicted by theory, we next take $h = \epsilon^{1.5}/4$ in order to resolve the inner layer K_3, as guided by our asymptotic analysis in Sec. 2.3.1. Later, in Secs. 3 and 4, we will test convergence with $\epsilon \propto h$. In Tabs. 3 and 4, we present convergence results for mDDM1-3 for case 1 (see also Tabs. 13-18 in Appendix B for other cases). Consistent with our analysis, we observe that mDDM1 and mDDM3 are approximately 2nd order accurate in the L^2 norm but approximately 1.5 order accurate in the L^∞ norm. Also as predicted, mDDM2 is only 1st order accurate in both norms. Therefore, in practice we recommend using mDDM1 or mDDM3.

However, taking $h \sim \epsilon^{1.5}$ is often too constraining for solving problems in 2D and 3D. Instead, one may choose $h = \epsilon/c$. Doing this, one can obtain results

with orders of accuracy ranging from 1.5 to 2 (1.5 for large c and approximately 2 for $c \approx 1$; see Tabs. 5 and 6 in Sec. 3 and Tabs. 25, 26 and 28 in Appendices).

ϵ	mDDM1				mDDM3			
	$E^{(2)}$	k	$E^{(\infty)}$	k	$E^{(2)}$	k	$E^{(\infty)}$	k
2.00E-01	4.88E-01	0.00	2.10E-01	0.00	1.55E-01	0.00	8.23E-02	0.00
1.00E-01	1.50E-01	1.70	6.59E-02	1.67	3.77E-02	2.04	2.23E-02	1.88
5.00E-02	4.46E-02	1.75	1.98E-02	1.74	8.54E-03	2.14	5.96E-03	1.91
2.50E-02	1.26E-02	1.83	5.58E-03	1.83	1.89E-03	2.17	2.03E-03	1.55
1.25E-02	3.34E-03	1.91	1.55E-03	1.85	4.16E-04	2.19	6.15E-04	1.72
6.25E-03	8.61E-04	1.95	4.91E-04	1.65	8.74E-05	2.25	2.29E-04	1.42

Table 3: The L^2 and L^∞ errors for mDDM1 and mDDM3 with $h = \epsilon^{1.5}/4$.

ϵ	mDDM2			
	$E^{(2)}$	k	$E^{(\infty)}$	k
2.00E-01	4.58E-01	0.00	2.19E-01	0.00
1.00E-01	2.24E-01	1.03	1.06E-01	1.05
5.00E-02	1.11E-01	1.02	5.19E-02	1.03
2.50E-02	5.47E-02	1.02	2.59E-02	1.00
1.25E-02	2.72E-02	1.01	1.29E-02	1.01
6.25E-03	1.36E-02	1.00	6.45E-03	1.00

Table 4: The L^2 and L^∞ errors for mDDM2 with $h = \epsilon^{1.5}/4$.

3. Time-dependent problems

We next extend the mDDMs to simulate time-dependent PDEs in a moving domain $D(t)$ with Dirichlet boundary conditions. As we discuss later, our approach and analysis holds for much more general time-dependent equations, but we consider the diffusion equation here for simplicity of presentation:

$$\partial_t u = \Delta u + f \text{ in } D(t), \tag{121}$$

$$u = g \text{ on } \partial D(t), \tag{122}$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \text{ in } D(t). \tag{123}$$

27

3.1. Derivation and analysis of high-order, modified DDMs for time-dependent PDEs

To approximate Eqs. (121) and (122), we formulate the diffuse domain model using an approximation (mDDMt3), which is analogous to mDDM3 in Sec. 2.3.1:

mDDMt3: $\quad \partial_t(\phi u_\epsilon) = \nabla(\phi \nabla u_\epsilon) - \dfrac{1}{\epsilon^2}|\nabla\phi|(u_\epsilon - g - r\boldsymbol{n} \cdot \nabla u_\epsilon) + \phi f \quad$ (124)

where as before r is the signed distance function, with $r < 0$ denoting the interior of D. Since the domain may be time-dependent, ϕ may depend on time and thus ϕ needs to be in the time derivative [46]. Note that in time-dependent problems, the initial condition needs to be extended outside D as well although this extension need not be constant in the normal direction.

The asymptotic analysis for the time-dependent problem is very similar to that for the Poisson equation presented in Sec. 2.3.1. The only difference is that now the time derivative must be analyzed as well. Assuming the inner and outer expansions and matching conditions hold as in Sec. 2.3.1, from the leading order outer equation, we find that $\bar{u}_1^{(0)}$ satisfies Eq. (121).

In the inner expansion, taking $z_3 = x/\epsilon^{1.5}$, the time derivative can be written as:

$$\partial_t = -\frac{v}{\epsilon^{1.5}}\partial_{z_3} + o(1) \tag{125}$$

where v is the normal velocity of the domain boundary. Since ∂_t is of order $\epsilon^{-1.5}$, the time derivative will not affect the higher-order terms in the inner expansions of mDDM3. Therefore, we can still derive $\hat{u}_3^{(0)} = g$ and $\hat{u}_3^{(0.5)} = \hat{u}_3^{(1)} = 0$. Plugging into the inner equation at the next leading order $(O(\epsilon^{-1.5}))$, we obtain the same solution for $\hat{u}_3^{(1.5)}$ as that for the time-independent equation. Thus, mDDMt3 should be $O(\epsilon^2)$ accurate in L^2 and $O(\epsilon^{1.5})$ accurate in L^∞ for the time-dependent (diffusion) equation. Analogously, mDDM1 has the same order of accuracy as mDDM3 and mDDM2 is $O(\epsilon)$ accurate in both L^2 and L^∞ (results not shown).

3.2. Numerical results using mDDMt3 for time-dependent problems

For time-dependent problems, we first assume the domain does not change in time. Then, taking $f = \cos(x)\cos(t) + \cos(x)\sin(t)$ and $g = \cos(1.111)\sin(t)$, we obtain the exact solution $u = \cos(x)\sin(t)$ on the domain D=[-1.111,1.111]. Therefore we take the initial $u_\epsilon(x,0) = 0$ and the constant normal extension of f is obtained analytically. We use central difference discretizations in space (as in Sec. 2.1) and the Crank-Nicholson method discretization in time to solve Eq. (124) on the larger domain $\Omega = [-2, 2]$ with ϕ from Eq. (6). We still use the Thomas algorithm to solve the tridiagonal matrix system. We calculate the errors at $t = 1$ in the L^2 and L^∞ norms by setting $dt = h$ where dt and h are the time step and grid size, respectively. Here, instead of taking $h \propto \epsilon^{1.5}$ as in the previous section, we instead take $h = \epsilon/c$ where c is a constant.

Various choices of c have been tested and here we present results using $c = 4, 16, 128$ in Tabs. 5, 6. We observe that mDDMt3 is 2nd order accurate in the L^2 norm for all choices of c. However, in L^∞, we find that when $c = 4$, the schemes are roughly 2nd order accurate in ϵ while if $c = 128$ the order of accuracy decreases to 1.5, consistent with our theory. In Fig. 6, we show the numerical solutions of mDDMt3 near the boundary $x = 1.111$ for difference choices of c. When $c = 4$, the boundary layer K_3 is not resolved and hence the numerical solution is less smooth near the boundary than is the continuous solution of mDDMt3. Nevertheless, the solution with $c = 4$ is actually closer to the exact solution close to the boundary of D. This occurs because the discretization error, $u_{h,\epsilon} - u_\epsilon$, and analytic error, $u_\epsilon - u$, have opposite signs but similar magnitudes so that they can at least partially cancel one another in the total error $u_{h,\epsilon} - u = (u_{h,\epsilon} - u_\epsilon) + (u_\epsilon - u)$, yielding higher order accuracy than might be expected. As h decreases, the total error eventually becomes dominated by the analytic error, which results in errors that scale like $O(\epsilon^{1.5})$ in L^∞ as predicted by theory. See App. E for details.

Outside of D, the mDDMt3 solutions tend to the extension of the exact solution, that is constant in the normal direction, because the boundary condition at $\partial\Omega$ is equal to the boundary condition at ∂D. Other choices of boundary

conditions at $\partial\Omega$ would yield similar errors near ∂D but different behavior far from D (results not shown).

ϵ	c=4	k	c=16	k	c=128	k
0.2	2.85E-02	0.00	2.88E-02	0.00	2.88E-02	0.00
0.1	7.19E-03	1.99	7.39E-03	1.96	7.40E-03	1.96
0.05	1.68E-03	2.10	1.76E-03	2.07	1.76E-03	2.07
0.025	4.08E-04	2.04	4.19E-04	2.07	4.20E-04	2.07
0.0125	9.71E-05	2.07	1.03E-04	2.02	1.03E-04	2.03
0.00625	2.48E-05	1.97	2.61E-05	1.98	2.62E-05	1.98

Table 5: The L^2 errors for simulating the time-dependent (diffusion) equation using mDDMt3 with $h = \epsilon/c$ on a fixed domain.

ϵ	c=4	k	c=16	k	c=128	k
0.2	3.87E-02	0.00	3.85E-02	0.00	4.01E-02	0.00
0.1	1.00E-02	1.95	1.10E-02	1.81	1.18E-02	1.76
0.05	2.15E-03	2.22	3.17E-03	1.79	3.56E-03	1.73
0.025	5.00E-04	2.10	1.08E-03	1.55	1.10E-03	1.69
0.0125	1.59E-04	1.65	3.27E-04	1.72	3.41E-04	1.69
0.00625	4.38E-05	1.86	8.86E-05	1.88	1.15E-04	1.57

Table 6: The L^∞ errors for simulating the time-dependent (diffusion) equation using mDDMt3 with $h = \epsilon/c$ on a fixed domain.

Figure 6: Numerical solutions of mDDMt3 near the boundary $x = 1.111$ on the 1D stationary domain with different c. The green dashed line denotes the extension (e.g., constant in the normal direction) of the exact solution out of the domain. The black dashed line marks the right boundary $x = 1.111$.

Next we assume the boundary of the domain is also moving and set $x_l(t) = -1.111$, $x_r(t) = 1.111 + 0.5t$, where $x_l(t)$ and $x_r(t)$ represent the left and right hand sides of the domain D, respectively. We take $f = \cos(x)\cos(t) + \cos(x)\sin(t)$, $g = \cos(x_*)\sin(t)$, where $x_* = x_l, x_r$, and the exact solution is $u = \cos(x)\sin(t)$.

In general, neither the signed distance function r nor the function ϕ are given analytically. Because we need to use the signed distance function in the modified DDMs, we find it convenient to apply the level-set method [76, 77] to determine both r and ϕ. Therefore, we solve the following Hamilton-Jacobi equation:

$$\partial_t r + v|\nabla r| = 0, \tag{126}$$

using a 5th order upwind WENO scheme [80] and a 2nd order Total Variation Diminishing (TVD) Runge-Kutta (RK) method time discretization to obtain an

31

accurate fully discrete solution [81]. In order to keep $|\nabla r| = 1$, we periodically perform reinitialization [82, 83] by solving the following equation,

$$\partial_\tau r = \mathrm{sgn}_h(r^0)(1 - |\nabla r|), \tag{127}$$

where τ is pseudo time and $\mathrm{sgn}_h(r^0)$ is a smoothed approximation function of the sign of the initial signed distance function [84]:

$$\mathrm{sgn}_h(r) = \begin{cases} -1 & r < -h \\ 1 & r > h \\ \frac{r}{h} + \frac{1}{\pi}\sin(\frac{\pi r}{h}) & \text{else} \end{cases} \tag{128}$$

To determine whether reinitialization is needed or not, we calculate the slope of r near the boundaries of $D(t)$. Then we check if the maximum difference between the absolute value of the slope and 1 exceeds a threshold, here taken to be 0.01. If so, then we perform reinitialization.

Once the signed distance function $r(x, t)$ is obtained, we construct the phase field function ϕ through Eq. (6). Next, we solve mDDMt3 using the same numerical setup as before. The results are very similar to the stationary domain case. We again find that mDDMt3 is 2nd order accurate in the L^2 norm and between 1.5 and 2nd order accurate in the L^∞ norm, which is consistent with our analysis. See Tabs. 25 and 26 in App. F.

