

# High-Order Numerical Methods for 2D Parabolic Problems in Single and Composite Domains

Gustav Ludvigsson<sup>1</sup> · Kyle R. Steffen<sup>2</sup> · Simon Sticko<sup>1</sup> · Siyang Wang<sup>3</sup> · Qing Xia<sup>2</sup> · Yekaterina Epshteyn<sup>2</sup> · Gunilla Kreiss<sup>1</sup>

Received: 26 July 2017 / Revised: 21 November 2017 / Accepted: 23 December 2017 /  
Published online: 10 January 2018  
© The Author(s) 2018. This article is an open access publication

**Abstract** In this work, we discuss and compare three methods for the numerical approximation of constant- and variable-coefficient diffusion equations in both single and composite domains with possible discontinuity in the solution/flux at interfaces, considering (i) the Cut Finite Element Method; (ii) the Difference Potentials Method; and (iii) the summation-by-parts Finite Difference Method. First we give a brief introduction for each of the three methods. Next, we propose benchmark problems, and consider numerical tests—with respect to accuracy and convergence—for linear parabolic problems on a single domain, and continue with similar tests for linear parabolic problems on a composite domain (with the interface defined either explicitly or implicitly). Lastly, a comparative discussion of the methods and numerical results will be given.

---

✉ Gustav Ludvigsson  
gustav.ludvigsson@it.uu.se

Kyle R. Steffen  
steffen@math.utah.edu

Simon Sticko  
simon.sticko@it.uu.se

Siyang Wang  
siyang.wang@chalmers.se

Qing Xia  
xia@math.utah.edu

Yekaterina Epshteyn  
epshteyn@math.utah.edu

Gunilla Kreiss  
gunilla.kreiss@it.uu.se

<sup>1</sup> Department of Information Technology, Uppsala University, Box 337, 751 05 Uppsala, Sweden

<sup>2</sup> Department of Mathematics, The University of Utah, 155 S 1400 E Rm. 233, Salt Lake City, UT 84112, USA

<sup>3</sup> Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, 412 96 Göthenburg, Sweden

**Keywords** Parabolic problems · Interface models · Level set · Complex geometry · Discontinuous solutions · SBP–SAT finite difference · Difference potentials · Spectral approach · Finite element method · Cut elements · Immersed boundary · Stabilization · Higher order accuracy and convergence

**Mathematics Subject Classification** 65M06 · 65M12 · 65M22 · 65M55 · 65M60 · 65M70 · 35K20

## 1 Introduction

Designing methods for the high-order accurate numerical approximation of partial differential equations (PDE) posed on composite domains with interfaces, or on irregular and geometrically complex domains, is crucial in the modeling and analysis of problems from science and engineering. Such problems may arise, for example, in materials science (models for the evolution of grain boundaries in polycrystalline materials), fluid dynamics (the simulation of homogeneous or multi-phase fluids), engineering (wave propagation in an irregular medium or a composite medium with different material properties), biology (models of blood flow or the cardiac action potential), etc. The analytic solutions of the underlying PDE may have non-smooth or even discontinuous features, particularly at material interfaces or at interfaces within a composite medium. Standard numerical techniques involving finite-difference approximations, finite-element approximation, etc., may fail to produce an accurate approximation near the interface, leading one to consider and develop new techniques.

There is extensive existing work addressing numerical approximation of PDE posed on composite domains with interfaces or irregular domains, for example, the boundary integral method [11,56], difference potentials method [3,6,26,27,58,65], immersed boundary method [30,42,61,74], immersed interface method [2,46,47,49,69], ghost fluid method [31,32,50,51], the matched interface and boundary method [82,84–86], Cartesian grid embedded boundary method [19,41,57,83], multigrid method for elliptic problems with discontinuous coefficients on an arbitrary interface [18], virtual node method [9,39], Voronoi interface method [35,36], the finite difference method [8,10,24,75,78,79] and finite volume method [22,34] based on mapped grids, or cut finite element method [13–15,37,38,71,76]. Indeed, there have been great advances in numerical methods for the approximation of PDE posed on composite domains with interfaces, or on irregular domains. However, it is still a challenge to design high-order accurate and computationally-efficient methods for PDE posed in these complicated geometries, especially for time-dependent problems, problems with variable coefficients, or problems with general boundary/interface conditions.

The aim of this work is to establish benchmark (test) problems for the numerical approximation of parabolic PDE defined in irregular or composite domains. The considered models (Sect. 2) arise in the study of mass or heat diffusion in single or composite materials, or as simplified models in other areas (e.g., biology, materials science, etc.). The formulated test problems (Sect. 4) are intended (a) to be suitable for comparison of high-order accurate numerical methods—and will be used as such in this study—and (b) to be useful in further research. Moreover, the proposed problems include a wide variety of possibilities relevant in applications, which any robust numerical method should resolve accurately, including constant diffusion; time-varying diffusion; high frequency oscillations in the analytical solution; large jumps in diffusion coefficients, solution, and/or flux; etc. For now, we will consider a simplified geometrical setting, with the intent of setting a “baseline” from which further

research, or more involved comparisons, might be conducted. Therefore, in Sect. 2 we will introduce two circular geometries, which are defined either explicitly, or implicitly via a level set function.

In Sect. 3, we briefly introduce the numerical methods we will consider in this work, i.e., second- and fourth-order versions of (i) the Cut Finite Element Method (cut-FEM); (ii) the Difference Potentials Method (DPM), with Finite Difference approximation as the underlying discretization in the current work; and (iii) the summation-by-parts Finite Difference Method combined with the simultaneous approximation term technique (SBP–SAT–FD). These three methods are all modern numerical methods which may be designed for problems in irregular or composite domains, allowing for high-order accurate numerical approximation, even at points close to irregular interfaces or boundaries. We will apply each method to the formulated benchmark problems, and compare results. From the comparisons, we expect to learn what further developments of the methods at hand would be most important.

To resolve geometrical features of irregular domains, both cut-FEM and DPM use a Cartesian grid on top of the domain, which need not conform with boundaries or interfaces. These types of methods are often characterized as “immersed” or “embedded”. In the finite difference framework, embedded methods for parabolic problems are developed in [1,23]. For comparison with cut-FEM and DPM, however, in this paper we use a finite difference method based on a conforming approach. The finite difference operators we use satisfy a summation-by-parts principle. Then, in combination with the SAT method to weakly impose boundary and interface conditions, an energy estimate of the semi-discretization can be derived to ensure stability. In addition, we use curvilinear grids and transfinite interpolation to resolve complex geometries.

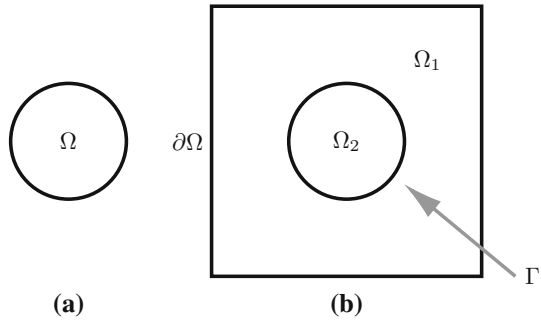
For recent work on SBP–SAT–FD for wave equations in composite domains, see [10,17,75,78], and the two review papers [21,73]; for recent work in DPM for elliptic/parabolic problems in composite domains with interface defined explicitly, see [3–6,26–28,58,59,65]; and for recent work in cut-FEM, see [13–15,37,53,71].

The paper is outlined as follows. In Sect. 2, we give brief overview of the continuous formulation of the parabolic problems in a single domain or a composite domain. In Sect. 3, we give introductions to the basics of the three proposed methods: cut-FEM, DPM, and SBP–SAT–FD. In Sect. 4, we formulate the numerical test problems. In Sect. 5, we present extensive numerical comparisons of errors and convergence rates, between the second- and fourth-order versions of each method. The comparisons include single domain problems with constant or time-dependent diffusivity; and interface problems with interface defined explicitly, or implicitly by a level set function. In Sect. 6, we give a comparative discussion of the three methods and the numerical results, together with a discussion on future research directions. Lastly, in Sect. 7, we give our concluding remarks.

## 2 Statement of Problem

In this section, we describe two diffusion problems, which will be the setting for our proposed benchmark (test) problems in Sect. 4. (Recall from Sect. 1 that these models arise, for example, in the study of mass or heat diffusion.) For brevity, in the following discussion, we denote  $u := u(x, y, t)$  and  $u_s := u_s(x, y, t)$ , with  $s = 1, 2$ .

**Fig. 1** The (a) single domain  $\Omega$  and (b) composite domain  $\Omega = \Omega_1 \cup \Omega_2$ . In (b),  $\partial\Omega_1$  has two connected components: the boundary  $\partial\Omega$  and interface  $\Gamma = \partial\Omega_2$



### 2.1 The Single Domain Problem

First, we consider the linear parabolic PDE on a single domain  $\Omega$  (e.g., Fig. 1a), with variable diffusion  $\lambda(t)$ :

$$\frac{\partial u}{\partial t} = \nabla \cdot (\lambda(t)\nabla u) + f(x, y, t), \quad (x, y, t) \in \Omega \times (0, T], \tag{1}$$

subject to initial and Dirichlet boundary conditions:

$$u(x, y, 0) = u^0(x, y), \quad (x, y) \in \Omega \quad \text{and} \quad u = \psi(x, y, t), \quad (x, y, t) \in \partial\Omega \times (0, T]. \tag{2}$$

Here, the initial and boundary data  $u^0(x, y)$  and  $\psi(x, y, t)$ , the diffusion coefficient  $\lambda(t)$ , the forcing function  $f(x, y, t)$ , and the final time  $T$  are known (given) data.

### 2.2 The Composite Domain Problem

Next, we consider the linear parabolic PDE on a composite domain  $\Omega := \Omega_1 \cup \Omega_2$  (e.g., Fig. 1b), with constant diffusion coefficients  $(\lambda_1, \lambda_2)$ :

$$\frac{\partial u_1}{\partial t} = \nabla \cdot (\lambda_1 \nabla u_1) + f_1(x, y, t), \quad (x, y, t) \in \Omega_1 \times (0, T], \tag{3}$$

$$\frac{\partial u_2}{\partial t} = \nabla \cdot (\lambda_2 \nabla u_2) + f_2(x, y, t), \quad (x, y, t) \in \Omega_2 \times (0, T], \tag{4}$$

subject to initial conditions:

$$u_1(x, y, 0) = u_1^0(x, y), \quad (x, y) \in \Omega_1, \tag{5}$$

$$u_2(x, y, 0) = u_2^0(x, y), \quad (x, y) \in \Omega_2, \tag{6}$$

Dirichlet boundary conditions:

$$u_1 = \psi(x, y, t), \quad (x, y, t) \in \partial\Omega \times (0, T], \tag{7}$$

and interface/matching conditions:

$$u_1 - u_2 = \mu_1(x, y, t), \quad (x, y, t) \in \Gamma \times (0, T], \tag{8}$$

$$\lambda_1 \frac{\partial u_1}{\partial n} - \lambda_2 \frac{\partial u_2}{\partial n} = \mu_2(x, y, t), \quad (x, y, t) \in \Gamma \times (0, T]. \tag{9}$$

In formula (9),  $\frac{\partial u_s}{\partial n}$ ,  $s = 1, 2$  denotes the normal derivative at the interface  $\Gamma$ , i.e.,  $\frac{\partial u_s}{\partial n} = \nabla u_s \cdot \mathbf{n}$ , where  $\mathbf{n}$  is the outward unit normal vector at the interface  $\Gamma$ . The initial, boundary,

and interface data  $u_1^0(x, y)$ ,  $u_2^0(x, y)$ ,  $\psi(x, y, t)$ ,  $\mu_1(x, y, t)$ , and  $\mu_2(x, y, t)$ ; the diffusion coefficients  $(\lambda_1, \lambda_2)$ ; the forcing functions  $f_1(x, y, t)$  and  $f_2(x, y, t)$ ; and the final time  $T$  are some known (given) data.