4. 2D numerical results

Now we consider the 2D diffusion equation Eqs. (121) and (122) on a moving 2D domain $D(t)$. The initial domain is enclosed by the polar curve (shown in Fig. 8(a)):

$$r(\theta) = 1 + 0.1\cos(3\theta) + 0.02\cos(5\theta). \tag{129}$$

We suppose the velocity of the domain $\boldsymbol{v}(\boldsymbol{x}, t)$ is given by,

$$\boldsymbol{v} = (0.2(1 + 2t)\cos(3\theta) + 0.12(1 + 6t)\cos(5\theta))(\cos(\theta), \sin(\theta)), \tag{130}$$

and solve for the signed distance function $r(x, t)$ using the level-set method as described in Sec. 3.2. The phase field function ϕ is determined from the signed

32

distance function r via Eq. (6). We choose f and g such that the exact solution is $u(r, \theta, t) = \frac{1}{4}r^2$. Accordingly, we take the initial condition $u_\epsilon(\mathbf{x}, 0) = \frac{1}{4}r^2$. The extensions of f, g off D that are constant in the normal direction are accomplished by solving [84]

$$\psi_t + \text{sgn}_h(r)\frac{\nabla r}{|\nabla r|} \cdot \nabla \psi = 0, \tag{131}$$

in a region near ∂D_t with width of $O(\epsilon)$, where $\psi = f$ or g and sgn_h is given by Eq. (128). Because the Hamilton-Jacobi equations (126)-(127) for the signed distance function r and Eq. (131) for the extensions of f and g need to be solved only in a narrow band around ∂D and explicit time stepping methods are used, the costs associated with solving these equations are negligible compared to those associated with solving the mDDM equations.

We solve mDDMt3 (Eq. (124)) on the larger domain $\Omega = [-2, 2] \times [-2, 2]$ using the Crank-Nicholson method and the standard second-order central difference scheme together with adaptive, block-structured mesh refinement as in [85]. The implicit equations are solved using a mass-conserving multigrid solver [70]. Empirically, we found the number of iterations of the multigrid solver is insensitive to the choice of the minimum grid size h_{fine} and only weakly dependent on ϵ. See Tab. 27 in App. G. The L^2 and L^∞ norms of the errors are calculated using 2D extensions of Eqs. (8) and (10). We use an adaptive mesh with a maximum of three levels of refinement. The mesh is refined according to the undivided gradient of $(u_\epsilon \phi)$, e.g., if $|\nabla (u_\epsilon \phi)| \geq 10^{-4}/(2h)$, where h is the local grid size, then the mesh is targeted for block-structured refinement (see [85] for details). We start with $\epsilon = 0.2$ and the coarse mesh grid size $h_{coarse} = 4/16$, which gives the fine mesh size $h_{fine} = 4/128 = \epsilon/6.4$, and the time step $dt = 0.1/128$. Then, the parameters ϵ, h_{coarse} and dt are refined at the same time while the number of levels and $h_{fine} = \epsilon/6.4$ are fixed.

Note that for the 2D and 3D numerical numerical implementations, we need to discretize the normal vector $\boldsymbol{n}$ in the mDDMs in the following way in order for the solvers to converge:

33

$$n = \begin{cases} -\frac{\nabla \phi}{|\nabla \phi|}, & \text{if } r \leq 0, \\ 0 & \text{else.} \end{cases}$$

We believe this is related to the smoothing properties of the relaxation step, but a full analysis is beyond the scope of this paper and will be considered in future work. Effectively this means that the modification in the mDDMs is performed in a region near the boundary but within the physical domain. The solution u_ϵ in the evolving domain is shown at $t = 0.1$ using $\epsilon = 0.025$ in Fig. 7(a). In Figs. 7(b) and 7(c), the L^2 and L^∞ errors are shown at this time (see also Tab. 28 in App. G). The results suggest mDDMt3 is 2nd order accurate in h, ϵ and dt in L^2 and between 1.5 and 2nd order in L^∞, consistent with our theory and numerical results from the previous section.

In Fig. 8, we present a long time simulation of the dynamics using mD-DMt3 (Eq. (124)) using the domain $\Omega = [-4, 4] \times [-4, 4]$ with $\epsilon = 0.025$, $h_{coarse} = 8/64$, three levels of mesh refinement and $dt = 0.1/512$. Here the computational domain is larger than that used for the convergence test presented earlier ($[-2, 2] \times [-2, 2]$) to accommodate the growth of the physical domain D (see Fig. 8). To improve efficiency, we could dynamically increased the size of Ω to accomodate the growth of D. In general, the buffer between ∂D and $\partial \Omega$ should be larger than $\approx 10\epsilon$. We present the solution restricted to the moving domain $D(t)$ at different times, up to t=1.9. The pointwise error at $t = 1.9$ is shown in Fig. 8(f) and the largest error, which is of order 10^{-3}, occurs near the right most finger. As can be seen from these figures, mDDMt3 is able to accurately simulate solutions on time-dependent, highly complex non-rectangular domains.

(a)

(b)

(c)

Figure 7: Error analysis of the 2D diffusion equation using mDDMt3 on the moving domain $D(t)$, see text for details. The green curve denotes the boundary ∂D. (a): The solution at $t = 0.1$ restricted on $D(t)$ with the boundary contour (green line), (b): The L^2 error at t=0.1, (c): The L^∞ error at t=0.1.

Figure 8: Solution of mDDMt3 from Eq. (124) at different times. The solution is restricted to $D(t)$ whose boundary ∂D is denoted by the green curve. See text for details. (a): t=0, (b): t=0.5, (c): t=1.0, (d): t=1.5, (e): t=1.9. (f): The point-wise error at t=1.9 together with the block-structured adaptive mesh.

36

5. 3D numerical results

5.1. 3D Poisson test problem

We next consider a 3D Poisson equation on a stationary complex domain D:

$$-\boldsymbol{\nabla} \cdot (\beta(\boldsymbol{x})\boldsymbol{\nabla}u(\boldsymbol{x})) = f(\boldsymbol{x}), \qquad \boldsymbol{x} \in D, \tag{132}$$

$$u = p(\boldsymbol{x}). \qquad \boldsymbol{x} \in \partial D. \tag{133}$$

Following [40], in which this equation was used as as test problem for the virtual node method, we take $\beta(x, y, z) = 7 + x + 2y + 3z$ and set f and p such that the exact solution is $u(x, y, z) = xe^y + e^z\sqrt{1 + y^2}$. The domain D is bounded by a torus centered at $(0, 0, 0)$ with major radius $R_a = 0.6$, minor radius $R_b = 0.3$, and axis along $(0, -\sin(0.75), \cos(0.75))$. See Fig. 9(left), which also shows the block-structured adaptive mesh we used.

To approximate Eqs.(132) and (133), we use a version of mDDM3:

$$-\boldsymbol{\nabla} \cdot (\phi\beta\boldsymbol{\nabla}u_\epsilon) + \frac{|\boldsymbol{\nabla}\phi|}{\epsilon^2}(u_\epsilon - g - r\boldsymbol{n} \cdot \boldsymbol{\nabla}u_\epsilon) = \phi f. \tag{134}$$

The signed distance function r is taken to be:

$$r = \sqrt{\left(\left(R_a - \sqrt{r_x{}^2 + r_y{}^2}\right)^2 + r_z{}^2\right)} - R_b, \tag{135}$$

where $r_x = x$, $r_y = y\sin(\theta) + z\cos(\theta)$, $r_z = y\cos(\theta) - z\sin(\theta)$, and $\theta = 135^o$. The phase field function ϕ is determined from the signed distance function r via Eq. (6).

We solve Eq. (134) on the larger domain $\Omega = [-1, 1]^3$ using the standard second-order difference scheme together with adaptive, block-structured mesh refinement as in Sec. 4. We start with $\epsilon = 0.2$ and a uniform grid with mesh size $h = 2/16$. We test convergence by decreasing ϵ by factors of two and concommittantly increasing the number of levels of refinement by one such that the finest mesh size $h_{fine} = \epsilon/3.2$ is fixed. Fig. 9(left) shows the torus surface (in grey) with an adaptive mesh of 2 refinement levels, which corresponds to

37

$\epsilon = 0.05$. Note that the refined mesh is concentrated near ∂D. The mDDM3 solution u_ϵ on ∂D is shown in Fig. 9(right).

The L^∞ errors, compared with those using the second order virtual node method [40], which like other specialized, higher-order embedded boundary methods is more difficult to implement than mDDM3, are shown in Fig. 10. The grid size N for mDDM3 is obtained by assuming that we are using an uniform grid N^3 so that $h_{fine} = 2/N$. The results suggest that the rate of convergence in N of mDDM3 is similar to that of the virtual node method (about 1.8 order), although the absolute errors in mDDM3 are larger by about an order of magnitude. The reasons for this discrepancy are not entirely clear but are likely due to a combination of factors including the choices of the relation between h_{fine} and ϵ and the scaling of the $|\nabla \phi|$ term in the boundary condition term Eq. (133). In particular, by varying these choices, there are many ways the error could be optimized. For example, for each h_{fine}, we could find an optimal choice of ϵ that minimizes the error. While we tested a few combinations that could reduce the error, the optimal relation between h_{fine} and ϵ to minimize the error is not clear. However, taking $\epsilon \propto h_{fine}$ as we did robustly yields nearly 2nd order convergence rates. In addition, following [49], one could introduce an additional parameter α in the boundary condition term in Eq. (133), e.g., replace $|\nabla \phi|$ with $\alpha |\nabla \phi|$, and try to minimize the error through judicious choices of α. Other aspects of the solver (e.g., distances over which extensions are performed, adaptive mesh parameters, etc.) could also be optimized. These are subjects for future work.

Figure 9: A 3D example of the Poisson equation in a complex domain D using mDDM3. (left): The boundary of D is the toroidal surface (in grey), which is shown together with a 2-level adaptive mesh that corresponds to the interface thickness $\epsilon = 0.05$, (right): the mDDM3 solution u_ϵ at the boundary ∂D. See text for details.

Figure 10: L^∞ errors in solving the 3D Poisson equation on the toroidal domain D from Fig. 9. The open circles correspond to results from [40] using the virtual node method while the diamonds correspond to results using mDDM3. N is the number of grid points in one dimension for the virtual node method, which uses a uniform mesh, while $N = 2/h_{fine}$ for mDDM3 in which an adaptive mesh is used. See text for details.

6. Simulation of drug perfusion in the human brain

As a final example, we simulate the perfusion of a drug in a 3D model of a human brain. In particular, we used an anatomical template of the human brain from the ICBM atlas [86, 87, 88, 89]. The atlas provides smooth, probabilistic segmentations for the anatomical structures in the brain.This data is given on a Cartesian 256^3 voxel grid Ω_h where the volume of each voxel is 1 mm^3. Each voxel in this grid can contain white matter (WM), grey matter (GM), cerebral spinal fluid (CSF) or no brain matter (empty space).

We thresholded the CSF such that in the interior brain voxels contain only CSF or solid tissue (GM/WM) and only those voxels near CSF/solid tissue boundaries contain a combination of CSF and solid tissue. The corresponding volume fractions of the WM and GM are ϕ_W and ϕ_G, which are smooth functions. The CSF region is inferred as the region where $\phi_W + \phi_G < 0.5$ in the brain interior. There is also a thin rim of CSF at the outer boundary of the brain. See Fig. 11 (leftmost column) where the regions from lightest to darkest denote the WM (where $\phi_W \geq 0.5$), the GM (where $\phi_G \geq 0.5$) and CSF. Denote these regions by D_W, D_G, and D_{CSF}.