*Remark 1* We consider the circular geometries depicted in Fig. 1 as the geometrical setting for our proposed benchmark problems in this work. In applications (Sect. 1), other geometries will likely be considered, some much more complicated than Fig. 1. While our methods can handle more complicated geometry, this is (to the best of our knowledge) the first work looking to establish benchmarks—and compare numerical methods—for parabolic interface problems (3–9). As such, we think that the geometries in Fig. 1 are a good “baseline”—without all the added complexities that more complicated geometries might produce—from which further research, or more involved comparisons, might be done.

To be more specific, we aim to define a simple set of test problems that can be easily implemented and tested for any numerical scheme of interest. With circular domains, it suffices for us to compare/contrast performance of the numerical methods on a simple geometry with smooth boundary versus on a composite domain with fixed interface (explicit or implicit). The approximation of the solution to such composite-domain problems are already challenging for any numerical methods, since (i) the solution may fail to be smooth (or may be discontinuous) at the interface, and (ii) there may be discontinuous material coefficients  $(\lambda_1 \neq \lambda_2)$ .

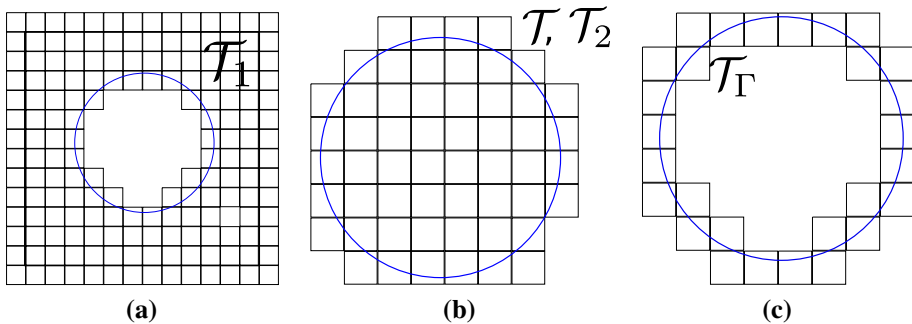
*Remark 2* For both the single and composite domain problems, we could also consider other boundary conditions, e.g., a Neumann boundary condition as in [6, 13], etc.

### 3 Overview of Numerical Methods

#### 3.1 Cut-FEM

In this section, we give a brief presentation of the cut-FEM method. For a more detailed presentation of cut-FEM, see, for example, [13, 14, 53].

Let  $\Omega_s$  be covered by a structured triangulation,  $\mathcal{T}_s$ , so that each element  $T \in \mathcal{T}_s$  has some part inside of  $\Omega_s$ ; see Fig. 2a and b. Here,  $s = 1, 2$  is an index for the composite domain problem (3–9), which will be omitted when referring to the single domain problem (1, 2).



**Fig. 2** The (a) subdomain  $\Omega_1$  immersed in a mesh  $\mathcal{T}_1$ , (b) subdomain  $\Omega_2$  immersed in a mesh  $\mathcal{T}_2$  and domain  $\Omega$  immersed in  $\mathcal{T}$ , and (c) intersected elements  $\mathcal{T}_\Gamma$

(For the latter, note that  $\mathcal{T}$  covers  $\Omega$ .) Typically  $\mathcal{T}_1$  and  $\mathcal{T}_2$  would be created from a larger mesh by removing some of the cells. Further, let  $\mathcal{T}_\Gamma = \{T \in \mathcal{T} : T \cap \Gamma \neq \emptyset\}$  be the set of intersected elements; see Fig. 2c. In the following, we shall use  $\Gamma$  both for the immersed boundary of the single domain problem and for the immersed interface of the composite domain problem, in order to make the connection to the set  $\mathcal{T}_\Gamma$  clearer.

To construct the finite element spaces we use Lagrange elements with Gauss–Lobatto nodes of order  $p$  ( $Q_p$ -elements). Let  $V_h^s$  denote a continuous finite element space on  $\Omega_s$ , consisting of  $Q_p$ -elements on the mesh  $\mathcal{T}_s$ :

$$V_h^s = \{v \in C^0(\Omega_s) : v|_T \in Q_p(T), T \in \mathcal{T}_s\}. \tag{10}$$

For the single domain problem (1, 2) we solve for the solution  $u \in V_h$ ; while for the composite domain problem (3–9), we solve for the pair  $\{u_1, u_2\} \in V_h^1 \times V_h^2$ . For the latter problem, this means that the degrees of freedom are doubled over elements belonging to  $\mathcal{T}_\Gamma$ .

We begin by stating the weak formulation for the single domain problem (1, 2). Let  $(\cdot, \cdot)_X$  and  $\langle \cdot, \cdot \rangle_Y$  be the  $L_2$  scalar products taken over the two- and one-dimensional domains  $X \subset \mathbb{R}^2$  and  $Y \subset \mathbb{R}^1$ , respectively. The present method is based on modifying the weak formulation by using Nitsche’s method [60] to enforce the boundary condition (2). By multiplying (1) with a test function  $v \in V_h$ , and integrating by parts, we obtain:

$$(\dot{u}, v)_\Omega + (\lambda \nabla u, \nabla v)_\Omega - \left\langle \lambda \frac{\partial u}{\partial n}, v \right\rangle_\Gamma = (f, v)_\Omega, \quad \forall v \in V_h. \tag{11}$$

Note that (2) is consistent with the following terms:

$$\frac{\gamma_D}{h_T} \langle \lambda u, v \rangle_\Gamma = \frac{\gamma_D}{h_T} \langle \lambda \psi, v \rangle_\Gamma, \tag{12}$$

$$- \left\langle u, \lambda \frac{\partial v}{\partial n} \right\rangle_\Gamma = - \left\langle \psi, \lambda \frac{\partial v}{\partial n} \right\rangle_\Gamma, \tag{13}$$

where  $\gamma_D$  is a constant, and  $h_T$  is the side length of the quadrilaterals in the triangulation. Now, adding (12, 13) to (11) gives the following weak form: Find  $u \in V_h$  such that

$$(\dot{u}, v)_\Omega + a(u, v) = L(v), \quad \forall v \in V_h, \tag{14}$$

where

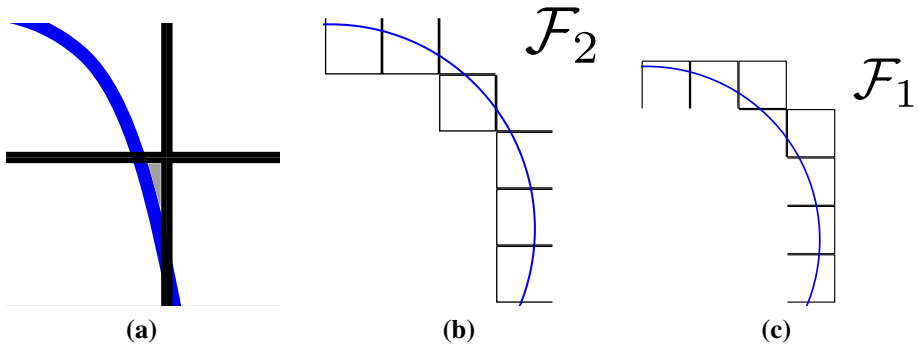
$$a(u, v) = (\lambda \nabla u, \nabla v)_\Omega - \left\langle \lambda \frac{\partial u}{\partial n}, v \right\rangle_\Gamma - \left\langle u, \lambda \frac{\partial v}{\partial n} \right\rangle_\Gamma + \frac{\gamma_D}{h_T} \langle \lambda u, v \rangle_\Gamma, \tag{15}$$

$$L(v) = (f, v)_\Omega + \left\langle \lambda \psi, \frac{\gamma_D}{h_T} v - \frac{\partial v}{\partial n} \right\rangle_\Gamma. \tag{16}$$

For  $\mathcal{T}_\Gamma$  (the elements intersected by  $\Gamma$ ), note that one must integrate only over the part of the element that lies inside  $\Omega$ . A problem with this is that one cannot control how the intersections (cuts) between  $\Omega$  and  $\mathcal{T}$  are made. Depending on how  $\Omega$  is located with respect to the triangulation, some elements can have an arbitrarily small intersection with the domain—see, for example, Fig. 3a. If  $\Omega$  is moved with respect to  $\mathcal{T}$  to make the cut arbitrarily small, then the condition numbers of the mass and stiffness matrices can become arbitrarily large.

To mitigate this issue, in this work we add a stabilizing term  $j$ —defined shortly in (19)—to the mass and stiffness matrices, so that their condition numbers are bounded, independently of how the domain  $\Omega$  is located with respect to the triangulation  $\mathcal{T}$  [14, 53]. Adding stabilization to (14) results in the following weak form: Find  $u \in V_h$  such that

$$(\dot{u}, v)_\Omega + \gamma_M j(\dot{u}, v) + a(u, v) + \gamma_A h_T^{-2} \lambda j(u, v) = L(v), \quad \forall v \in V_h, \tag{17}$$



**Fig. 3** (a) An element having a small intersection, shown in gray, with the domain; (b) faces belonging to  $\mathcal{F}_2$ ; and (c) faces belonging to  $\mathcal{F}_1$

where  $\gamma_M$  and  $\gamma_A$  are scalar constants.

In order to state the definition of stabilization (19), denote by  $\mathcal{F}_s$  the set of faces, as seen in Fig. 3b and c. That is,  $\mathcal{F}_s$  is the set of all faces of the elements in  $\mathcal{T}_\Gamma$ , excluding the boundary faces of  $\mathcal{T}_s$ :

$$\mathcal{F}_s = \{F = T_A \cap T_B : T_A \in \mathcal{T}_\Gamma \text{ or } T_B \in \mathcal{T}_\Gamma, T_A, T_B \in \mathcal{T}_s\}. \tag{18}$$

Then, the stabilization term is defined as:

$$j_s(u, v) = \sum_{F \in \mathcal{F}_s} \sum_{k=1}^p \frac{h_T^{2k+1}}{(2k+1)(k!)^2} \left( [\partial_n^k u], [\partial_n^k v] \right)_F, \tag{19}$$

where  $[u] = u|_{F_+} - u|_{F_-}$  is the jump over a face,  $F$ ;  $n$  refers to a normal of  $F$ ; and  $\partial_n^k u$  denotes the  $k$ -th order normal derivative. The scaling with respect to  $k$  of the terms in (19) is based on how the stabilization was derived. In particular, the  $k!$ -factors come from the Taylor-expansion and the factor  $2k + 1$  comes from integrating each term once.