Let u_W and u_G be the drug concentrations in D_W and D_G, respectively. As a simple model, we assume that the drug is delivered through the CSF, which is modeled as a Dirichlet boundary condition for the drug concentration at the boundary between the solid tissue and the CSF regions. Therefore, we model the perfusion of drug by:

$$\partial_t u_W(\mathbf{x}, t) - \boldsymbol{\nabla} \cdot (\beta_W \boldsymbol{\nabla} u_W) + \alpha u_W = 0, \qquad \mathbf{x} \in D_W, \tag{136}$$

$$\partial_t u_G(\mathbf{x}, t) - \boldsymbol{\nabla} \cdot (\beta_G \boldsymbol{\nabla} u_G) + \alpha u_G = 0, \qquad \mathbf{x} \in D_G, \tag{137}$$

$$u_W(\mathbf{x}, t) = u_G(\mathbf{x}, t) = g(\mathbf{x}, t), \qquad \mathbf{x} \in \partial D_{CSF}, \tag{138}$$

$$u_W(\mathbf{x}, t) = u_G(\mathbf{x}, t), \qquad \mathbf{x} \in \partial D_G \cap \partial D_W, \tag{139}$$

$$\beta_W \nabla u_W \cdot \mathbf{n}(\mathbf{x}, t) = \beta_G \nabla u_G \cdot \mathbf{n}(\mathbf{x}, t), \qquad \mathbf{x} \in \partial D_G \cap \partial D_W, \tag{140}$$

where β_W, β_G are the diffusion coefficients in the WM and GM, respectively,

and for simplicity we have assumed that the uptake rate of drug, α, is the same in WM and GM.

To approximate Eqs.(136)-(140), we first nondimensionalize using the length scale $L = 64 \ mm$ and time scale $L^2/\bar{\beta}$, where $\bar{\beta}$ is a characteristic diffusion coefficient. Then, we formulate a nondimensional diffuse domain model using a time-dependent version of mDDM1:

$$(\phi_S u_\epsilon)_t - \boldsymbol{\nabla} \cdot (\phi_S \beta \boldsymbol{\nabla} u_\epsilon) + \alpha \phi_S u_\epsilon + \frac{1 - \phi_S}{\epsilon^3}(u_\epsilon - g - r\boldsymbol{n} \cdot \boldsymbol{\nabla} u_\epsilon) = 0, \quad (141)$$

where $\beta = \beta_W \phi_W + \beta_G \phi_G$ is a nondimensional diffusion coefficient and ϕ_S is the hyperbolic tangent function in Eq. (6) using the signed-distance function r obtained by solving Eq. (127) using the initial condition

$$r(\mathbf{x}, 0) = \phi_W + \phi_G - 0.5. \quad (142)$$

Note that this formulation also utilizes a diffuse domain approximation of the continuity of concentration and flux at the WM/GM interfaces [46, 90]. We solve the system using a 3D version of the solver in Sec. 4 on the uniform 256^3 grid (e.g., the nondimensional spatial grid size is $h = 4/256$). We take $g = 1$, $\epsilon = 0.03 \approx 2h$, $\alpha = 0.005$ and time step $\Delta t = 0.001$. The results are shown in Fig. 11, which depicts the evolution of the concentration field in the solid tissue $\phi_s u_\epsilon$ using different diffusion coefficient ratios β_W/β_G.

As time evolves, the extent of the drug penetration into the brain tissue depends on β_W/β_G and by time $t = 0.05$, the drug concentration field has equilibrated. In Fig. 11(a), where $\beta_W = 1$ and $\beta_G = 0.1$, the drug is mainly confined to regions of the brain around the CSF since most of the CSF is surrounded by GM where the diffusion coefficient is small. For this reason in Fig. 11(b), where the diffusivity is larger in GM ($\beta_W = 0.1$ and $\beta_G = 1.0$), the drug penetrates much farther into the brain tissue. The most drug perfusion is observed in Fig. 11(c) where the diffusivities are matched ($\beta_G = \beta_W = 1$). While we considered the diffusivities to be isotropic in the brain tissue, the presence of fibers in the WM can introduce anisotropic diffusion. This can be considered in future work.

41

Figure 11: Drug perfusion in an anatomical template for the human brain from the ICBM atlas [86] (see text). Leftmost column: Light gray (white matter, WM), darker grey (grey matter, GM), darkest grey (cerebral spinal fluid, CSF). A Dirichlet boundary condition is posed at the boundary of the CSF region. The evolution of the drug concentration is shown over time for different ratios of drug diffusivities β_W and β_G in the WM and GM, respectively. (a): $\beta_W/\beta_G = 10$, (b): $\beta_W/\beta_G = 0.1$, (c): $\beta_W/\beta_G = 1$.

7. Conclusions and future work

We have analyzed several diffuse domain methods (DDMs) originally developed in [46] for both the Poisson equation and the diffusion equation with Dirchlet boundary conditions on stationary and moving domains. Advantages of DDM methods is that they are flexible and easy to implement. The same solver can be used for any domain, whether moving or stationary, and that the methods can be implemented using the tools provided in standard software packages so that one does not need to write specialized code that requires treating the region near the interface differently from that in bulk regions.

Our analysis reveals why different DDM formulations yield varying degrees of accuracy. Guided by our analysis, we presented new modifications of the DDMs (mDDMs) that provide higher-order accuracy. While the DDM and mDDM systems require the solution of non-constant coefficient equations that introduce extra length scales to the problem, iterative matrix solvers available in most software packages are sufficient to solve the discrete equations. Here, we used a mass-conservative, finite-difference multigrid method for which the number of iterations is not sensitive to the smoothing thickness ϵ or the local grid size h.

Using a matched asymptotic analysis, two methods, mDDM1 and mDDM3, were shown to be 2nd order accurate in L^2, e.g., $O(\epsilon^2)$ where ϵ is the diffuse interface smoothing parameter. The analysis shows that the errors in the L^∞ norm scale as $O(\epsilon^{1.5})$. Using numerical simulations in 1D, 2D and 3D for selected test cases, where the level-set method was used to implicitly capture the domain movement, to construct the smoothed characteristic function needed by the methods and to perform the needed constant normal extensions, these theoretical predictions were confirmed when the grid size $h \sim \epsilon^{1.5}$ is used to resolve the boundary layer (e.g., K_3 in Sec. 2.3.1).

In addition, numerical simulations revealed that in the L^2 norm, mDDM1 and mDDM3 and their time-dependent versions (mDDMt1, mDDMt3) are $O(\epsilon^2)$ even when $h \propto \epsilon$, which is much cheaper computationally and hence more cost-

effective to use for 2D and 3D problems. In this case, however, the L^∞ errors were found to be more variable over the range of parameters tested. In L^∞, simulations with $h = \epsilon/c$ show that when c is small, the schemes are roughly $O(\epsilon^2)$ whereas when c is large the errors are roughly $O(\epsilon^{1.5})$. This occurs because when c is small, the truncation and analytic errors are close in magnitude but of opposite signs so that there is cancellation in the total error and the total error is smaller than would be expected. However, as h decreases, the analytic errors dominate and the total errors scale as $O(\epsilon^{1.5})$ as predicted by theory.

We also compared the performance of mDDM3 in 3D to the virtual node method in [40], which is more difficult to implement than mDDM3. We found that mDDM3 and the virtual node method converge to the exact solution at a similar rate in L^∞ but that the error is larger using mDDM3 for the same minimum grid size h_{min}. While the reasons for this are unclear, we identified ways that the accuracy of mDDM3 could be further improved. However, it is open question as to whether the mDDM3 could be made as accurate as the virtual node method.

As a final example in 3D, we simulated the dynamics of drug perfusion in an anatomical template of the human brain. Although the geometry is highly complex, no changes in the algorithm are needed. Assuming the drug is administered through the cerebral spinal fluid, which is modeled using a Dirichlet boundary condition, we investigated the effect of different drug diffusivities in the white and grey matter regions of the brain on the spatiotemporal drug distributions.

Although we have focused on the Poisson and diffusion equations here, the mDDMs can be applied to general elliptic and parabolic partial differential equations following previous work (e.g., [46, 48]). For example, consider the following general partial differential equation with Dirichlet boundary conditions in a moving domain $D(t)$,

$$\partial_t u - \nabla \cdot (\boldsymbol{A}\nabla u) + \boldsymbol{b} \cdot \nabla u + cu = f \text{ in } D(t), \tag{143}$$

$$u = g \text{ on } \partial D(t), \tag{144}$$

with $\boldsymbol{A} = \boldsymbol{A}(u, \nabla u, x, t)$ a positive definite matrix, $\boldsymbol{b} = \boldsymbol{b}(u, \nabla u, x, t)$ a vector, $c = c(u, \nabla u, x, t) \in \mathbb{R}$ and $f = f(x, t)$. Then, the corresponding version of mDDMt3 is given by,

$$\partial_t(\phi u) - \nabla \cdot (\phi \boldsymbol{A} \nabla u) + \phi \boldsymbol{b} \cdot \nabla u + \phi c u + \frac{|\nabla \phi|}{\epsilon^2}(u - g - r\boldsymbol{n} \cdot \nabla u) = \phi f \text{ in } \Omega. \quad (145)$$

with $\boldsymbol{A}, \boldsymbol{b}$ and c extended coefficients. The other mDDMs can be defined analogously.

While the matched asymptotic analysis was presented here in for the Poisson and diffusion equations 1D, the same results hold in 2D and 3D for arbitrary smooth domains and for general PDEs such as in Eqs. (143)-(144). This is because in 2D and 3D, the boundary appears flat at leading order and the effects of curvature appear in lower order terms that ultimately do not affect the expansions to the orders we calculated here, provided that the interface thickness ϵ is sufficiently small, e.g., smaller than any physically relevant length scale, such as the radius of curvature of the boundary, and the region around the boundary over which functions can be smoothly extended out of the physical domain. The additional terms in the equations (advection and reaction) also do not affect the expansions we calculated here because they are lower order as well.

In future work, we plan to develop a theoretical proof of convergence for the mDDMs. In addition, we plan to apply the methods developed here to important physical applications including fluid flows in complex geometries, which involves applying the theory developed here to the Navier-Stokes equations, and to the epitaxial growth of graphene heterostructures, which involves solving diffusion equations in domains that evolve according to flux balance conditions determined from the solutions to the diffusion equations themselves.

Appendices

A. Additional numerical results of DDMs for 1D time-independent problems

We have tested the following three DDMs using $h = \epsilon/4$:

DDM1: $\nabla(\phi \nabla u) - \frac{1}{\epsilon^3}(1 - \phi)(u - g) = \phi f$,

DDM2: $\phi \Delta u - \frac{1}{\epsilon^2}(1 - \phi)(u - g) = \phi f$,

DDM3: $\nabla(\phi \nabla u) - \frac{1}{\epsilon^2}|\nabla \phi|(u - g) = \phi f$,

on the following seven cases,

Case1: $u = \frac{x^2}{2}$;

Case2: $u = (x^2 - 1.111)^2$;

Case3: $u = \frac{1}{x^2 + 1}$;

Case4: $u = \cos(x)$;

Case5: $u = (x^2 + 1)^2$;

Case6: $u = \log(x^2 + 1)$;

Case7: $u = \sqrt{x^2 + 1}$.

The original Poisson equation with Dirichlet boundary condition is again defined on $[-1.111, 1.111]$ and we solve the DDMs on a larger domain $[-2, 2]$. Tables 7 - 12 show the L^2 and L^∞ norms of errors for the three DDMs for the seven cases. In cases 3-7, the convergence orders of DDM1-3 in both L^2 and L^∞ are similar to those in case 1. However in case 2, where the derivative of the exact solution is 0 at the boundary ($A = 0$), we observe 2nd order convergence in DDM2 and higher than 1st order convergence in DDM1 and DDM3. Our analysis in Sec. 2.2 indicates that $\bar{u}_1^{(1)}(0) = \frac{A}{3}(-\ln 6 + \gamma)$, see Eq. (41), $-\frac{A}{6}\ln \epsilon + \frac{A}{3}(-\ln 6 + \gamma)$, see Eq. (71), and $-\frac{A}{6}\ln \epsilon + \frac{A}{3}(-\ln(\frac{1}{2}\sqrt{\frac{2}{3}}) + \gamma)$, see Eq. (73) for DDM2, DDM1 and DDM3, respectively. Note that these all vanish when $A = 0$. Hence, DDMs can achieve higher than 1st order accuracy when $A = 0$. In fact, the errors in both norms for DDM1 and DDM3 are dominated by $O(\epsilon^2(\ln(\epsilon))^2)$. This term rises in the next order matching in K_2, that is $\lim_{z_2 \to -\infty} \hat{u}_2^{(2)} = \bar{u}_2^{(2)}(0) + (z_2 + \ln \epsilon/6)^2 \frac{d^2}{dx^2}\bar{u}_1^{(0)}(0) + \dots$. Using an analogous argument to that

given in Sec. 2.2.2, we obtain $\bar{u}_2^{(2)}(0) \sim O((\ln \epsilon)^2)$. In Fig. 12(a), we plot $D(\epsilon) = \lim_{z_2 \to -\infty}(u_\epsilon - u)/\epsilon^2$ versus $\ln(\epsilon)$ using the numerical solution of DDM1 in case 2 and find that it is a quadratic function of $\ln(\epsilon)$.