We now consider the composite domain problem (3–9). To derive the weak formulation, one follows essentially the same steps as for the single domain problem, namely:

1. For both (3) and (4), multiply the equation for  $u_s$  with a test function  $v_s \in V_h^s$ , and then integrate by parts;
2. Add terms consistent with the interface and boundary conditions; and
3. Add stabilization terms  $j_1$  and  $j_2$  over  $\mathcal{F}_1$  and  $\mathcal{F}_2$ , respectively.

This results in the following weak formulation for (3–9). Find  $u = \{u_1, u_2\} \in V_h^1 \times V_h^2$  such that:

$$\begin{aligned} M(\dot{u}, v) + A(u, v) + a_\Gamma(u, v) + a_{\partial\Omega}(u, v) \\ = L_\Omega(v) + L_\Gamma(v) + L_{\partial\Omega}, \quad \forall v = \{v_1, v_2\} \in V_h^1 \times V_h^2, \end{aligned} \tag{20}$$

where the bilinear forms  $M$  and  $A$  correspond to the stabilized mass and stiffness matrices:

$$M(\dot{u}, v) = \sum_{s=1}^2 (\dot{u}_s, v_s)_{\Omega_s} + \gamma_M j_s(\dot{u}_s, v_s), \tag{21}$$

$$A(u, v) = \sum_{s=1}^2 (\lambda \nabla u_s, \nabla v_s)_{\Omega_s} + \gamma_A h_T^{-2} \lambda j_s(u_s, v_s); \tag{22}$$

$L_\Omega$  corresponds to the forcing function:

$$L_\Omega(v) = \sum_{s=1}^2 (f_s, v_s)_{\Omega_s}; \tag{23}$$

$a_\Gamma$  and  $L_\Gamma$  consistently enforce the interface conditions (8, 9):

$$a_\Gamma(u, v) = - \left\langle [u], \left\{ \lambda \frac{\partial v}{\partial n} \right\} \right\rangle_\Gamma - \left\langle \left\{ \lambda \frac{\partial u}{\partial n} \right\}, [v] \right\rangle_\Gamma + \left\langle \frac{\gamma_\Gamma}{h_T} [u], [v] \right\rangle_\Gamma, \tag{24}$$

$$L_\Gamma(v) = \left\langle \frac{\gamma_\Gamma}{h_T} \mu_1, [v] \right\rangle_\Gamma + \langle \kappa_1 \mu_2, v_2 \rangle_\Gamma + \langle \kappa_2 \mu_2, v_1 \rangle_\Gamma - \left\langle \mu_1, \left\{ \lambda \frac{\partial v}{\partial n} \right\} \right\rangle_\Gamma; \tag{25}$$

and the terms  $a_{\partial\Omega}$  and  $L_{\partial\Omega}$  enforce the boundary condition (7) along the outer boundary,  $\partial\Omega$ :

$$a_{\partial\Omega}(u, v) = - \left\langle \lambda \frac{\partial u_1}{\partial n}, v_1 \right\rangle_{\partial\Omega} - \left\langle u_1, \lambda \frac{\partial v_1}{\partial n} \right\rangle_{\partial\Omega} + \frac{\gamma_D}{h_T} \langle \lambda u_1, v_1 \rangle_{\partial\Omega}, \tag{26}$$

$$L_{\partial\Omega}(v) = \left\langle \lambda \psi, \frac{\gamma_D}{h_T} v_1 - \frac{\partial v_1}{\partial n} \right\rangle_{\partial\Omega}. \tag{27}$$

In (24–27),  $n$  denotes the outward pointing normal at either  $\Gamma$  or  $\partial\Omega$  (depending on the domain of integration);  $\kappa_1 + \kappa_2 = 1$ , so that  $\{v\} = \kappa_1 v_1 + \kappa_2 v_2$  is a convex combination; and  $\gamma_\Gamma, \kappa_1, \kappa_2$  are chosen as in [13]:

$$\kappa_1 = \frac{\lambda_2}{\lambda_1 + \lambda_2}, \quad \kappa_2 = \frac{\lambda_1}{\lambda_1 + \lambda_2}, \quad \gamma_\Gamma = \gamma_D \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}. \tag{28}$$

The remaining parameters (appearing in Eqs. 21, 22, 26–28) are given by:

$$\gamma_M = 0.75, \quad \gamma_A = 1.5, \quad \gamma_D = 5p^2. \tag{29}$$

The scaling of  $\gamma_D$  with respect to  $p$  follows from an inverse inequality. When  $p = 1$  these reduce to the same parameters as the ones used in [71], where  $\gamma_M$  was chosen based on numerical experiments on the condition number of the mass matrix. This also agrees with the choice of  $\gamma_A$  and  $\gamma_D$  in [14], where  $\gamma_A$  was investigated numerically.

In order to use cut-FEM, one needs a way to perform integration over the intersected elements  $\mathcal{T}_\Gamma$ . For example, with the interface problem, on each element  $K \in \mathcal{T}_\Gamma$ , we need a quadrature rule for the  $K \cap \Omega_1, K \cap \Omega_2$  and  $K \cap \Gamma$ . For the numerical tests in this work (Sect. 4), we represent the geometry by a level set function, and compute high-order accurate quadrature rules with the algorithm from [68].

*Remark 3* Optimal (second-order) convergence was rigorously proven for cut-FEM applied to the Poisson problem in [14]. As far as we know, there is no rigorous proof of higher-order convergence for cut-FEM, though such a proof would likely be similar to the second-order case.

### 3.2 DPM

We continue in this section with a brief introduction to the Difference Potentials Method (DPM), which was originally proposed by Ryaben’kii (see [64–66], and see [29,33] for papers in his honor). Our aim is to consider the numerical approximation of PDEs on arbitrary, smooth geometries (defined either explicitly or implicitly) using the DPM together with standard, finite-difference discretizations of (1) or (3, 4) on uniform, Cartesian grids, which



need not conform with boundaries or interfaces. To this end, we work with high-order methods for interface problems based on Difference Potentials, which were originally developed in [67] and [3–6,26,28]. We also introduce new developments here for handling implicitly-defined geometries. (The reader can consult [65] for the general theory of the Difference Potentials Method.)

Broadly, the main idea of the DPM is to reduce uniquely solvable and well-posed boundary value problems in a domain  $\Omega$  to pseudo-differential *Boundary Equations with Projections* (BEP) on the boundary of  $\Omega$ . First, we introduce a computationally simple auxiliary domain as part of the method. The original domain is embedded into the auxiliary domain, which is then discretized using a uniform Cartesian grid. Next, we define a Difference Potentials operator *via* the solution of a simple Auxiliary Problem (defined on the auxiliary domain), and construct the discrete, pseudo-differential *Boundary Equations with Projections* (BEP) at grid points near the continuous boundary or interface  $\Gamma$ . (This set of grid points is called the *discrete grid boundary*.) Once constructed, the BEP are then solved together with the boundary/interface conditions to obtain the value of the solution at the discrete grid boundary. Lastly, using these reconstructed values of the solution at the discrete grid boundary, the approximation to the solution in the domain  $\Omega$  is obtained through the *discrete, generalized Green’s formula*.

Mathematically, the DPM is a discrete analog of the method of Calderón’s potentials in the theory of partial differential equations. The DPM, however, does not require explicit knowledge of Green’s functions. Although we use an Auxiliary Problem (AP) discretized by finite differences, the DPM is not limited to this choice of spatial discretization. Indeed, numerical methods based on the idea of Difference Potentials can be designed with whichever choice of spatial discretization is most natural for the problem at hand (e.g., see [25]).

Practically, the main computational complexity of the DPM reduces to the required solutions of the AP, which can be done very efficiently using fast, standard  $\mathcal{O}(N \log N)$  solvers. Moreover, in general the DPM can be applied to problems with general boundary or interface conditions, with no change to the discretization of the PDE.

Let us now briefly introduce the DPM for the numerical approximation of parabolic interface models (3–9). First, we must introduce the point-sets that will be used throughout the DPM. (Note that the main construction of the method below applies to the single domain problem (1, 2), after omitting the index  $s$  and replacing interface conditions with boundary conditions; see [6].)

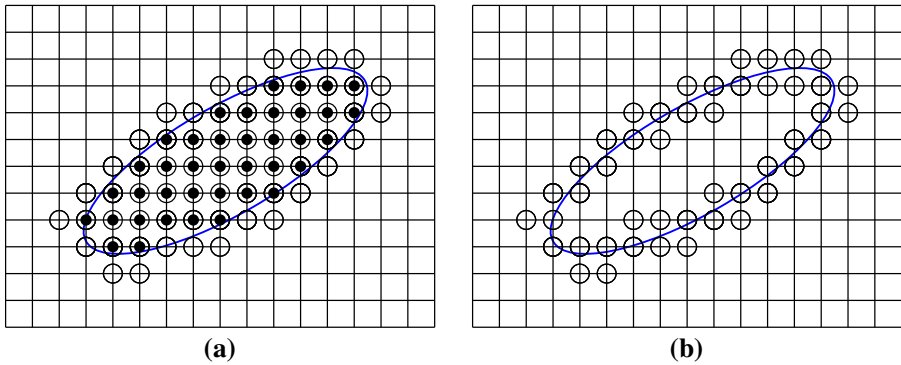
Let  $\Omega_s$  ( $s = 1, 2$ ) be embedded in a rectangular auxiliary domain  $\Omega_s^0$ . Introduce a uniform, Cartesian grid denoted  $M_s^0$  on  $\Omega_s^0$ , with grid-spacing  $h_s$ . Let  $M_s^+ = M_s^0 \cap \Omega_s$  denote the grid points inside each subdomain  $\Omega_s$ , and  $M_s^- = M_s^0 \setminus M_s^+$  the grid points outside each subdomain  $\Omega_s$ . Note that the auxiliary domains  $\Omega_1^0, \Omega_2^0$  and auxiliary grids  $M_1^0, M_2^0$  need not agree, and indeed may be selected completely independently, given considerations regarding accuracy, adaptivity, or efficiency.

Define a finite-difference stencil  $N_{j,k}^{s,\alpha}$ , with  $\alpha = 5, 9$ , to be the stencil of the standard five-point or a wide nine-point Laplacian, i.e.,

$$\begin{aligned} N_{j,k}^{s,5} &= \{(x_j, y_k), (x_{j\pm 1}, y_k), (x_j, y_{k\pm 1})\} \text{ or} \\ N_{j,k}^{s,9} &= \{(x_j, y_k), (x_{j\pm 1}, y_k), (x_j, y_{k\pm 1}), (x_{j\pm 2}, y_k), (x_j, y_{k\pm 2})\}. \end{aligned} \tag{30}$$

Next, with  $\alpha$  fixed, define the point-sets

$$N_s^0 = \bigcup_{(x_j, y_k) \in M_s^0} N_{j,k}^{s,\alpha}, \quad N_s^+ = \bigcup_{(x_j, y_k) \in M_s^+} N_{j,k}^{s,\alpha}, \quad \text{and} \quad N_s^- = \bigcup_{(x_j, y_k) \in M_s^-} N_{j,k}^{s,\alpha}. \tag{31}$$



**Fig. 4** An example of the point-sets for the second-order Difference Potentials Method, applied to a single domain ( $s$  omitted for brevity), with a rotated ellipse  $\Omega$  and a rectangular auxiliary domain  $\Omega^0$ , showing (a) the physical grids  $M^+$  (solid dots) and  $N^+$  (open circles), with  $M^+ \subset N^+$ , and (b) the discrete grid boundary  $\gamma$  (open circles)

The point-set  $N_s^+$  ( $N_s^-, N_s^0$ ) enlarges the point-set  $M_s^+$  ( $M_s^-, M_s^0$ ), by taking the union of finite-difference stencils at every point in  $M_s^+$  ( $M_s^-, M_s^0$ ). Therefore,  $N_s^+$  contains a “thin row” of points belonging to the complement of  $\Omega_s$ , so that  $N_s^+ \not\subset \Omega_s$ , even though  $M_s^+ \subset \Omega_s$ . (Likewise,  $N_s^- \not\subset \Omega \setminus \Omega_s$ , even though  $M_s^- \subset \Omega \setminus \Omega_s$ ).

Lastly, we now define the important point-set

$$\gamma_s = N_s^+ \cap N_s^-, \tag{32}$$

which we call the *discrete grid boundary*. In words,  $\gamma_s$  is the set of grid points that straddle the continuous interface  $\Gamma$ . (See Fig. 4 for an example of these point-sets, given a single elliptical domain  $\Omega$ .) Note that the point-sets  $M_s^+$ ,  $N_s^+$ , and  $\gamma_s$  will be used throughout the Difference Potentials Method.

Here, we define the fully-discrete finite-difference discretization of (3, 4), and then define the Auxiliary Problem. Indeed, the discretization we consider is

$$L_{\Delta t, h}^s u_s^{i+1} = F_s^{i+1}, \quad (x_j, y_k) \in M_s^+, \tag{33}$$

where (i)  $L_{\Delta t, h}^s u_s^{i+1} := \lambda_s(t^{i+1})\Delta_h u_s^{i+1} - \sigma u_s^{i+1}$ , (ii)  $\Delta_h$  is either a five- or nine-point Laplacian in each subdomain, and (iii)  $\sigma$  and  $F_s^{i+1}$  follow from the choice of time- and spatial-discretizations. (Here, we have simplified notation slightly by assuming that  $h := h_1 = h_2$ , which need not be the same in general.) For full details of the discretization, including the choice of BDF2 or BDF4 in the time discretization, we refer the reader to “Appendix 8.1”.

The choices of discretization (33) in each subdomain need not be the same. As in [3, 6], one could choose a second- and fourth-order discretization on  $M_1^+$  and  $M_2^+$ , respectively, given considerations about accuracy, adaptivity, expected regularity of the analytical solution in each domain, etc.

Next, we define the discrete Auxiliary Problem, which plays a central role in the construction of the Difference Potentials operator, the resulting Boundary Equations with Projection at the discrete grid boundary, and in the numerical approximation of the solution *via* the discrete, generalized Green’s formula.

**Definition 1** (*Discrete Auxiliary Problem (AP)*) At time  $t^{i+1}$ , given the right-hand side grid function  $q_s^{i+1}: M_s^0 \rightarrow \mathbb{R}$ , the following difference Eqs. (34, 35) are defined as the discrete AP.

$$L_{\Delta t, h}^s u_s^{i+1} = q_s^{i+1}, \quad (x_j, y_k) \in M_s^0 \tag{34}$$

$$u_s^{i+1} = 0, \quad (x_j, y_k) \in N_s^0 \setminus M_s^0 \tag{35}$$

*Remark 4* For a given right-hand side  $q_s^{i+1}$ , the solution of the discrete AP (34, 35) defines a discrete Green’s operator  $G_{\Delta t, h}^s q_s^{i+1}$ . The choice of boundary conditions (35) will affect the resulting grid function  $G_{\Delta t, h}^s q_s^{i+1}$ , and thus the Boundary Equations with Projection defined below. However, the choice of boundary conditions (35) in the AP will not affect the numerical approximation of (3–9), so long as the discrete AP is uniquely solvable and well-posed.

Let us denote by  $G_{\Delta t, h}^s F_s^{i+1}$  the particular solution on  $N_s^+$  of the fully-discrete problem (33), defined by solving the AP (34, 35) with

$$q_s^{i+1} = \begin{cases} F_s^{i+1}, & (x_j, y_k) \in M_s^+, \\ 0, & (x_j, y_k) \in M_s^-, \end{cases} \tag{36}$$

and restricting the solution from  $N_s^0$  to  $N_s^+$ .

Let us also introduce a linear space  $\mathbf{V}_{\gamma_s}$  of all grid functions denoted  $v_{\gamma_s}^{i+1}$ , which are defined on  $\gamma_s$  and extended by zero to the other points of  $N_s^0$ . These grid functions are referred to as *discrete densities on  $\gamma_s$* .

**Definition 2** (*The Difference Potential of a density.*) The Difference Potential of a given density  $v_{\gamma_s}^{i+1}$  is the grid function  $P_{N_s^+}^{i+1} v_{\gamma_s}^{i+1}$  on  $N_s^+$ , defined by solving the AP (34, 35) with

$$q_s^{i+1} = \begin{cases} 0, & (x_j, y_k) \in M_s^-, \\ L_{\Delta t, h}^s [v_{\gamma_s}^{i+1}], & (x_j, y_k) \in M_s^+, \end{cases} \tag{37}$$

and restricting the solution from  $N_s^0$  to the point-set  $N_s^+$ .

Note that  $P_{N_s^+}^{i+1}: \mathbf{V}_{\gamma_s} \rightarrow N_s^+$  is a linear operator on the space  $\mathbf{V}_{\gamma_s}$  of densities. Moreover, the coefficients of  $P_{N_s^+}^{i+1}$  can be computed by solving the AP (Definition 1) with the appropriate density  $v_{\gamma_s}^{i+1}$  defined at the points  $(x_j, y_k) \in \gamma_s$ .

**Definition 3** (*The Trace operator.*) Given a grid function  $v_s^{i+1}$ , we denote by  $\text{Tr}_{\gamma_s} [v_s^{i+1}]$  the Trace (or Restriction) from  $N_s^+$  to  $\gamma_s$ .

Moreover, for a given density  $v_{\gamma_s}^{i+1}$ , denote the trace of the Difference Potential of  $v_{\gamma_s}^{i+1}$  by  $P_{\gamma_s} v_{\gamma_s}^{i+1}$ . In other words,  $P_{\gamma_s} v_{\gamma_s}^{i+1} = \text{Tr}_{\gamma_s} [P_{N_s^+}^{i+1} v_{\gamma_s}^{i+1}]$ .

Now we can state the central theorem of the Difference Potentials Method that will allow us to reformulate the finite-difference Eq. (33) on  $M_s^+$  (without imposing any boundary or interface conditions yet) into the equivalent Boundary Equations with Projections on  $\gamma_s$ .

**Theorem 1** (*Boundary Equations with Projection (BEP)*) At time-level  $t^{i+1}$ , the discrete density  $u_{\gamma_s}^{i+1}$  ( $s = 1, 2$ ) is the trace of some solution  $u_s^{i+1}$  on domain  $\Omega_s$  to the Difference Eq. (33), i.e.,  $u_{\gamma_s}^{i+1} := \text{Tr}_{\gamma_s} [u_s^{i+1}]$ , if and only if the following BEP holds

$$u_{\gamma_s}^{i+1} - P_{\gamma_s}^{i+1} u_{\gamma_s}^{i+1} = \text{Tr}_{\gamma_s} [G_{\Delta t, h}^{i+1} F_s^{i+1}], \quad (x_j, y_k) \in \gamma_s, \tag{38}$$

with  $\text{Tr}_{\gamma_s} [\cdot]$  and  $P_{\gamma_s}$  defined in Definition 3.

*Proof* See [65] for the general theory of DPM (including the proof for general elliptic PDE), or one of [3,5,6] for the proof in the case of parabolic interface problems.  $\square$

*Remark 5* A given density  $v_{\gamma_s}^{i+1}$  is the trace of some solution of the fully-discrete finite-difference Eq. (33) if and only if it is a solution of the BEP.

However, since boundary or interface conditions have not yet been imposed, the BEP will have infinitely many solutions  $u_{\gamma_s}^{i+1}$ . As originally disussed in [3–6,26,28], in this work we consider the following approach in order to find a unique solution of the BEP.

At each time level  $t^{i+1}$ , one can approximate the solution of (3–9) at the discrete grid boundary  $\gamma_s$ , using the Cauchy data of (3–9) on the continuous interface  $\Gamma$ , up to the desired second- or fourth-order accuracy. (By Cauchy data, we mean the trace of the solution of (3–9), together with the trace of its normal derivative, on  $\Gamma$ .) Below, we will define an Extension Operator which will extend the Cauchy data of (3–9) from  $\Gamma$  to  $\gamma_s$ .

As we will see, the Extension Operator in this work depends only on the given parabolic interface model. Moreover, we will use a finite-dimensional, spectral representation for the Cauchy data of (3–9) on  $\Gamma$ . Then, we will use the Extension Operator, together with the BEP (38) and the interface conditions (8, 9), to obtain a linear system of equations for the coefficients of the finite-dimensional, spectral representation. Hence, the derived BEP will be solved for the unknown coefficients of the Cauchy data. Using this obtained Cauchy data, we will construct the approximation of (3–9) using the Extension Operator, together with the discrete, generalized Green’s formula.

Let us now briefly discuss the Extension Operator for the second-order numerical method, and refer the reader to “Appendix 8.2” for details (including details for the fourth-order numerical method). For points in the vicinity of  $\Gamma$ , we define a coordinate system  $(d, \vartheta)$ , where  $\vartheta$  is arclength from some reference point, and  $d$  is the signed distance in the normal direction from the point to  $\Gamma$ . Now, as a first step towards defining the Extension Operator, we define a new function

$$v_s^{i+1}(d, \vartheta) = v_s^{i+1}(0, \vartheta) + \sum_{l=1}^p \frac{1}{l!} \frac{\partial^l v_s^{i+1}(0, \vartheta)}{\partial n^l} d^l, \tag{39}$$

where  $n$  is the unit outward normal vector at  $\Gamma$ . We choose  $p = 2$  for the second-order method (which we will discuss now) and  $p = 4$  for the fourth-order method (see “Appendix 8.2”).