Figure 12: (a): $D(\epsilon)$ for DDM1 in case 2, (b): $D(\epsilon)$ for mDDM1 in case 2.

ϵ	case 1	k	case 2	k	case 3	k
2.00E-01	4.88E-01	0.00	1.55E-02	0.00	6.79E-02	0.00
1.00E-01	1.12E-01	2.12	2.75E-03	2.50	1.59E-02	2.10
5.00E-02	9.88E-03	3.51	9.39E-05	4.87	1.40E-03	3.51
2.50E-02	1.31E-02	-0.41	1.93E-04	-1.04	1.87E-03	-0.43
1.25E-02	1.41E-02	-0.11	1.15E-04	0.75	2.02E-03	-0.10
6.25E-03	1.04E-02	0.44	4.94E-05	1.22	1.49E-03	0.44
3.13E-03	6.80E-03	0.62	1.85E-05	1.41	9.70E-04	0.62
1.56E-03	4.16E-03	0.71	6.46E-06	1.52	5.94E-04	0.71

case 4	k	case 5	k	case 6	k	case 7	k
1.25E-01	0.00	4.88E-01	0.00	3.03E-01	0.00	7.32E-02	0.00
2.95E-02	2.08	1.13E-01	2.10	6.97E-02	2.12	1.72E-02	2.09
2.63E-03	3.49	1.01E-02	3.49	6.09E-03	3.52	1.52E-03	3.50
3.52E-03	-0.42	1.33E-02	-0.40	8.11E-03	-0.41	2.03E-03	-0.42
3.80E-03	-0.11	1.44E-02	-0.11	8.72E-03	-0.10	2.19E-03	-0.11
2.81E-03	0.43	1.07E-02	0.43	6.44E-03	0.44	1.62E-03	0.43
1.83E-03	0.62	6.95E-03	0.62	4.19E-03	0.62	1.06E-03	0.62
1.12E-03	0.71	4.25E-03	0.71	2.57E-03	0.71	6.46E-04	0.71

Table 7: The L^2 errors for DDM1

ϵ	case 1	k	case 2	k	case 3	k
2.00E-01	2.12E-01	0.00	1.09E-02	0.00	5.36E-02	0.00
1.00E-01	5.26E-02	2.01	1.93E-03	2.49	1.28E-02	2.07
5.00E-02	7.96E-03	2.72	8.82E-05	4.45	1.93E-03	2.73
2.50E-02	5.89E-03	0.44	1.42E-04	-0.69	1.46E-03	0.40
1.25E-02	6.31E-03	-0.10	7.86E-05	0.86	1.57E-03	-0.10
6.25E-03	4.66E-03	0.44	3.28E-05	1.26	1.16E-03	0.44
3.13E-03	3.04E-03	0.62	1.22E-05	1.43	7.53E-04	0.62
1.56E-03	1.86E-03	0.71	4.20E-06	1.53	4.60E-04	0.71

case 4	k	case 5	k	case 6	k	case 7	k
1.04E-01	0.00	2.32E-01	0.00	1.47E-01	0.00	5.82E-02	0.00
2.57E-02	2.02	5.81E-02	2.00	3.60E-02	2.03	1.43E-02	2.02
3.88E-03	2.72	8.83E-03	2.72	5.43E-03	2.73	2.17E-03	2.72
2.91E-03	0.41	6.48E-03	0.45	4.06E-03	0.42	1.62E-03	0.42
3.14E-03	-0.11	6.96E-03	-0.10	4.35E-03	-0.10	1.74E-03	-0.10
2.32E-03	0.43	5.14E-03	0.44	3.21E-03	0.44	1.29E-03	0.44
1.51E-03	0.62	3.35E-03	0.62	2.09E-03	0.62	8.40E-04	0.62
9.26E-04	0.71	2.05E-03	0.71	1.28E-03	0.71	5.14E-04	0.71

Table 8: The L^∞ errors for DDM1

ϵ	case 1	k	case 2	k	case 3	k
2.00E-01	3.42E-01	0.00	5.80E-02	0.00	4.44E-02	0.00
1.00E-01	1.70E-01	1.01	1.37E-02	2.09	2.28E-02	0.96
5.00E-02	8.45E-02	1.01	3.31E-03	2.04	1.16E-02	0.98
2.50E-02	4.17E-02	1.02	8.19E-04	2.02	5.84E-03	0.99
1.25E-02	2.07E-02	1.01	2.03E-04	2.01	2.93E-03	0.99
6.25E-03	1.03E-02	1.00	5.07E-05	2.00	1.47E-03	1.00
3.13E-03	5.16E-03	1.00	1.27E-05	2.00	7.35E-04	1.00
1.56E-03	2.58E-03	1.00	3.16E-06	2.00	3.68E-04	1.00

case 4	k	case 5	k	case 6	k	case 7	k
9.27E-02	0.00	3.74E-01	0.00	1.99E-01	0.00	5.26E-02	0.00
4.54E-02	1.03	1.79E-01	1.06	1.01E-01	0.97	2.60E-02	1.02
2.25E-02	1.02	8.77E-02	1.03	5.12E-02	0.99	1.29E-02	1.01
1.12E-02	1.00	4.30E-02	1.03	2.55E-02	1.01	6.44E-03	1.01
5.59E-03	1.00	2.13E-02	1.01	1.27E-02	1.00	3.21E-03	1.00
2.79E-03	1.00	1.06E-02	1.01	6.36E-03	1.00	1.60E-03	1.00
1.39E-03	1.00	5.29E-03	1.00	3.18E-03	1.00	8.02E-04	1.00
6.97E-04	1.00	2.64E-03	1.00	1.59E-03	1.00	4.01E-04	1.00

Table 9: The L^2 errors for DDM2

ϵ	case 1	k	case 2	k	case 3	k
2.00E-01	1.65E-01	0.00	3.82E-02	0.00	3.53E-02	0.00
1.00E-01	7.95E-02	1.05	9.03E-03	2.08	1.81E-02	0.96
5.00E-02	3.88E-02	1.04	2.19E-03	2.04	9.11E-03	0.99
2.50E-02	1.91E-02	1.02	5.42E-04	2.02	4.61E-03	0.98
1.25E-02	9.51E-03	1.00	1.35E-04	2.01	2.33E-03	0.98
6.25E-03	4.82E-03	0.98	3.38E-05	2.00	1.19E-03	0.97
3.13E-03	2.40E-03	1.00	8.42E-06	2.00	5.93E-04	1.00
1.56E-03	1.20E-03	1.01	2.10E-06	2.00	2.96E-04	1.00

case 4	k	case 5	k	case 6	k	case 7	k
7.84E-02	0.00	1.94E-01	0.00	1.06E-01	0.00	4.34E-02	0.00
3.83E-02	1.03	9.08E-02	1.10	5.26E-02	1.01	2.13E-02	1.03
1.88E-02	1.03	4.36E-02	1.06	2.61E-02	1.01	1.05E-02	1.02
9.39E-03	1.00	2.13E-02	1.04	1.30E-02	1.01	5.22E-03	1.01
4.71E-03	0.99	1.06E-02	1.01	6.50E-03	1.00	2.62E-03	1.00
2.40E-03	0.97	5.34E-03	0.98	3.31E-03	0.98	1.33E-03	0.98
1.20E-03	1.00	2.66E-03	1.01	1.65E-03	1.00	6.64E-04	1.00
5.96E-04	1.01	1.32E-03	1.01	8.22E-04	1.01	3.31E-04	1.01

Table 10: The L^∞ errors for DDM2

ϵ	case 1	k	case 2	k	case 3	k
2.00E-01	3.82E-02	0.00	1.37E-02	0.00	6.82E-03	0.00
1.00E-01	9.45E-02	-1.31	7.22E-03	0.92	1.44E-02	-1.08
5.00E-02	8.03E-02	0.24	3.12E-03	1.21	1.17E-02	0.30
2.50E-02	5.39E-02	0.58	1.19E-03	1.39	7.79E-03	0.59
1.25E-02	3.33E-02	0.69	4.18E-04	1.51	4.80E-03	0.70
6.25E-03	1.97E-02	0.76	1.40E-04	1.58	2.83E-03	0.76
3.13E-03	1.14E-02	0.79	4.49E-05	1.64	1.63E-03	0.80
1.56E-03	6.42E-03	0.82	1.43E-05	1.66	9.18E-04	0.83

case 4	k	case 5	k	case 6	k	case 7	k
1.10E-02	0.00	3.57E-02	0.00	2.61E-02	0.00	6.36E-03	0.00
2.59E-02	-1.23	9.39E-02	-1.39	6.02E-02	-1.21	1.50E-02	-1.23
2.16E-02	0.26	8.08E-02	0.22	5.02E-02	0.26	1.25E-02	0.26
1.46E-02	0.57	5.46E-02	0.56	3.35E-02	0.58	8.39E-03	0.57
9.01E-03	0.69	3.39E-02	0.69	2.06E-02	0.70	5.19E-03	0.69
5.34E-03	0.76	2.01E-02	0.75	1.22E-02	0.76	3.07E-03	0.76
3.07E-03	0.80	1.16E-02	0.79	7.02E-03	0.80	1.77E-03	0.80
1.74E-03	0.82	6.57E-03	0.82	3.96E-03	0.82	9.99E-04	0.82

Table 11: The L^2 errors for DDM3

ϵ	case 1	k	case 2	k	case 3	k
2.00E-01	4.73E-02	0.00	9.52E-03	0.00	1.06E-02	0.00
1.00E-01	4.42E-02	0.10	4.92E-03	0.95	1.15E-02	-0.11
5.00E-02	3.66E-02	0.27	2.08E-03	1.24	9.17E-03	0.32
2.50E-02	2.44E-02	0.59	7.82E-04	1.41	6.08E-03	0.59
1.25E-02	1.50E-02	0.70	2.72E-04	1.52	3.73E-03	0.70
6.25E-03	8.84E-03	0.76	9.04E-05	1.59	2.20E-03	0.76
3.13E-03	5.09E-03	0.80	2.89E-05	1.64	1.26E-03	0.80
1.56E-03	2.87E-03	0.82	9.14E-06	1.66	7.12E-04	0.83

case 4	k	case 5	k	case 6	k	case 7	k
2.27E-02	0.00	5.44E-02	0.00	3.11E-02	0.00	1.26E-02	0.00
2.19E-02	0.06	4.75E-02	0.19	3.12E-02	-0.01	1.22E-02	0.04
1.80E-02	0.28	3.99E-02	0.25	2.54E-02	0.30	1.00E-02	0.28
1.21E-02	0.58	2.67E-02	0.58	1.68E-02	0.59	6.71E-03	0.58
7.45E-03	0.70	1.65E-02	0.70	1.03E-02	0.70	4.14E-03	0.70
4.40E-03	0.76	9.74E-03	0.76	6.09E-03	0.76	2.44E-03	0.76
2.53E-03	0.80	5.61E-03	0.79	3.50E-03	0.80	1.41E-03	0.80
1.43E-03	0.82	3.17E-03	0.82	1.98E-03	0.82	7.94E-04	0.82