As a next step for the second-order method (BDF2–DPM2), we define

$$v_s^{i+1}(0, \vartheta) = u_s^{i+1}|_{\Gamma}, \quad \frac{\partial v_s^{i+1}(0, \vartheta)}{\partial n} = \frac{\partial u_s^{i+1}}{\partial n} \Big|_{\Gamma}, \quad \text{and} \quad \frac{\partial^2 v_s^{i+1}(0, \vartheta)}{\partial n^2} = \frac{\partial^2 u_s^{i+1}}{\partial n^2} \Big|_{\Gamma}, \tag{40}$$

where  $u_s^{i+1} := u_s(x, y, t^{i+1})$ ,  $\frac{\partial u_s^{i+1}}{\partial n} := \frac{\partial u_s(x, y, t^{i+1})}{\partial n}$ , etc. As a last step, a straightforward sequence of calculations (see “Appendix 8.2”) shows that

$$\frac{\partial^2 u_s^{i+1}}{\partial n^2} \approx \frac{1}{\lambda_s} \left( \frac{3u_s^{i+1} - 4u_s^i + u_s^{i-1}}{2\Delta t} - f^{i+1} \right) - \frac{\partial^2 u_s^{i+1}}{\partial \vartheta^2} + \kappa \frac{\partial u_s^{i+1}}{\partial n}, \tag{41}$$

where  $\kappa$  denotes the curvature of  $\Gamma$ . Therefore, with  $v_s^{i+1}(d, \vartheta)$  defined by (39–41), the only unknown data at each time step  $t^{i+1}$  are the unknown Dirichlet data  $u_s^{i+1}$  and the unknown Neumann data  $\frac{\partial u_s^{i+1}}{\partial n}$ . The Extension Operator will incorporate the interface conditions (8, 9) when it is combined with the BEP (38), so that the only independent unknowns at each time

step  $t^{i+1}$  will be  $\left(u_1^{i+1}, \frac{\partial u_1^{i+1}}{\partial n}\right)\Big|_\Gamma$  or  $\left(u_2^{i+1}, \frac{\partial u_2^{i+1}}{\partial n}\right)\Big|_\Gamma$ . (This is also true for the fourth-order numerical method—see Appendices 8.2 and 8.3.)

Now we are ready to define the Extension Operator that extends the Cauchy data of (3–9) from  $\Gamma$  to  $\gamma_s$ .

**Definition 4** (*The Extension Operator*) Let  $v_s^{i+1}(d, \vartheta)$  be defined by (39–41). Let  $u_{s,\Gamma}^{i+1}$  denote the Cauchy data of (3–9) at  $t^{i+1}$  on  $\Gamma$ , i.e.,  $u_{s,\Gamma}^{i+1} = \left(u_s^{i+1}, \frac{\partial u_s^{i+1}}{\partial n}\right)\Big|_\Gamma$ . The extension operator  $\text{Ex}_s$  that extends  $u_{s,\Gamma}^{i+1}$  from  $\Gamma$  to  $\gamma_s$  is

$$\text{Ex}_s u_{s,\Gamma}^{i+1} := v_s^{i+1}(d, \vartheta)|_{\gamma_s}. \tag{42}$$

For a given point  $(x_j, y_k) \in \gamma_s$ , note that  $d$  is the signed distance between  $(x_j, y_k)$  and its orthogonal projection on  $\Gamma$ , while  $\vartheta$  is the arclength along  $\Gamma$  between a reference point and the orthogonal projection of  $(x_j, y_k)$ .

Next, we briefly discuss the finite-dimensional, spectral representation of Cauchy data  $u_{s,\Gamma}^{i+1}$ . Indeed, we wish to choose a basis  $\phi_\nu(\vartheta)$  on  $\Gamma$  ( $\nu = 1, 2, 3, \dots$ ) in order to accurately approximate the two components of the Cauchy data  $u_{s,\Gamma}^{i+1}$ . To be specific, whichever basis we choose, we require that

$$\begin{aligned} &\varepsilon_{\mathcal{N}^0, \mathcal{N}^1}(u_{s,\Gamma}^{i+1}) \\ &= \min_{c_{1,\nu}^{s,i+1}, c_{2,\nu}^{s,i+1}} \int_\Gamma \left( \left| u_s^{i+1} - \sum_{\nu=1}^{\mathcal{N}^0} c_{1,\nu}^{s,i+1} \phi_\nu(\vartheta) \right|^2 + \left| \frac{\partial u_s^{i+1}}{\partial n} - \sum_{\nu=1}^{\mathcal{N}^1} c_{2,\nu}^{s,i+1} \phi_\nu(\vartheta) \right|^2 \right) d\vartheta \end{aligned} \tag{43}$$

tends to zero as  $\mathcal{N}^0, \mathcal{N}^1 \rightarrow \infty$ , for some sequence of real numbers  $(c_{1,\nu}^{s,i+1})_{\nu=1}^{\mathcal{N}^0}$  and  $(c_{2,\nu}^{s,i+1})_{\nu=1}^{\mathcal{N}^1}$ . In other words, we require

$$\lim_{\mathcal{N}^0, \mathcal{N}^1 \rightarrow \infty} \varepsilon_{\mathcal{N}^0, \mathcal{N}^1}(u_{s,\Gamma}^{i+1}) = 0. \tag{44}$$

Now let us discuss a choice of basis. In this work, recall that we consider interfaces  $\Gamma$  that are at least  $C^2(\Gamma)$  (due to the choice of smooth, circular geometries). Also, as we will see in Sect. 4.1, each function  $u$  considered in the test problems on a composite domain (TP–2A, TP–2B, TP–2C) is locally smooth, in the sense that  $u|_{\Omega_1} = u_1$  and  $u|_{\Omega_2} = u_2$  are smooth in  $\Omega_1$  and  $\Omega_2$ , respectively. Moreover, each component of the Cauchy data  $u_{1,\Gamma}^{i+1}$  and  $u_{2,\Gamma}^{i+1}$  are smooth, periodic functions of arclength  $\vartheta$ . (Note that  $u_{1,\Gamma}^{i+1}$  and  $u_{2,\Gamma}^{i+1}$  need not agree, and indeed do not—neither  $\mu_1(x, y, t)$  nor  $\mu_2(x, y, t)$  in (8, 9) are identically equal to zero, for any of our test problems on a composite domain.)

Therefore, in this work, we choose a standard trigonometric basis  $\phi_\nu(\vartheta)$ , with

$$\phi_1(\vartheta) = 1, \quad \phi_{2K}(\vartheta) = \cos\left(\frac{2\pi K \vartheta}{|\Gamma|}\right), \quad \phi_{2K+1}(\vartheta) = \sin\left(\frac{2\pi K \vartheta}{|\Gamma|}\right), \tag{45}$$

and  $K > 1$ . Moreover, at every time step  $t^{i+1}$ , we will discretize the Cauchy data  $u_{s,\Gamma}^{i+1} = \left(u_s^{i+1}, \frac{\partial u_s^{i+1}}{\partial n}\right)\Big|_\Gamma$  using this basis. Therefore, we let

$$\tilde{u}_{s,\Gamma}^{i+1} = \sum_{\nu=1}^{\mathcal{N}^0} c_{1,\nu}^{s,i+1} \Phi_\nu^0(\vartheta) + \sum_{\nu=1}^{\mathcal{N}^1} c_{2,\nu}^{s,i+1} \Phi_\nu^1(\vartheta) \quad \text{and} \quad \tilde{u}_{s,\Gamma}^{i+1} \approx u_{s,\Gamma}^{i+1}, \tag{46}$$

where  $\Phi_v^0 = (\phi_v, 0)$  and  $\Phi_v^1 = (0, \phi_v)$  are the set of basis functions used to represent the Cauchy data on the interface  $\Gamma$ .

*Remark 6* It should be also possible to relax regularity assumption on the domain under consideration. For example, one can consider piecewise-smooth, locally-supported basis functions (defined on  $\Gamma$ ) as the part of the Extension Operator. For example, [52] use this approach to design a high-order accurate numerical method for the Helmholtz equation, in a geometry with a reentrant corner. Furthermore, [80,81] combine the DPM together with the XFEM, and design a DPM for linear elasticity in a non-Lipschitz domain (with a cut).

Next, in “Appendix 8.3”, we derive a linear system for the coefficients  $(c_{1,v}^{s,i+1})_{v=1}^{N^0}$  and  $(c_{2,v}^{s,i+1})_{v=1}^{N^1}$ , by combining the interface conditions (8, 9), the BEP (38), the Extension Operator (39–41), and the spectral discretization (46). Then, the numerical approximation  $u_s^{i+1} \approx u_s(x_j, y_k, t^{i+1})$  of (3–9) at all grid-points  $(x_j, y_k) \in N_s^+$  follows directly from the discrete, generalized Green’s formula, which we state now.

**Definition 5** (*Discrete, generalized Green’s formula.*) At each time step  $t^{i+1}$ , the numerical approximation  $u_s^{i+1} \approx u_s(x_j, y_k, t^{i+1})|_{(x_j, y_k) \in N_s^+}$  of (3–9) is given by

$$u_s^{i+1} := P_{N_s^+}^{i+1} u_{\gamma_s}^{i+1} + G_{\Delta t, h}^s F_s^{i+1}. \tag{47}$$

Here,  $u_{\gamma_s}^{i+1} = \text{Ex}_s \tilde{u}_{s, \Gamma}^{i+1}$ , and  $\tilde{u}_{s, \Gamma}^{i+1}$  is constructed from (i) the solution of the BEP (see “Appendix 8.3”) and (ii) the spectral discretization (46). (Recall that  $P_{N_s^+}^{i+1} u_{\gamma_s}^{i+1}$  is the Difference Potential of the density  $u_{\gamma_s}^{i+1}$ , while  $G_{\Delta t, h}^s F_s^{i+1}$  is the Particular Solution.)

In this work, we also propose a novel feature of DPM, extending the method originally developed in [67] and [3–6,26,28] to the composite domain problem (3–9) with implicitly-defined geometry. The primary difference between Difference Potentials Methods on explicitly-defined versus implicitly-defined composite domains is in the approximation of the interface  $\Gamma$ , which must be done accurately and efficiently, in order to maintain the desired second- or fourth-order accuracy.

The main idea of DPM-based methods for implicitly-defined geometry is to seek an accurate and efficient explicit parameterization of the implicit boundary/interface. First, we represent the geometry implicitly *via* a level set function  $F(x, y)$  on  $M^0$ . Then we construct a local interpolant  $\tilde{F}(x, y)$  of  $F(x, y)$  on a subset of  $M^0$  near the continuous interface  $\Gamma$ . Next, we parameterize  $\Gamma$  by arc-length using numerical quadrature. With this parameterization, we (i) compute the Fourier series expansion from initial conditions for the Cauchy data  $u_{s, \Gamma}^{i+1}$  on the implicit interface  $\Gamma$ , and (ii) construct the extension operators (Definition 4) with  $p = 2$  or  $p = 4$ .

**Conjecture 1** (High-order accuracy of the DPM with implicit geometry) *Due to the second- or fourth-order accuracy (in both space and time) of the underlying discretization (33), the extension operator (42) with  $p = 2$  or  $p = 4$ , and the established error estimates and convergence results for the DPM for general linear elliptic boundary value problems on smooth domains (presented in [33,62,63,65]), we expect second- and fourth-order accuracy in the maximum norm for the error in the computed solution (59 or 60) for both the single and composite domain parabolic problems.*

*Remark 7* Indeed, in the numerical results (Sect. 5) we see that the computed solution (47) at every time level  $t^{i+1}$  has accuracy  $\mathcal{O}(h^2 + \Delta t^2)$  for the second-order method, and  $\mathcal{O}(h^4 + \Delta t^4)$

for the fourth-order method, for both the single and composite domain problems, with explicit or implicit geometry. See [3, 6, 27, 67] for more details and numerical tests involving explicit (circular and elliptical) geometries.