Table 12: The L^∞ errors for DDM3

B. Additional numerical results of mDDMs for 1D time-independent problems

We have tested the following three mDDMs using $h = \epsilon^{1.5}/4$:

mDDM1: $\nabla(\phi\nabla u) - \frac{1}{\epsilon^3}(1-\phi)(u - g - r\boldsymbol{n} \cdot \nabla u) = \phi f$,

mDDM2: $\phi\Delta u - \frac{1}{\epsilon^2}(1-\phi)(u - g - r\boldsymbol{n} \cdot \nabla u) = \phi f$,

mDDM3: $\nabla(\phi\nabla u) - \frac{1}{\epsilon^2}|\nabla\phi|(u - g - r\boldsymbol{n} \cdot \nabla u) = \phi f$,

on the seven cases given in Appendix A. The problem setup and the numerical discretization are analogous to those in Appendix A. In case 3-7, mDDM1-3 perform analogously as those in case 1. However in case 2, in which $A = 0$, we

observe similar behavior as those corresponding DDMs presented in Appendix A. Our analysis in Sec. 2.3.2 indicates $\bar{u}_1^{(1)}(0) \approx -A/2.92$ (Eq. (119)) and $\hat{u}_1^{(1)}(0) \approx Ae^{1/36}/2.92$ (Eq. (120)) for mDDM2, which both vanish when $A = 0$. Hence, mDDM2 can achieve 2nd order accuracy when $A = 0$. As for mDDM1 and mDDM3, the errors are again dominated by $O(\epsilon^2(\ln(\epsilon))^2)$, whose coefficient is not affected by our modification in the next order matching when $A = 0$. In Fig. 12(b), we plot $D(\epsilon) = \lim_{z_2 \to -\infty}(u_\epsilon - u)/\epsilon^2$ versus $\ln(\epsilon)$ using the numerical solution of mDDM1 in case 2 and find that it is a quadratic function of $\ln(\epsilon)$.

ϵ	case 1	k	case 2	k	case 3	k
2.00E-01	4.88E-01	0.00	3.11E-02	0.00	7.00E-02	0.00
1.00E-01	1.50E-01	1.70	9.11E-03	1.77	2.19E-02	1.68
5.00E-02	4.46E-02	1.75	2.64E-03	1.79	6.57E-03	1.73
2.50E-02	1.26E-02	1.83	8.14E-04	1.70	1.87E-03	1.81
1.25E-02	3.34E-03	1.91	2.69E-04	1.60	5.05E-04	1.89
6.25E-03	8.61E-04	1.95	9.25E-05	1.54	1.33E-04	1.92

case 4	k	case 5	k	case 6	k	case 7	k
1.25E-01	0.00	4.81E-01	0.00	3.07E-01	0.00	7.37E-02	0.00
3.96E-02	1.66	1.49E-01	1.69	9.44E-02	1.70	2.31E-02	1.68
1.20E-02	1.73	4.45E-02	1.74	2.81E-02	1.75	6.94E-03	1.73
3.40E-03	1.82	1.25E-02	1.83	7.94E-03	1.82	1.97E-03	1.82
9.09E-04	1.90	3.31E-03	1.92	2.12E-03	1.90	5.25E-04	1.90
2.36E-04	1.95	8.44E-04	1.97	5.53E-04	1.94	1.36E-04	1.95

Table 13: The L^2 errors for mDDM1

ϵ	case 1	k	case 2	k	case 3	k
2.00E-01	2.10E-01	0.00	2.06E-02	0.00	5.51E-02	0.00
1.00E-01	6.59E-02	1.67	5.93E-03	1.80	1.71E-02	1.69
5.00E-02	1.98E-02	1.74	1.70E-03	1.80	5.12E-03	1.74
2.50E-02	5.58E-03	1.83	5.22E-04	1.70	1.45E-03	1.82
1.25E-02	1.55E-03	1.85	1.72E-04	1.60	3.92E-04	1.89
6.25E-03	4.91E-04	1.65	5.91E-05	1.54	1.22E-04	1.68

case 4	k	case 5	k	case 6	k	case 7	k
1.05E-01	0.00	2.26E-01	0.00	1.49E-01	0.00	5.85E-02	0.00
3.29E-02	1.67	7.11E-02	1.67	4.64E-02	1.68	1.83E-02	1.67
9.90E-03	1.73	2.14E-02	1.73	1.39E-02	1.74	5.51E-03	1.73
2.81E-03	1.82	6.01E-03	1.83	3.94E-03	1.82	1.56E-03	1.82
7.71E-04	1.86	1.70E-03	1.82	1.07E-03	1.88	4.28E-04	1.87
2.45E-04	1.65	5.42E-04	1.65	3.39E-04	1.66	1.36E-04	1.65

Table 14: The L^∞ errors for mDDM1

ϵ	case 1	k	case 2	k	case 3	k
2.00E-01	2.85E-01	0.00	3.73E-02	0.00	3.75E-02	0.00
1.00E-01	1.41E-01	1.01	8.77E-03	2.09	1.92E-02	0.97
5.00E-02	6.98E-02	1.01	2.13E-03	2.04	9.70E-03	0.98
2.50E-02	3.46E-02	1.01	5.24E-04	2.02	4.88E-03	0.99
1.25E-02	1.73E-02	1.00	1.30E-04	2.01	2.45E-03	1.00
6.25E-03	8.62E-03	1.00	3.23E-05	2.01	1.23E-03	1.00

case 4	k	case 5	k	case 6	k	case 7	k
7.65E-02	0.00	3.07E-01	0.00	1.68E-01	0.00	4.36E-02	0.00
3.77E-02	1.02	1.48E-01	1.05	8.48E-02	0.98	2.16E-02	1.01
1.87E-02	1.01	7.23E-02	1.03	4.25E-02	1.00	1.08E-02	1.01
9.33E-03	1.00	3.57E-02	1.02	2.12E-02	1.00	5.37E-03	1.00
4.66E-03	1.00	1.77E-02	1.01	1.06E-02	1.00	2.68E-03	1.00
2.33E-03	1.00	8.84E-03	1.00	5.31E-03	1.00	1.34E-03	1.00

Table 15: The L^2 errors for mDDM2

ϵ	case 1	k	case 2	k	case 3	k
2.00E-01	1.33E-01	0.00	2.50E-02	0.00	2.96E-02	0.00
1.00E-01	6.48E-02	1.04	5.86E-03	2.09	1.52E-02	0.96
5.00E-02	3.19E-02	1.02	1.42E-03	2.05	7.67E-03	0.98
2.50E-02	1.59E-02	1.01	3.52E-04	2.01	3.88E-03	0.98
1.25E-02	7.93E-03	1.00	8.73E-05	2.01	1.95E-03	1.00
6.25E-03	3.96E-03	1.00	2.18E-05	2.00	9.77E-04	1.00

case 4	k	case 5	k	case 6	k	case 7	k
6.43E-02	0.00	1.55E-01	0.00	8.67E-02	0.00	3.56E-02	0.00
3.17E-02	1.02	7.35E-02	1.07	4.34E-02	1.00	1.76E-02	1.02
1.57E-02	1.01	3.57E-02	1.04	2.16E-02	1.00	8.73E-03	1.01
7.89E-03	1.00	1.77E-02	1.02	1.09E-02	1.00	4.38E-03	1.00
3.94E-03	1.00	8.78E-03	1.01	5.43E-03	1.00	2.19E-03	1.00
1.97E-03	1.00	4.39E-03	1.00	2.72E-03	1.00	1.09E-03	1.00

Table 16: The L^∞ errors for mDDM2

ϵ	case 1	k	case 2	k	case 3	k
2.00E-01	1.55E-01	0.00	3.89E-02	0.00	2.80E-02	0.00
1.00E-01	3.77E-02	2.04	1.38E-02	1.50	7.15E-03	1.97
5.00E-02	8.54E-03	2.14	4.87E-03	1.50	1.80E-03	1.99
2.50E-02	1.89E-03	2.17	1.72E-03	1.50	4.71E-04	1.94
1.25E-02	4.16E-04	2.19	5.88E-04	1.55	1.27E-04	1.89
6.25E-03	8.74E-05	2.25	1.94E-04	1.60	3.46E-05	1.88

case 4	k	case 5	k	case 6	k	case 7	k
4.42E-02	0.00	1.44E-01	0.00	1.07E-01	0.00	2.57E-02	0.00
1.08E-02	2.03	3.31E-02	2.12	2.69E-02	1.99	6.31E-03	2.02
2.51E-03	2.10	6.78E-03	2.29	6.50E-03	2.05	1.48E-03	2.09
5.86E-04	2.10	1.25E-03	2.44	1.59E-03	2.03	3.48E-04	2.09
1.38E-04	2.09	1.91E-04	2.71	4.01E-04	1.99	8.28E-05	2.07
3.20E-05	2.11	1.39E-05	3.78	1.01E-04	1.98	1.96E-05	2.08

Table 17: The L^2 errors for mDDM3

ϵ	case 1	k	case 2	k	case 3	k
2.00E-01	8.23E-02	0.00	2.58E-02	0.00	2.18E-02	0.00
1.00E-01	2.23E-02	1.88	8.98E-03	1.52	5.82E-03	1.91
5.00E-02	5.96E-03	1.91	3.14E-03	1.51	1.56E-03	1.90
2.50E-02	2.03E-03	1.55	1.10E-03	1.51	5.17E-04	1.59
1.25E-02	6.15E-04	1.72	3.76E-04	1.55	1.56E-04	1.73
6.25E-03	2.29E-04	1.42	1.24E-04	1.60	5.73E-05	1.44

case 4	k	case 5	k	case 6	k	case 7	k
4.10E-02	0.00	8.94E-02	0.00	5.78E-02	0.00	2.29E-02	0.00
1.12E-02	1.88	2.40E-02	1.89	1.58E-02	1.88	6.22E-03	1.88
2.99E-03	1.90	6.39E-03	1.91	4.22E-03	1.90	1.67E-03	1.90
1.02E-03	1.56	2.21E-03	1.53	1.41E-03	1.58	5.64E-04	1.56
3.07E-04	1.72	6.72E-04	1.72	4.27E-04	1.73	1.71E-04	1.72
1.14E-04	1.43	2.52E-04	1.41	1.58E-04	1.43	6.35E-05	1.43

Table 18: The L^∞ errors for mDDM3

C. Validation of the asymptotic analysis

Here we present validations of our asymptotic analysis using the numerical results for the seven cases. In Tab. 19, we present $\bar{u}_1^{(1)}(0)$ from DDM2 obtained from both our asymptotic analysis theory (Eq. (41)) and the numerical results. In Tab. 20, we compare the slope of $C(\epsilon)$ computed numerically (through an analogous linear fit as in Sec. 2.2.2) with that derived from our asymptotic analysis theory $(-A/6)$ for DDM1 and DDM3. In Tab. 21, we show $\hat{u}_3^{(1.5)}(0)$ from our asymptotic theory $(-A/\sqrt{2\pi}$ for mDDM1 and $-A/\sqrt{6\pi}$ for mDDM3) together with those calculated from the numerical results for mDDM1 and mDDM3. in Tab. 22, $\bar{u}_1^{(1)}(0)$ and $\hat{u}^{(1)}(0)$ from mDDM2 obtained from both our asymptotic analysis theory (Eqs. (119) and (120)) and the numerical results are presented. Clearly our theory is consistent with the numerical results.

case	Theory	Numerics
case 1	-0.450	-0.450
case 2	0.000	0.000
case 3	0.180	0.183
case 4	0.363	0.368
case 5	-4.020	-4.072
case 6	-0.403	-0.408
case 7	-0.301	-0.305

Table 19: Comparisons between $\bar{u}_1^{(1)}$ from theory and that from numerical results.

case	Theory	Numerics (DDM1)	Numerics (DDM3)
case 1	-0.185	-0.186	-0.188
case 2	0.000	0.000	0.000
case 3	0.074	0.075	0.075
case 4	0.149	0.150	0.152
case 5	-1.655	-1.667	-1.686
case 6	-0.166	-0.167	-0.168
case 7	-0.124	-0.125	-0.126

Table 20: Comparisons between the slope of $C(\epsilon)$ from theory and that from numerical results.

	mDDM1		mDDM3	
case	Theory	Numerics	Theory	Numerics
case 1	-0.443	-0.442	-0.256	-0.257
case 2	0.000	0.001	0.000	-0.001
case 3	0.178	0.177	0.103	0.103
case 4	0.356	0.357	0.206	0.207
case 5	-3.961	-3.952	-2.287	-2.299
case 6	-0.397	-0.396	-0.229	-0.230
case 7	-0.297	-0.296	-0.171	-0.172

Table 21: Comparisons between $\hat{u}_3^{(1.5)}(0)$ from asymptotic theory and from numerical simulations of mDDM1 and mDDM3.

	$\bar{u}^{(1)}(0)$		$\hat{u}^{(1)}(0)$	
case	Predictions	Numerics	Predictions	Numerics
case 1	-0.381	-0.380	-0.391	-0.391
case 2	0.000	0.000	0.000	0.000
case 3	0.152	0.152	0.157	0.157
case 4	0.307	0.307	0.316	0.315
case 5	-3.401	-3.399	-3.496	-3.495
case 6	-0.341	-0.340	-0.350	-0.350
case 7	-0.255	-0.254	-0.262	-0.262

Table 22: Comparisons between the asymptotic theory and the numerical results for $\bar{u}^{(1)}(0)$ and $\hat{u}^{(1)}(0)$ from mDDM2.

D. Derivation of the solution to Eq. (108)

Recall the homogeneous ordinary differential equation from Eq. (108) in the main text,

$$y'' - e^{6x}(y - xy') = 0.$$

Clearly, $y_1 = x$ is one of the linearly independent solutions. We derive the other solution through a reduction of order. We assume $y_2(x) = v(x)y_1(x) = xv(x)$ and plug into the equation to get:

$$xv'' + 2v' + x^2 e^{6x} v' = 0, \tag{146}$$

which gives $v(x) = \int \frac{e^{e^{6x}(1-6x)/36}}{x^2} dx$ for $x \neq 0$. Hence,

$$y_2 = x \int \frac{e^{e^{6x}(1-6x)/36}}{x^2} dx = -e^{e^{6x}(1-6x)/36} - x \int h(x)dx \text{ for } x \neq 0, \tag{147}$$

where $h(x) = e^{e^{6x}(1-6x)/36+6x}$. It is not hard to verify that y_2 is a solution to the equation for all x including 0. Although the anti-derivative of $h(x)$ is not an elementary function, $h(x) \in L^1(-\infty, +\infty)$ and $\int_{-\infty}^{+\infty} h(x)dx \approx 2.92$. Let $H'(x) = h(x)$, then y_2 can be written as

$$y_2 = -e^{e^{6x}(1-6x)/36} - x \int_0^x h(t)dt - H(0)x. \tag{148}$$

Note that $H(0)x$ is linearly dependent with respect to y_1, thus y_2 can be simplified as

$$y_2 = -e^{e^{6x}(1-6x)/36} - x \int_0^x h(t)dt. \tag{149}$$

Hence, the general solution to Eq. (108) is

$$y = C_1 x + C_2(-e^{e^{6x}(1-6x)/36} - x \int_0^x h(t)dt). \tag{150}$$

E. Analysis of the truncation and analytic errors

In Sec. 3.2, we showed that the errors using mDDMt3 to solve time-dependent problems could actually be smaller using $h = \epsilon/c$ when c is small (e.g., $c = 4$) than when c is large (e.g., $c = 128$). Recall Fig. 6 and Tables 5 and 6. Here, we present an analysis of the numerical results to explain this surprising behavior. Because the errors are dominated by the space discretization, we consider, for simplicity, the analogous time-independent problem in case 1 using mDDM3 (see App. A for a definition of case 1; this is also the problem considered in Sec. 2.1).

In Fig. 13 below, we show that for the time-independent problem in case 1, the solution using mDDM3 with $c = 4$ is also closer to the exact solution than those with larger values of c. Indeed, the behavior of the solution and the errors are very similar to the time-dependent case presented in the main text in Sec. 3.2. The actual relative errors $||u_{h,\epsilon} - u||_\infty / ||u||_\infty$ are given in Table 23. To understand this behavior write the total error as

$$||u_{h,\epsilon} - u||_\infty = ||\,(u_{h,\epsilon} - u_\epsilon) + (u_\epsilon - u)\,||_\infty,$$

where u_ϵ is the solution of the continuous modified equation. The first term on the right hand side is the truncation error and the second term on the right is the analytic error. The truncation error can be approximated using the consecutive errors $E^\infty_{h,h/2,\epsilon} = u_{h,\epsilon} - u_{h/2,\epsilon}$ evaluated at the point $x^*_{h,h/2} = argmax|u_{h,\epsilon} - u_{h/2,\epsilon}|$. The consecutive errors with $h = \epsilon/c$ are shown in Table 24 below. Note that there is no norm in the evaluation of $E^\infty_{h,h/2,\epsilon}$ and that all the consecutive errors are positive. The truncation error can then be approximated by $u_{h,\epsilon} - u_\epsilon \approx \sum_{j=0}^\infty E^\infty_{h/2^j,h/2^{j+1},\epsilon}$, which is thus also positive. Next, as shown in Sec. 2.3.1, the analytic error near the boundary of the physical domain D can be approximated to leading order as $u_\epsilon - u \sim -\frac{A\epsilon^{1.5}}{\sqrt{6\pi}} < 0$. where A is the normal derivative of the analytic solution at the boundary ($A = 1.111$ for the problem in case 1). The leading-order analytic relative errors $\frac{u_\epsilon - u}{||u||_\infty}$ are shown in the last row of Table 24 and are of a similar magnitude as the truncation errors (at least for ϵ smaller than 0.1 and c is small) but have the opposite sign. For moderate sizes of h and ϵ, the truncation and analytic errors partially cancel one another. As h decreases, however, the analytic error tends to dominate and the convergence rate decreases from close to 2 for larger h (smaller c) to 1.5 for small h (large c). A careful analysis shows that when ϵ is small (e.g., 0.05, 0.025, 0.0125), the combination of truncation and leading-order asymptotic analytic errors provides a good estimate of the total error. For larger ϵ, the estimate is not as good because the leading-order asymptotic error does not provide as good an estimate for the overall analytic error.

Figure 13: Numerical solutions of mDDM3 near the boundary x=1.111 for the time-independent problem in case 1 with different c such that $c = \epsilon/h$. The green dashed line shows the extension of the exact solution outside the domain (e.g., constant in the normal direction). The vertical dotted line represents the boundary of the physical domain (x=1.111). Observe that the mDDM3 solution with c=4 is closer to the exact solution than those with larger c

ϵ	c=4	$\mathbf{k}$	c=16	$\mathbf{k}$	c=128	$\mathbf{k}$
0.2	8.19E-02	0.00	8.15E-02	0.00	8.29E-02	0.00
0.1	2.05E-02	2.00	2.22E-02	1.88	2.38E-02	1.80
0.05	4.26E-03	2.26	6.34E-03	1.81	7.12E-03	1.74
0.025	9.22E-04	2.21	2.17E-03	1.55	2.20E-03	1.69
0.0125	2.93E-04	1.66	6.55E-04	1.73	6.82E-04	1.69
0.00625	8.76E-05	1.74	1.77E-04	1.88	2.30E-04	1.57

Table 23: The total L^∞ relative error $\frac{||u_{h,\epsilon}-u||_\infty}{||u||_\infty}$ with $h = \epsilon/c$, as labeled, for the time-independent problem in case 1 using mDDM3.

c	$\epsilon = 0.2$		$\epsilon = 0.1$		$\epsilon = 0.05$		$\epsilon = 0.025$		$\epsilon = 0.0125$	
	$E^{\infty,r}_{h,h/2,\epsilon}$	k	$E^{\infty,r}_{h,h/2,\epsilon}$	k	$E^{\infty,r}_{h,h/2,\epsilon}$	k	$E^{\infty,r}_{h,h/2,\epsilon}$	k	$E^{\infty,r}_{h,h/2,\epsilon}$	k
4	2.90E-03		2.24E-03		1.41E-03		3.92E-04		3.29E-04	
8	5.37E-04	2.43	4.14E-04	2.43	2.36E-04	2.58	1.54E-04	1.34	1.50E-04	1.13
16	1.15E-04	2.22	7.61E-05	2.44	6.09E-05	1.96	5.62E-05	1.46	4.13E-05	1.86
32	2.73E-05	2.07	1.87E-05	2.03	1.54E-05	1.98	1.60E-05	1.81	1.27E-05	1.7
64	6.73E-06	2.02	4.64E-06	2.01	4.23E-06	1.87	4.09E-06	1.97	3.29E-06	1.94
128	1.68E-06	2	1.16E-06	2.01	1.06E-06	2	1.03E-06	1.99	8.30E-07	1.99
256	4.24E-07	1.99	2.85E-07	2.02	2.63E-07	2.01	2.57E-07	2	2.11E-07	1.98
512	1.06E-07	1.99	6.71E-08	2.09	6.36E-08	2.05	6.46E-08	1.99	5.30E-08	1.99
Analytic error	-4.58E-02		-1.62E-02		-5.7E-03		-2.02E-03		-7.16E-04	

Table 24: The consecutive relative truncation errors $E^{\infty,r}_{h,h/2,\epsilon}/||u||_\infty$ with $c = \epsilon/h$ as labeled for the time-independent problem in case 1 using mDDM3. The results show that for fixed ϵ, the consecutive errors are positive and that the scheme converges with 2nd order accuracy in h. The last row shows the leading-order asymptotic analytic relative error $\frac{u_\epsilon - u}{||u||_\infty} \sim -\frac{2\epsilon^{1.5}}{\sqrt{6\pi}}$.

F. Numerical results for mDDMt3 for 1D time-dependent problems on moving domains

In Tables 25 and 26 below, we present the L^2 and L^∞ errors for the problem considered in Sec. 3.2 and order of convergence **k** with $h = \epsilon/c$.

ϵ	c = 4	**k**	c = 16	**k**	c = 128	**k**
0.2	3.07E-02	0.00	3.07E-02	0.00	3.07E-02	0.00
0.1	7.92E-03	1.95	8.05E-03	1.93	8.06E-03	1.93
0.05	1.90E-03	2.06	1.96E-03	2.04	1.96E-03	2.04
0.025	4.74E-04	2.00	4.79E-04	2.03	4.79E-04	2.03
0.0125	1.18E-04	2.01	1.21E-04	1.99	1.21E-04	1.99
0.00625	2.97E-05	1.99	3.12E-05	1.96	3.13E-05	1.95

Table 25: The L^2 errors for simulating the time-dependent (diffusion) equation using mDDMt3 with $h = \epsilon/c$ on a moving domain.

ϵ	c = 4	**k**	c = 16	**k**	c = 128	**k**
0.2	4.38E-02	0.00	4.35E-02	0.00	4.51E-02	0.00
0.1	1.16E-02	1.92	1.25E-02	1.80	1.35E-02	1.74
0.05	2.57E-03	2.17	3.62E-03	1.79	4.05E-03	1.74
0.025	6.36E-04	2.01	1.23E-03	1.56	1.25E-03	1.70
0.0125	2.01E-04	1.66	3.69E-04	1.74	3.85E-04	1.70
0.00625	5.01E-05	2.00	1.00E-04	1.88	1.29E-04	1.58

Table 26: The L^∞ errors for simulating the time-dependent (diffusion) equation using mD-DMt3 with $h = \epsilon/c$ on a moving domain.