*Main Steps of the algorithm* Let us summarize the main steps for the Difference Potentials Method.

- *Step 1* Introduce a computationally simple Auxiliary Domain  $\Omega_s^0$  ( $s = 1, 2$ ) and formulate the Auxiliary Problem (AP; Definition 1).
- *Step 2* At each time step  $t^{i+1}$ , compute the Particular Solution  $u_s^{i+1} = G_{\Delta t, h}^{i+1} F_s^{i+1}$ ,  $(x_j, y_k) \in N_s^+$ , using the AP with the right-hand side (36).
- *Step 3* Construct the matrix in the boundary Eq. (79) (discussed in “Appendix 8.3”), derived from the Boundary Equation with Projection (BEP) (38), via several solutions of the AP. (When the diffusion coefficients  $\lambda_s$  are constant, this is done once, as a pre-processing step before the first time step.)
- *Step 4* Solve boundary Eq. (79) and compute the approximation of the density  $u_{\gamma_s}^{i+1}$ , by applying the Extension Operator (42) to the solution of (79).
- *Step 5* Construct the Difference Potentials  $P_{N_s^+ \gamma_s} u_{\gamma_s}^{i+1}$  of the density  $u_{\gamma_s}^{i+1}$ , using the AP with the right-hand side (37).
- *Step 6* Compute the numerical approximation  $u_s^{i+1} \approx u_s(x_j, y_k, t^{i+1})$  of the PDE (3–9) using the discrete, generalized Green’s formula (47).

### 3.3 SBP–SAT–FD

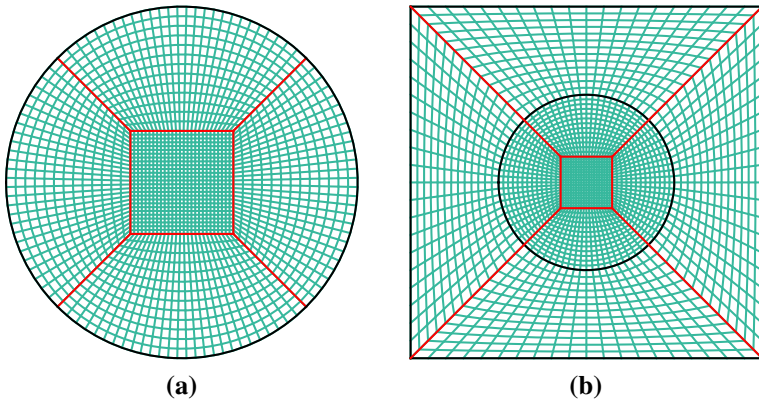
We continue in this section with a brief presentation of SBP–SAT–FD, for solving the parabolic problems presented in Sect. 2. For more detailed discussions of the SBP–SAT–FD method, we refer the reader to two review papers [21, 73].

The SBP–SAT–FD method was originally used on Cartesian grids. To resolve complex geometries, we consider a grid mapping approach by transfinite interpolation [43]. A smooth mapping requires that the physical domain is a quadrilateral, possibly with smooth, curved sides. If the physical domain does not have the desired shape, we then partition the physical domain into subdomains, so that each subdomain can be mapped smoothly to the reference domain. As an example, the single domain of equation (1, 2), shown in Figure 5a, is divided into five subdomains. The five subdomains consist of one square subdomain, and four identical quadrilateral subdomains (modulo rotation by  $\pi/2$ ) with curved sides. Similarly, the composite domain of equation (3–9) is divided into nine subdomains, as shown in Fig. 5b. Suitable interface conditions are imposed to patch the subdomains together.

Although the side-length of the centered square is arbitrary (as long as the square is strictly inside the circle), its size and position have a significant impact on the quality of the curvilinear grid. In a high-quality mesh, the elements should not be skewed too much, and the sizes of the elements should be nearly uniform. In practice, it is usually difficult to know a priori the optimal way of domain division.

A Cartesian grid in the reference domain is mapped to a curvilinear grid in each subdomain. The grids are aligned with boundaries and interfaces, thus avoiding small-cut difficulties sometimes associated with embedded methods. In this paper, we only consider conforming grid interfaces, i.e., the grid points from two adjacent blocks match on the interface. For numerical treatment of non-conforming grid interfaces in the SBP–SAT–FD framework, see [44, 55].

When a physical domain is mapped to a reference domain, the governing equation is transformed to the Cartesian coordinate in the reference domain. The transformed equation



**Fig. 5** A (a) circular domain (cf. Fig. 1), divided into five subdomains; and (b) composite domain, divided into nine subdomains

is usually in a more complicated form than the original equation. In general, a parabolic problem

$$u_t = u_{xx} + u_{yy}, \quad (x, y) \in \Omega \tag{48}$$

in a physical domain will be transformed to

$$Ju_t = (\alpha u_\xi)_\xi + (\beta u_\eta)_\xi + (\beta u_\xi)_\eta + (\gamma u_\eta)_\eta, \quad (\xi, \eta) \in [0, 1]^2, \tag{49}$$

where  $(\xi, \eta)$  is the Cartesian coordinate in the unit square, and  $J(\xi, \eta)$ ,  $\alpha(\xi, \eta)$ ,  $\beta(\xi, \eta)$ ,  $\gamma(\xi, \eta)$  depend on the geometry of the physical domain and on the chosen mapping. In particular, we use transfinite interpolation for the grid mapping. In this case, the precise form of (49) and the derivation of the grid transformation are presented in Section 3.2 of [7]. Even though the original equation is in the simplest form with unit coefficients, the transformed equation has variable coefficients and mixed derivatives. Therefore, it is important to construct multi-block finite difference methods solving the transformed equation (49). Hence, we need two SBP operators,  $D_1 \approx \partial/\partial x$  to approximate a first derivative, and  $D_2^{(b)} \approx \partial/\partial x(b(x)\partial/\partial x)$  to approximate a second derivative with variable coefficient, where  $b(x) > 0$  is a known function. Below we discuss SBP properties, and start with the first derivative.

Consider two smooth functions  $u(x), v(x)$  on  $x \in [0, 1]$ . We discretize  $[0, 1]$  uniformly by  $N$  grid points, and denote the restriction of  $u(x), v(x)$  onto the grid by  $\mathbf{u}, \mathbf{v}$ , respectively. Integration by parts states:

$$\int_0^1 u_x v \, dx = uv \Big|_0^1 - \int_0^1 uv_x \, dx. \tag{50}$$

The SBP operator  $D_1$  mimics integration by parts:

$$(D_1 \mathbf{u})^T H \mathbf{v} = \mathbf{u}^T B \mathbf{v} - \mathbf{u}^T H D_1 \mathbf{v}, \tag{51}$$

where  $H$  is symmetric positive definite—thus defining an inner product—and

$$B = \text{diag}(-1, 0, \dots, 0, 1).$$

In fact,  $H$  is also a quadrature [20]. It is easy to verify that (51) is equivalent to

$$D_1^T H + H D_1 = B, \tag{52}$$



which is the SBP property for the first derivative operator. At the grid points in the interior of the domain, standard, central, finite-difference stencils can be used in  $D_1$ , and the weights of the standard, discrete  $L_2$ -norm are used in  $H$ . At a few points close to boundaries, special stencils and weights must be constructed in  $D_1$  and  $H$ , respectively, to satisfy (52).

The SBP operators  $D_1$  were first constructed in [45] and later revisited in [72]. The SBP norm  $H$  can be diagonal or non-diagonal. While non-diagonal norm SBP operators have a better accuracy property than diagonal norm SBP operators, when terms with variable coefficients are present in the equation, a stability proof is only possible with diagonal norm SBP operators. Therefore, we use diagonal norm SBP operators in this paper.

For a second derivative with variable coefficients, the SBP operators  $D_2^{(b)}$  were constructed in [54]. We remark that applying  $D_1$  twice also approximates a second derivative, but is less accurate and more computationally expensive than  $D_2^{(b)}$ .

Due to the choice of centered difference stencils at interior grid points, the order of accuracy of the SBP operators is even at these points, and is often denoted by  $2p$ . To fulfill the SBP property, at a few grid points near boundaries, the order of accuracy is reduced to  $p$  for diagonal norm operators. This detail notwithstanding, such a scheme is often referred to as  $2p^{\text{th}}$ -order accurate. In fact, for the second- and fourth-order SBP–SAT–FD schemes used in this paper to solve parabolic problems, we can expect a second- and fourth-order overall convergence rate, respectively [77].

An SBP operator only approximates a derivative. When imposing boundary and interface conditions, it is important that the SBP property is preserved and an energy estimate is obtained. For this reason, we consider the SAT method [16], where penalty terms are added to the semi-discretization, imposing the boundary and interface conditions weakly. This bears similarities with the Nitsche finite element method [60] and the discontinuous Galerkin method [40].

We note that in [75], SBP–SAT–FD methods were developed for the wave equation

$$Jv_{tt} = (av_\xi)_\xi + (bv_\eta)_\xi + (bv_\xi)_\eta + (cv_\eta)_\eta, \quad (\xi, \eta) \in [0, 1]^2, \quad (53)$$

with Dirichlet boundary conditions, Neumann boundary conditions, and interface conditions. Comparing Eq. (53) with (49), the only difference is that the wave equation has a second derivative in time, while the heat equation has a first derivative in time. The spatial derivatives of (53) and (49) are the same.

Assuming homogeneous boundary data for simplified notation, we write the SBP–SAT–FD discretization of (53) as

$$\mathbf{v}_{tt} = Q\mathbf{v}, \quad (54)$$

where  $Q$  is the spatial discretization operator including the boundary implementation. For the scheme developed in [75], stability is proved by the energy method by multiplying (54) by  $\mathbf{v}_t^T H_2$  from the left,

$$\mathbf{v}_t^T H_2 \mathbf{v}_{tt} = \mathbf{v}_t^T H_2 Q \mathbf{v}, \quad (55)$$

where  $H_2$  is a diagonal, positive-definite operator, obtained through a tensor product from the corresponding SBP norm,  $H$ , in one spatial dimension. It is shown in [75] that  $H_2 Q$  is symmetric and negative semi-definite. Therefore, we can write (55) as

$$\frac{d}{dt} (\mathbf{v}_t^T H_2 \mathbf{v}_t - \mathbf{v}^T H_2 Q \mathbf{v}) = 0,$$

where the discrete energy,  $\mathbf{v}_t^T H_2 \mathbf{v}_t - \mathbf{v}^T H_2 Q \mathbf{v}$ , for (53) is conserved.