G. Numerical results for mDDMt3 in a moving domain in 2D

Table 27 below shows the number of multigrid iterations needed to solve a time-independent version of the problem considered in Sec. 4 using mDDM3, where the time derivative in the diffusion equation is dropped and the corresponding Poisson problem is solved in the $t = 0$ domain. The number of multigrid iterations needed to achieve a tolerance of 10^{-10} is independent of minimum grid size h_{fine} and weakly dependent on ϵ. Fewer iterations are required for the time-dependent problem using mDDMt3 (e.g., number of iterations decreases as the time step decreases) but the dependence on h and ϵ is very similar. The results presented in Sec. 4 correspond to using h and ϵ along the diagonal entries in the table, e.g., $(\epsilon, h) = (0.2, 1/32)$, $(0.1, 1/64)$, $(0.05, 1/128)$, $(0.025, 1/256)$, $(0.0125, 1/512)$.

66

ϵ	h_{fine}=1/32	$h_{fine} = 1/64$	$h_{fine} = 1/128$	$h_{fine} = 1/256$	$h_{fine} = 1/512$
0.2	9	9	9	9	9
0.1	10	10	10	10	10
0.05	12	13	13	13	13
0.025	13	14	15	15	15
0.0125	14	15	16	16	17

Table 27: The number of multigrid iterations needed to achieve a tolerance of 10^{-10} for a time-independent problem analogous to that presented in Sec. 4. See text. The number of iterations is independent of h and weakly dependent on ϵ.

Table 28 below shows the values of the L^2 and L^∞ errors and the order of convergence **k** with $h_{fine} = \epsilon/6.4$ for mDDMt3 applied to the 2D problem in a moving domain described in Sec. 4.

ϵ	L^2 **error**	**k**	L^∞ **error**	**k**
0.2	8.89E-02	0.00	1.01E-01	0.00
0.1	2.18E-02	2.03	2.91E-02	1.80
0.05	4.91E-03	2.15	8.35E-03	1.80
0.025	1.13E-03	2.12	2.37E-03	1.82
0.0125	2.90E-04	1.96	6.78E-04	1.81

Table 28: The L^2 and L^∞ errors for simulating the 2D time-dependent diffusion equation at t=0.1 using mDDMt3 on the moving domain $D(t)$.

References

[1] L. Chen, H. Wei, M. Wen, An interface-fitted mesh generator and virtual element methods for elliptic interface problems, JOURNAL OF COMPUTATIONAL PHYSICS 334 (2017) 327–348. `doi:{10.1016/j.jcp.2017.01.004}`.

[2] P. Galenko, D. Alexandrov, E. Titova, The boundary integral theory for

slow and rapid curved solid/liquid interfaces propagating into binary systems, Phil. Trans. Roy. Soc. A 376 (2113) (2018) 20170218.

[3] M. Feischl, T. Fuehrer, N. Heuer, M. Karkulik, D. Praetorius, Adaptive boundary element methods a posteriori error estimators, adaptivity, convergence, and implementation, Arch. Comp. Meth. Eng. 22 (3) (2015) 309–389.

[4] T. Askham, A. J. Cerfon, An adaptive fast multipole accelerated Poisson solver for complex geometries, JOURNAL OF COMPUTATIONAL PHYSICS 344 (2017) 1–22. `doi:{10.1016/j.jcp.2017.04.063}`.

[5] R. GLOWINSKI, T. PAN, J. PERIAUX, A FICTITIOUS DOMAIN METHOD FOR DIRICHLET PROBLEM AND APPLICATIONS, COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING 111 (3-4) (1994) 283–303. `doi:{10.1016/0045-7825(94)90135-X}`.

[6] A. Massing, M. G. Larson, A. Logg, M. E. Rognes, A Stabilized Nitsche Fictitious Domain Method for the Stokes Problem, JOURNAL OF SCIENTIFIC COMPUTING 61 (3) (2014) 604–628. `doi:{10.1007/s10915-014-9838-9}`.

[7] S. Gallier, E. Lemaire, L. Lobry, F. Peters, A fictitious domain approach for the simulation of dense suspensions, JOURNAL OF COMPUTATIONAL PHYSICS 256 (2014) 367–387. `doi:{10.1016/j.jcp.2013.09.015}`.

[8] N. Noormohammadi, B. Boroomand, A fictitious domain method using equilibrated basis functions for harmonic and bi-harmonic problems in physics, JOURNAL OF COMPUTATIONAL PHYSICS 272 (2014) 189–217. `doi:{10.1016/j.jcp.2014.04.011}`.

[9] R. Mittal, G. Iaccarino, Immersed boundary methods, ANNUAL REVIEW OF FLUID MECHANICS 37 (2005) 239–261. `doi:{10.1146/annurev.fluid.37.061903.175743}`.

[10] J. Favier, A. Revell, A. Pinelli, A Lattice Boltzmann-Immersed Boundary method to simulate the fluid interaction with moving and slender flexible objects, JOURNAL OF COMPUTATIONAL PHYSICS 261 (2014) 145–161. `doi:{10.1016/j.jcp.2013.12.052}`.

[11] A. Calderer, S. Kang, F. Sotiropoulos, Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures, JOURNAL OF COMPUTATIONAL PHYSICS 277 (2014) 201–227. `doi:{10.1016/j.jcp.2014.08.010}`.

[12] D. B. Stein, R. D. Guy, B. Thomases, Immersed Boundary Smooth Extension (IBSE): A high-order method for solving incompressible flows in arbitrary smooth domains, JOURNAL OF COMPUTATIONAL PHYSICS 335 (2017) 155–178. `doi:{10.1016/j.jcp.2017.01.010}`.

[13] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y. Jan, A front-tracking method for the computations of multiphase flow, JOURNAL OF COMPUTATIONAL PHYSICS 169 (2) (2001) 708–759. `doi:{10.1006/jcph.2001.6726}`.

[14] M. Irfan, M. Muradoglu, A front tracking method for direct numerical simulation of evaporation process in a multiphase system, JOURNAL OF COMPUTATIONAL PHYSICS 337 (2017) 132–153. `doi:{10.1016/j.jcp.2017.02.036}`.

[15] H. Shahin, S. Mortazavi, Three-dimensional simulation of microdroplet formation in a co-flowing immiscible fluid system using front tracking method, JOURNAL OF MOLECULAR LIQUIDS 243 (2017) 737–749. `doi:{10.1016/j.molliq.2017.08.082}`.

[16] C. Hirt, A. Amsden, J. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds (Reprinted from the Journal of Computational Physics, vol 14, pg 227-253, 1974), JOURNAL OF COMPUTATIONAL PHYSICS 135 (2) (1997) 203–216. `doi:{10.1006/jcph.1997.5702}`.

[17] A. J. Barlow, P.-H. Maire, W. J. Rider, R. N. Rieben, M. J. Shashkov, Arbitrary Lagrangian-Eulerian methods for modeling high-speed compressible multimaterial flows, JOURNAL OF COMPUTATIONAL PHYSICS 322 (2016) 603–665. `doi:{10.1016/j.jcp.2016.07.001}`.

[18] W. Bo, M. Shashkov, Adaptive reconnection-based arbitrary Lagrangian Eulerian method, JOURNAL OF COMPUTATIONAL PHYSICS 299 (2015) 902–939. `doi:{10.1016/j.jcp.2015.07.032}`.

[19] C. Klingenberg, G. Schnucke, Y. Xia, An Arbitrary Lagrangian-Eulerian Local Discontinuous Galerkin Method for Hamilton-Jacobi Equations, JOURNAL OF SCIENTIFIC COMPUTING 73 (2-3, SI) (2017) 906–942. `doi:{10.1007/s10915-017-0471-2}`.

[20] R. LEVEQUE, Z. LI, THE IMMERSED INTERFACE METHOD FOR ELLIPTIC-EQUATIONS WITH DISCONTINUOUS COEFFICIENTS AND SINGULAR SOURCES, SIAM JOURNAL ON NUMERICAL ANALYSIS 31 (4) (1994) 1019–1044. `doi:{10.1137/0731054}`.

[21] R. Hu, Z. Li, Error analysis of the immersed interface method for Stokes equations with an interface, APPLIED MATHEMATICS LETTERS 83 (2018) 207–211. `doi:{10.1016/j.aml.2018.03.034}`.

[22] S. Amat, Z. Li, J. Ruiz, On an New Algorithm for Function Approximation with Full Accuracy in the Presence of Discontinuities Based on the Immersed Interface Method, JOURNAL OF SCIENTIFIC COMPUTING 75 (3) (2018) 1500–1534. `doi:{10.1007/s10915-017-0596-3}`.

[23] Z. Li, M.-C. Lai, X. Peng, Z. Zhang, A least squares augmented immersed interface method for solving Navier-Stokes and Darcy coupling equations, COMPUTERS & FLUIDS 167 (2018) 384–399. `doi:{10.1016/j.compfluid.2018.03.032}`.

[24] Z. Li, P. Song, Adaptive mesh refinement techniques for the immersed

interface method applied to flow problems, Computers & structures 122 (2013) 249–258.

[25] E. Kolahdouz, A. Pal Singh Bhalla, C. B.A., B. Griffith, An immersed interface method for discrete surfaces, J. Comput. Phys. 400 (2020) 108854.

[26] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), JOURNAL OF COMPUTATIONAL PHYSICS 152 (2) (1999) 457–492. doi:{10.1006/jcph.1999.6236}.

[27] B. Lalanne, L. R. Villegas, S. Tanguy, F. Risso, On the computation of viscous terms for incompressible two-phase flows with Level Set/Ghost Fluid Method, JOURNAL OF COMPUTATIONAL PHYSICS 301 (2015) 289–307. doi:{10.1016/j.jcp.2015.08.036}.

[28] L. R. Villegas, R. Alis, M. Lepilliez, S. Tanguy, A Ghost Fluid/Level Set Method for boiling flows and liquid evaporation: Application to the Leidenfrost effect, JOURNAL OF COMPUTATIONAL PHYSICS 316 (2016) 789–813. doi:{10.1016/j.jcp.2016.04.031}.

[29] Z. Ge, J.-C. Loiseau, O. Tammisola, L. Brandt, An efficient mass-preserving interface-correction level set/ghost fluid method for droplet suspensions under depletion forces, JOURNAL OF COMPUTATIONAL PHYSICS 353 (2018) 435–459. doi:{10.1016/j.jcp.2017.10.046}.

[30] D. Ingram, D. Causon, C. Mingham, Developments in Cartesian cut cell methods, MATHEMATICS AND COMPUTERS IN SIMULATION 61 (3-6) (2003) 561–572, 2nd IMACS Conference on Mathematical Modelling and Computational Methods in Mechanics, Physics, Biomechanics and Geodynamics, PLZEN, CZECH REPUBLIC, JUN 19-25, 2001. doi:{10.1016/S0378-4754(02)00107-6}.

[31] N. Gokhale, N. Nikiforakis, R. Klein, A dimensionally split Cartesian cut cell method for hyperbolic conservation laws, JOURNAL OF COMPUTA-

TIONAL PHYSICS 364 (2018) 186–208. `doi:{10.1016/j.jcp.2018.03.005}`.

[32] F. Nikfarjam, Y. Cheny, O. Botella, The LS-STAG immersed boundary/cut-cell method for non-Newtonian flows in 3D extruded geometries, COMPUTER PHYSICS COMMUNICATIONS 226 (2018) 67–80. `doi:{10.1016/j.cpc.2018.01.006}`.

[33] B. Muralidharan, S. Menon, Simulation of moving boundaries interacting with compressible reacting flows using a second-order adaptive Cartesian cut-cell method, JOURNAL OF COMPUTATIONAL PHYSICS 357 (2018) 230–262. `doi:{10.1016/j.jcp.2017.12.030}`.

[34] R. I. Saye, J. A. Sethian, Analysis and applications of the Voronoi Implicit Interface Method, JOURNAL OF COMPUTATIONAL PHYSICS 231 (18) (2012) 6051–6085. `doi:{10.1016/j.jcp.2012.04.004}`.

[35] A. Guittet, M. Lepilliez, S. Tanguy, F. Gibou, Solving elliptic problems with discontinuities on irregular domains - the Voronoi Interface Method, JOURNAL OF COMPUTATIONAL PHYSICS 298 (2015) 747–765. `doi:{10.1016/j.jcp.2015.06.026}`.