If we use the same operator  $Q$  to discretize the heat equation (49) with the same boundary condition as the wave equation (53), then the scheme

$$\mathbf{v}_t = Q\mathbf{v}, \tag{56}$$

is also stable. To see this, we multiply (56) by  $\mathbf{v}^T H_2$  from the left, and obtain

$$\frac{d}{dt}(\mathbf{v}^T H_2 \mathbf{v}) = \mathbf{v}^T H_2 Q \mathbf{v} \leq 0, \tag{57}$$

where  $\mathbf{v}^T H_2 \mathbf{v}$  is the discrete energy for (49). In this paper, we use the spatial discretization operators developed in [75] to solve both the single (1, 2) and composite domain problems (3–9).

In [10], SBP–SAT–FD methods are discussed for the one-dimensional heat equation with constant coefficients, both in a single domain and a composite domain. In theory, these schemes can also be generalized to solve Eq. (49), but are different from the ones used in this paper.

### 4 Test Problems

In this section, we first list the test problems that we will consider (in Sect. 4.1), and then briefly motivate and discuss these choices (in Sect. 4.2). The tests we propose are “manufactured solutions”, in the sense that we state an exact solution  $u(x, y, t)$  or  $(u_1(x, y, t), u_2(x, y, t))$  and a diffusion coefficient  $\lambda(t)$  or  $(\lambda_1, \lambda_2)$ . From (1, 2) (for the single domain problem) or (3–9) (for the composite domain problem) we compute the (i) right-hand side, (ii) initial conditions, (iii) boundary condition, and (iv) functions  $(\mu_1(x, y, t), \mu_2(x, y, t))$  for the interface/matching conditions. Then, (i–iv), together with the diffusion coefficient, serve as the inputs for our numerical methods.

#### 4.1 List of Test Problems

1. Single-domain, with an explicitly-defined boundary for DPM and SBP–SAT–FD, or an implicitly-defined boundary for cut-FEM.
  - (a) Constant diffusion (Test Problem 1A; TP–1A): Consider the PDE (1, 2), with  $\lambda(t) \equiv 1$ ,  $\Omega = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\}$ , and the final time  $T = 1.0$ . Then, TP–1A (adapted from [6]), is given by

$$u(x, y, t) = x^9 y^8 e^{-t}. \tag{TP–1A}$$

- (b) Time-varying diffusion (Test Problem 3A; TP–3A): Same as TP–1A, but with diffusion coefficient

$$\lambda(t) = 11/10 + \sin(10\pi t). \tag{TP–3A}$$

2. Composite-domain, with an explicitly-defined interface (for DPM and SBP–SAT–FD) or implicitly-defined interface (for cut-FEM and DPM). Consider the PDE (3–9), with  $\Omega = [-2, 2] \times [-2, 2]$ ,  $\Omega_2 = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\}$ ,  $\Omega_1 = \Omega \setminus \Omega_2$ ,  $\Gamma = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1\}$ , and the final time  $T = 1.0$ .

- (a) (Test Problem 2A; TP–2A): A modified version of the test adapted from [6,48]. Let  $(\lambda_1, \lambda_2) = (10, 1)$ , and

$$u(x, y, t) = \begin{cases} e^{-t} \sin x \cos y, & (x, y) \in \Omega_1, \\ e^{-t}(x^2 - y^2), & (x, y) \in \Omega_2. \end{cases} \quad (\text{TP-2A})$$

- (b) High-frequency oscillations (Test Problem 2B; TP–2B): A modified version of the test adapted from [6]. Let  $(\lambda_1, \lambda_2) = (10, 1)$ , and

$$u(x, y, t) = \begin{cases} e^{-t} \sin(3\pi x) \cos(7\pi y), & (x, y) \in \Omega_1, \\ e^{-t}(x^2 - y^2), & (x, y) \in \Omega_2. \end{cases} \quad (\text{TP-2B})$$

- (c) Large contrast in diffusion coefficients, and large jumps in both solution and flux at interface (Test Problem 2C; TP–2C): Let  $(\lambda_1, \lambda_2) = (1000, 1)$ , and

$$u(x, y, t) = \begin{cases} 0, & (x, y) \in \Omega_1, \\ 1000 \sin(10t)x^4y^5, & (x, y) \in \Omega_2. \end{cases} \quad (\text{TP-2C})$$

## 4.2 Motivation of the Chosen Test Problems

Test Problem 1A (TP–1A) involves a high-degree polynomial, with total degree of 17. This is a rather straightforward test problem, which allows us to establish a good “baseline” with which to compare each method. The choice of high degree ensures that there will be no cancellation of local truncation error, so that we should see—at most—second- or fourth-order convergence for the given methods, barring some type of superconvergence. Next, (TP–3A) adds on (incrementally) the complication of time-varying diffusion.

Likewise, (TP–2A) offers a straightforward “baseline” with which to consider the interface problem: The test problem is piecewise-smooth, and the geometry is simplified (see Remark 1). However, there is a jump in both the analytical solution and its flux, which requires a well-designed numerical method to accurately approximate. Moreover, (TP–2A) was first proposed in [48] (see also [6]), and is a good comparison with the immersed interface method therein.

Then, (TP–2B) adds additional challenges onto (TP–2A) in the form of much higher-frequency oscillations; while (TP–2C) adds onto (TP–2A) in the form of both (i) large contrast in diffusion, and (ii) large jumps in the analytical solution and its flux.

## 5 Numerical Results

### 5.1 Time Discretization

The spatial discretization for each method is discussed in Sect. 3. For the time discretization, the backward differentiation formulas of second- and fourth-order (BDF2 and BDF4) are used for the second- and fourth-order methods, respectively. In each case, the time-step is given by

$$\Delta t = 0.5h. \quad (58)$$

However, note that  $h$  in (58) bears different physical meanings for each method. Indeed, for cut-FEM,  $h$  is the average distance between the Gauss–Lobatto points; for DPM,  $h$  is the grid spacing in the uniform, Cartesian grid  $M^0$  (see the text prior to (33)); and for SBP–SAT–FD,  $h$  is the minimum grid spacing in the reference domain.

**Table 1** Convergence in the maximum norm (59), for the second- and fourth-order versions of each method, applied to Test Problem 1A (TP-1A), with diffusion coefficient  $\lambda = 1$ , and time-step  $\Delta t = 0.5h$

DOF	<i>E</i> : CUT2	Rate	DOF	<i>E</i> : CUT4	Rate
9944	4.9327E−5	–	10,276	3.1799E−6	–
40,072	1.3798E−5	1.80	39,613	1.9848E−7	4.00
159,912	3.7114E−6	1.88	159,700	1.3330E−8	3.82
DOF	<i>E</i> : DPM2	Rate	DOF	<i>E</i> : DPM4	Rate
10,000	1.7105E−5	–	10,000	2.4782E−6	–
40,000	4.1980E−6	2.03	40,000	5.9672E−8	5.38
160,000	1.0135E−6	2.05	160,000	1.7396E−9	5.10
DOF	<i>E</i> : SBP2	Rate	DOF	<i>E</i> : SBP4	Rate
9861	1.5328E−5	–	9861	2.0636E−6	–
40,365	3.6210E−6	2.08	40,365	1.3083E−7	3.98
163,317	8.8008E−7	2.04	163,317	8.1180E−9	4.01

### 5.2 Measure for Comparison

Let  $u^i_{j,k}$  denote the computed numerical approximation of  $u(x, y, t)$  at the grid-point  $(x_j, y_k) \in \Omega$  and time  $t^i = i\Delta t \in (0, T]$ . For the three methods, we will compare the size of the maximum error in  $u$  at the grid points, with respect to the number of degrees of freedom (DOF). For the single domain problem (1, 2), the maximum error is computed as:

$$E := \max_{t^i \in (0, T]} \max_{(x_j, y_k) \in \Omega} |u(x_j, y_k, t^i) - u^i_{j,k}|, \tag{59}$$

and for the composite domain problem (3–9) as:

$$E := \max_{t^i \in (0, T]} \max_{(x_j, y_k) \in \Omega_1 \cup \Omega_2} |u(x_j, y_k, t^i) - u^i_{j,k}|. \tag{60}$$

### 5.3 Convergence Results

In the following tables and figures, we state the number of degrees of freedom in the grid, maximum error (59, 60 for the single- and composite-domain problems, respectively), and an estimate of the rate of convergence.

In Tables 1, 2, 3, 4 and 5, the estimate of rate of convergence is computed as follows. Let  $(DOF_n, E_n)$  be given, with  $n = 1, 2, 3$  referring to the first, second, and third grids (from coarsest to finest). Then, for  $n = 2, 3$ , compute the standard estimate

$$\rho_n = \frac{\log(E_{n-1}/E_n)}{\log(DOF_{n-1}/DOF_n)}, \tag{61}$$

which is the estimated rate of convergence, denoted in Tables 1, 2, 3, 4 and 5 by “Rate”.

In Figs. 6, 7, 10, 11 and 12, the estimate of rate of convergence is computed differently. Computing a least-square linear regression for the data  $(\log_{10}(\sqrt{DOF_n}), \log_{10}(E_n))$  gives a line with slope  $m$ , where  $m$  is the estimate of rate of convergence, reported in the legend on the right side of each figure.

Overall, we see in Tables 1, 2, 3, 4 and 5 that the error for second-order methods (denoted, for brevity, as CUT2, DPM2, SBP2) on the finest mesh is similar, or sometimes larger, than the error for fourth-order methods (denoted CUT4, DPM4, SBP4) on the coarsest mesh—this

**Table 2** Convergence in the maximum norm (59), for the second- and fourth-order versions of each method, applied to Test Problem 3A (TP-3A), with diffusion coefficient  $\lambda(t) = 1.1 + \sin(\pi t)$ , and time-step  $\Delta t = 0.5h$

DOF	<i>E</i> : CUT2	Rate	DOF	<i>E</i> : CUT4	Rate
9944	4.9605E−5	–	10,276	3.0791E−6	–
40,072	1.3851E−5	1.80	39,613	1.9435E−7	3.99
159,912	3.7176E−6	1.89	159,700	1.3161E−8	3.81
DOF	<i>E</i> : DPM2	Rate	DOF	<i>E</i> : DPM4	Rate
10,000	1.7721E−5	–	10,000	2.3422E−6	–
40,000	4.3619E−6	2.02	40,000	5.7588E−8	5.35
160,000	1.0526E−6	2.05	160,000	1.8398E−9	4.97
DOF	<i>E</i> : SBP2	Rate	DOF	<i>E</i> : SBP4	Rate
9861	1.5665E−5	–	9861	1.8858E−6	–
40,365	3.6965E−6	2.08	40,365	1.1949E−7	3.98
163,317	8.9731E−7	2.04	163,317	7.4149E−9	4.01

**Table 3** Convergence in the maximum norm (60), for the second- and fourth-order versions of each method, applied to Test Problem 2A (TP-2A), with diffusion coefficients  $(\lambda_1, \lambda_2) = (10, 1)$ , and time-step  $\Delta t = 0.5h$ . (DPM2-I/DPM4-I refers to the extension of the DPM method, to consider implicit geometry.)