[36] T.-P. Fries, T. Belytschko, The extended/generalized finite element method: An overview of the method and its applications, INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING 84 (3) (2010) 253–304. `doi:{10.1002/nme.2914}`.

[37] T. Belytschko, R. Gracie, G. Ventura, A review of extended/generalized finite element methods for material modeling, MODELLING AND SIMULATION IN MATERIALS SCIENCE AND ENGINEERING 17 (4). `doi:{10.1088/0965-0393/17/4/043001}`.

[38] J. Carlos Martinez, L. V. Vanegas Useche, M. A. Wahab, Numerical prediction of fretting fatigue crack trajectory in a railway axle using

XFEM, INTERNATIONAL JOURNAL OF FATIGUE 100 (1) (2017) 32–49. `doi:{10.1016/j.ijfatigue.2017.03.009}`.

[39] J. Bedrossian, J. H. Von Brecht, S. Zhu, E. Sifakis, J. M. Teran, A second order virtual node method for elliptic problems with interfaces and irregular domains, Journal of Computational Physics 229 (18) (2010) 6405–6426.

[40] J. L. Hellrung, L. Wang, E. Sifakis, J. M. Teran, A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions, Journal of Computational Physics 231 (4) (2012) 2015 – 2048. `doi:https://doi.org/10.1016/j.jcp.2011.11.023`.
URL `http://www.sciencedirect.com/science/article/pii/S0021999111006784`

[41] D. Shirokoff, J.-C. Nave, A sharp-interface active penalty method for the incompressible navier–stokes equations, Journal of Scientific Computing 62 (1) (2015) 53–77.

[42] J. Kockelkoren, H. Levine, W.-J. Rappel, Computational approach for modeling intra- and computational approach for modeling intra- and extracellular dynamics, Phys. Rev. E 68 (2003) 037702.

[43] A. Bueno-Orovio, V. Perez-Garcia, Spectral smoothed boundary methods: The role of external boundary conditions, Numer. Meth. Partial Diff. Eqns. 22 (2005) 435–448.

[44] A. Bueno-Orovio, V. Perez-Garcia, F. Fenton, Spectral methods for partial differential equations in irregular domains: The spectral smoothed boundary method, SIAM J. Sci. Comput. 28 (2006) 886–900.

[45] A. Raetz, A. Voigt, Pde's on surfaces — a diffuse interface approach, Comm. Math. Sci. 4 (3) (2006) 575–590.

[46] X. Li, J. Lowengrub, A. Raetz, A. Voigt, Solving PDEs in complex geometries: a diffuse domain approach, COMMUNICATIONS IN MATHEMATICAL SCIENCES 7 (1) (2009) 81–107.

[47] K. E. Teigen, X. Li, J. Lowengrub, F. Wang, A. Voigt, A diffuse-interface approach for modeling transport, diffusion and adsorption/desorption of material quantities on a deformable interface, COMMUNICATIONS IN MATHEMATICAL SCIENCES 7 (4) (2009) 1009–1037.

[48] H.-C. Yu, H.-Y. Chen, K. Thornton, Extended smoothed boundary method for solving partial differential equations with general boundary conditions on complex boundaries, MODELLING AND SIMULATION IN MATERIALS SCIENCE AND ENGINEERING 20 (2012) 075008.

[49] S. Poulsen, P. Voorhees, Smoothed boundary method for diffusion-related partial differential equations in complex geometries, Int. J. Comput. Meth. 15 (3) (2018) 1850014.

[50] F. Fenton, E. Cherry, A. Karma, W.-J. Rappel, Modeling wave propagation in realistic heart geometries using the phase-field method, Chaos 15 (2005) 013502.

[51] S. Aland, C. Landsberg, R. Mueller, F. Stenger, M. Bobeth, A. Langheinrich, A. Voigt, Adaptive diffuse domain approach for calculating mechanically induced deformation of trabecular bone, Comp. Meth. Biomech. Biomed. Eng. 17 (1) (2011) 31–38.

[52] B. Camley, Y. Zhao, B. Li, H. Levine, W.-J. Rappel, Periodic migration in a physical model of cells on micropatterns, Phys. Rev. Lett. 111 (2013) 158102.

[53] Y. Chen, J. Lowengrub, Tumor growth in complex, evolving microenvironmental geometries: A diffuse domain approach, J. Theor. Biol. 361 (2014) 14–30.

[54] A. Raetz, M. Roeger, Symmetry breaking in a bulk-surface reaction-diffusion model for signaling networks, Nonlinearity 27 (2014) 1805–1827.

[55] A. Raetz, A new diffuse-interface model for step flow in epitaxial growth, IMA J. Appl. Math. 80 (3) (2015) 697–711.

[56] J. Lowengrub, J. Allard, S. Aland, Numerical simulation of endocytosis: Viscous flow driven by membranes with non-uniformly distributed curvature-inducing molecules, J. Comput. Phys. 309 (2016) 112–128.

[57] B. Camley, Y. Zhao, B. Li, H. Levine, W.-J. Rappel, Crawling and turning in a minimal reaction-diffusion cell motility model: Coupling cell shape and biochemistry, Phys. Rev. E 95 (2017) 012401.

[58] J. Lipkova, Computational modelling in neuro-onconolgy, Ph.D. Thesis, T.U. Munich.

[59] K. E. Teigen, P. Song, J. Lowengrub, A. Voigt, A diffuse-interface method for two-phase flows with soluble surfactants, JOURNAL OF COMPUTATIONAL PHYSICS 230 (2) (2011) 375–393. doi:{10.1016/j.jcp.2010.09.020}.

[60] S. Aland, J. Lowengrub, A. Voigt, Two-phase flow in complex geometries: A diffuse domain approach, CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES 57 (1) (2010) 77–107.

[61] S. Aland, J. Lowengrub, A. Voigt, A continuum model of colloid-stabilized interfaces, Phys. Fluids 23 (6) (2011) 062103.

[62] S. Aland, J. Lowengrub, A. Voigt, Particles at fluid-fluid interfaces: A new navier-stokes-cahn-hilliard surface-phase-field-crystal model, Phys. Rev. E 86 (4) (2012) 046321.

[63] S. Aland, S. Egerer, J. Lowengrub, A. Voigt, Diffuse interface models of locally inextensible vesicles in a viscous fluid, J. Comput. Phys. 277 (2014) 32–47.

[64] H.-C. Yu, M.-J. Choe, G. Amatucci, Y.-M. Chiang, K. Thornton, Smoothed boundary method for simulating bulk and grain boundary transport in complex polycrystalline microstructures, Comput. Mater. Sci. 121 (2016) 14–22.

[65] A. Raetz, Diffuse-interface approximations of osmosis free boundary problems diffuse-interface approximations of osmosis free boundary problems, SIAM J. Appl. Math. 76 (3) (2016) 910–929.

[66] L. Hong, L. Liang, S. Bhattacharyya, W. Xing, L.-Q. Chen, Anisotropic li intercalation in a lixfepo4 nano-particle: a spectral smoothed boundary phase-field model, Phys, Chem. Chem. Phys. 18 (2016) 9537–9543.

[67] A. Chadwick, J. Steward, R. Du, K. Thornton, Numerical modeling of localized corrosion using phase-field and smoothed boundary methods, J. Electrochemical Soc. 165 (10) (2018) C633–C646.

[68] S. Vey, A. Voigt, Amdis: Adaptive multidimensional simulations, Comput. Vis. Sci. 10 (2007) 57–67.

[69] D. Rossinelli, B. Hejazialhosseini, W. van Rees, M. Gazzola, M. Bergdorf, P. Koumoutsakos, Mrag-i2d: Multi-resolution adapted grids for remeshed vortex methods on multicore architectures, J. Comput. Phys. 288 (2015) 1–18.

[70] W. Feng, Z. Guo, J. Lowengrub, S. Wise, Mass-conservative adaptive fas multigrid solver for cell-centered finite difference methods on block-structured, locally-cartesian grids, J. Comput. Phys. 352 (2018) 463–497.

[71] S. Franz, H.-G. Roos, R. Gärtner, A. Voigt, A note on the convergence analysis of a diffuse-domain approach, COMPUTATIONAL METHODS IN APPLIED MATHEMATICS 12 (2) (2012) 153–167.

[72] H. Abels, K. Lam, B. Stinner, Analysis of the diffuse domain approach for a bulk-surface coupled pde system, SIAM J. Math. Anal. 47 (5) (2015) 3687–3725.

[73] M. Schlottbom, Error analysis of a diffuse interface method for elliptic problems with dirichlet boundary conditions, Appl. Num. Math. 109 (2016) 109–122.

[74] M. Burger, O. Elvetun, M. Schlottbom, Analysis of the diffuse domain method for second order elliptic boundary value problems, Found. Comput. Math. 17 (2017) 627–674.

[75] K. Y. Lervåg, J. Lowengrub, Analysis of the diffuse-domain method for solving PDEs in complex geometries, COMMUNICATIONS IN MATHEMATICAL SCIENCES 13 (6) (2015) 1473–1500. `doi:{10.4310/CMS.2015.v13.n6.a6}`.

[76] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed-algorithms based on hamilton-jacobi formulations, J. Comput. Phys. 79 (1) (1988) 12–49.

[77] F. Gibou, R. Fedkiw, S. Osher, A review of level-set methods and some recent applications, J. Comput. Phys. 353 (2018) 82–109.

[78] J. D. Hoffman, S. Frankel, Numerical methods for engineers and scientists, CRC press, 2001.

[79] C. M. Bender, S. A. Orszag, Advanced mathematical methods for scientists and engineers I: Asymptotic methods and perturbation theory, Springer Science & Business Media, 2013.

[80] G. Jiang, C. Shu, Efficient implementation of weighted ENO schemes, JOURNAL OF COMPUTATIONAL PHYSICS 126 (1) (1996) 202–228. `doi:{10.1006/jcph.1996.0130}`.

[81] S. Gottlieb, C. Shu, Total variation diminishing Runge-Kutta schemes, MATHEMATICS OF COMPUTATION 67 (221) (1998) 73–85. `doi:{10.1090/S0025-5718-98-00913-2}`.

[82] C. Min, On reinitializing level set functions, JOURNAL OF COMPUTATIONAL PHYSICS 229 (8) (2010) 2764–2772. `doi:{10.1016/j.jcp.2009.12.032}`.

[83] G. Russo, P. Smereka, A remark on computing distance functions, JOURNAL OF COMPUTATIONAL PHYSICS 163 (1) (2000) 51–67. `doi:{10.1006/jcph.2000.6553}`.

[84] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-based fast local level set method, JOURNAL OF COMPUTATIONAL PHYSICS 155 (2) (1999) 410–438. `doi:{10.1006/jcph.1999.6345}`.

[85] S. Wise, J. Kim, J. Lowengrub, Solving the regularized, strongly anisotropic Cahn-Hilliard equation by an adaptive nonlinear multigrid method, JOURNAL OF COMPUTATIONAL PHYSICS 226 (1) (2007) 414–446. `doi:{10.1016/j.jcp.2007.04.020}`.

[86] Icbm 152 nonlinear atlases version 2009, bic.mni.mcgill.ca.

[87] D. Collins, A. Zijdenbos, W. Baare, A. Evans, Animal+insect: Improved cortical structure segmentation, IPMI Lect. Notes Comput. Sci. 1613/1999 (1999) 210–223.

[88] V. Fonov, A. Evans, K. Botteron, R. McKinstry, C. Almli, D. Collins, Unbiased nonlinear average age-appropriate brain templates from birth to adulthood, NeuroImage 47 (2009) S102, organization for Human Brain Mapping 2009 Annual Meeting.

[89] V. Fonov, A. Evans, K. Botteron, C. Almli, R. McKinstry, D. Collins, BDCG, Unbiased nonlinear average age-appropriate atlases for pediatric studies, Comput. Vis. Sci. 54 (2011) 1053–8119.

[90] K. Lervag, J. Lowengrub, Analysis of the diffuse-domain method for solving pdes in complex geometries, Commun. Math. Sci. 6 (2015) 1473–1500.