DOF	<i>E</i> : CUT2	Rate	DOF	<i>E</i> : CUT4	Rate
9988	1.0933E−3	–	10,129	2.2215E−6	–
39,988	2.7169E−4	1.97	39,952	1.3254E−07	3.98
159,988	7.2092E−05	1.89	160,729	8.1985E−09	3.93
DOF	<i>E</i> : DPM2	Rate	DOF	<i>E</i> : DPM4	Rate
10,000	3.6380E−5	–	10,000	7.7484E−9	–
40,000	8.8360E−6	2.04	40,000	4.5617E−10	4.09
160000	2.1331E−6	2.05	160,000	2.6398E−11	4.11
DOF	<i>E</i> : DPM2-I	Rate	DOF	<i>E</i> : DPM4-I	Rate
10,000	3.6381E−5	—	10,000	7.7484E−9	–
40,000	8.8360E−6	2.04	40,000	4.5617E−10	4.09
160,000	2.1331E−6	2.05	160,000	2.6396E−11	4.11
DOF	<i>E</i> : SBP2	Rate	DOF	<i>E</i> : SBP4	Rate
10,537	4.7387E−4	–	10,537	3.4655E−5	–
40,905	1.2049E−4	1.98	40,905	4.3052E−6	3.01
161,161	3.0267E−5	1.99	161,161	5.3535E−7	3.01

illustrates the effectiveness of higher-order methods, when high accuracy is important. Additionally, comparing the three methods together, the size of the errors for the single-domain problems (TP-1A, TP-3A) are similar, up to a constant factor; while for the composite-domain problems (TP-2A, TP-2B, TP-2C) we do see differences of one or two orders of magnitude, with the DPM having the smallest errors.

In Table 1 and Fig. 6, we observe that the measured rates of convergence for the numerical approximation of Test Problem 1A (TP-1A) are all  $\approx 2$  (for the second-order versions) or  $\approx 4$  (for the fourth-order versions), except for DPM4, which for this test problem is superconvergent, with fifth-order convergence. Such higher-than-expected convergence might occur

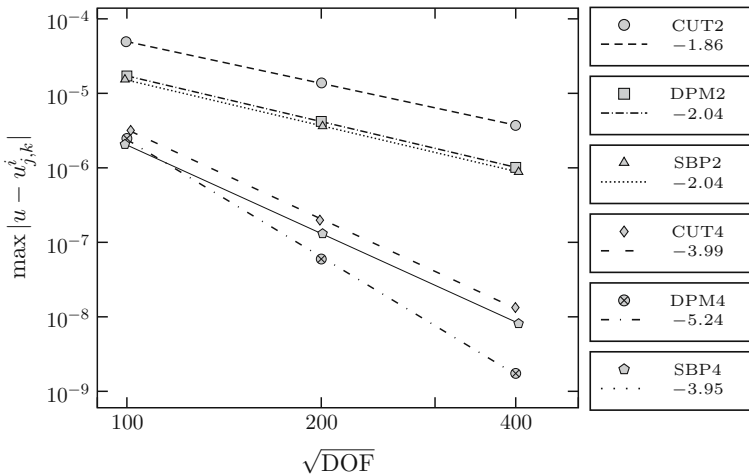
**Table 4** Convergence in the maximum norm (60), for the second- and fourth-order versions of each method, applied to Test Problem 2B (TP-2B), with diffusion coefficients  $(\lambda_1, \lambda_2) = (10, 1)$ , and time-step  $\Delta t = 0.5h$ . (DPM2-I/DPM4-I refers to the extension of the DPM method, to consider implicit geometry.)

DOF	<i>E</i> : CUT2	Rate	DOF	<i>E</i> : CUT4	Rate
9988	2.4855E-1	–	10,129	4.7064E-1	–
39,988	5.6850E-2	2.08	39,952	3.6816E-2	3.60
159,988	1.2346E-2	2.18	160,729	2.2361E-3	3.95
DOF	<i>E</i> : DPM2	Rate	DOF	<i>E</i> : DPM4	Rate
10,000	7.1899E-2	–	10,000	7.3065E-3	–
40,000	1.7868E-2	2.01	40,000	6.0014E-4	3.61
160,000	4.4952E-3	1.99	160,000	3.3086E-5	4.18
DOF	<i>E</i> : DPM2-I	Rate	DOF	<i>E</i> : DPM4-I	Rate
10,000	7.1899E-2	–	10,000	7.3065E-3	–
40,000	1.7868E-2	2.01	40,000	6.0014E-4	3.61
160,000	4.4952E-3	1.99	160,000	3.3086E-5	4.18
DOF	<i>E</i> : SBP2	Rate	DOF	<i>E</i> : SBP4	Rate
10,537	3.2863E-1	–	10,537	2.8321E-1	–
40,905	1.1075E-1	1.57	40,905	3.9277E-2	2.85
161,161	3.5769E-2	1.63	161,161	3.7081E-3	3.40

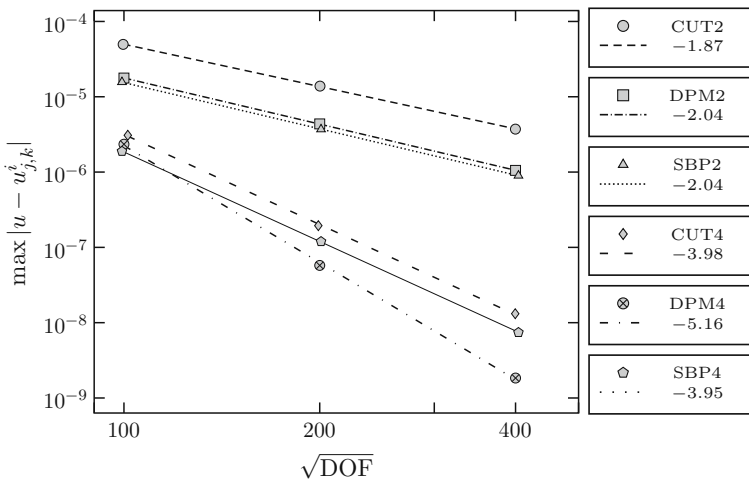
**Table 5** Convergence in the maximum norm (60), for the second- and fourth-order versions of each method, applied to Test Problem 2C (TP-2C), with diffusion coefficients  $(\lambda_1, \lambda_2) = (1000, 1)$ , and time-step  $\Delta t = 0.5h$ . (DPM2-I/DPM4-I refers to the extension of the DPM method, to consider implicit geometry.)

DOF	<i>E</i> : CUT2	Rate	DOF	<i>E</i> : CUT4	Rate
9988	6.4110E-1	–	10129	7.1811E-2	–
39988	1.6506E-1	1.92	39952	3.9995E-3	4.08
159988	3.8719E-2	2.07	160729	2.8978E-4	3.71
DOF	<i>E</i> : DPM2	Rate	DOF	<i>E</i> : DPM4	Rate
10,000	1.1178E-1	–	10,000	1.1392E-3	–
40,000	1.8941E-2	2.56	40,000	5.9291E-5	4.26
160,000	4.0950E-3	2.21	160,000	3.2716E-6	4.18
DOF	<i>E</i> : DPM2-I	Rate	DOF	<i>E</i> : DPM4-I	Rate
10,000	1.0377E-1	–	10,000	1.0905E-3	–
40,000	1.7727E-2	2.55	40,000	5.5494E-5	4.30
160,000	3.8853E-3	2.19	160,000	3.0003E-6	4.21
DOF	<i>E</i> : SBP2	Rate	DOF	<i>E</i> : SBP4	Rate
10,537	1.0025E-1	–	10,537	5.9131E-3	–
40,905	2.5318E-2	1.99	40,905	4.8624E-4	3.60
161,161	6.3459E-3	2.00	161,161	3.5001E-5	3.80

due to several reasons—for example, (i) if the geometry is smooth; (ii) if the magnitude of the derivatives have fast decay (effectively reducing the local truncation error by a factor of  $h$ ); or (iii) if there is cancellation of error due to symmetries in the geometry, or in the analytical solution.



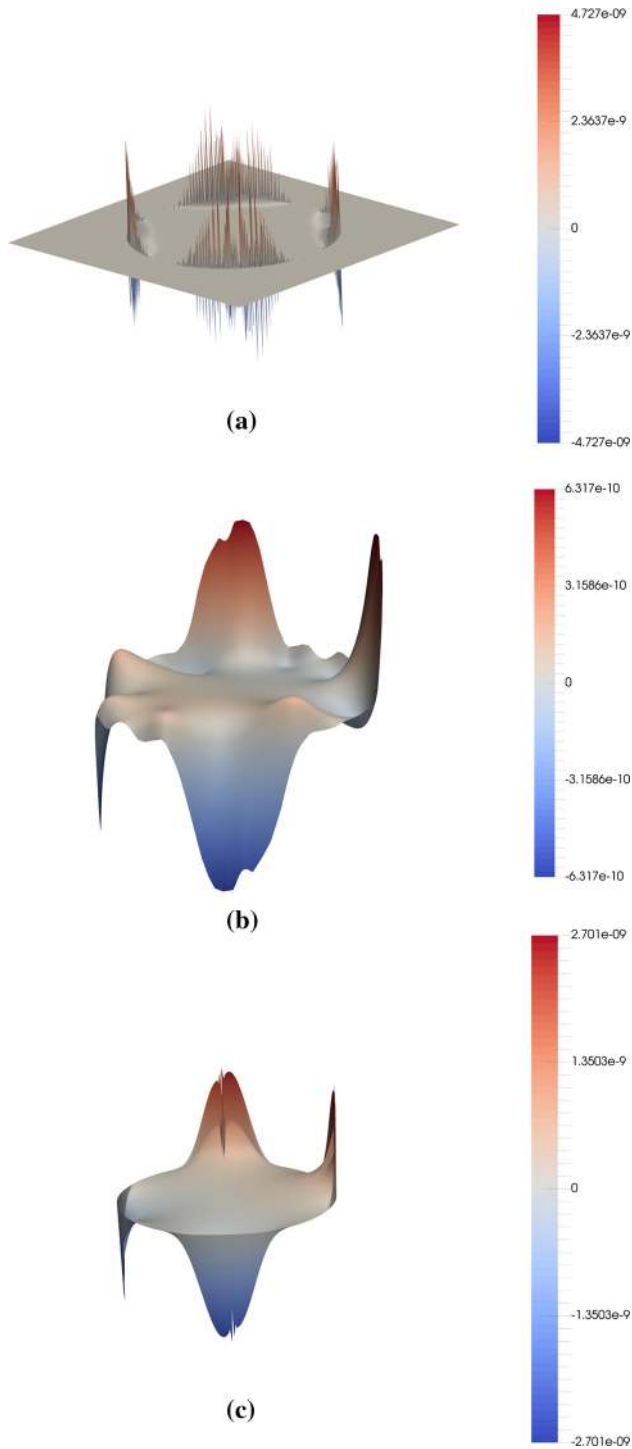
**Fig. 6** Log–log plot of absolute error (59) versus  $\sqrt{\text{DOF}}$ , and estimated rate of convergence, for the second- and fourth-order versions of each method, applied to Test Problem 1A (TP-1A). See Table 1 for more details



**Fig. 7** Log–log plot of absolute error (59) versus  $\sqrt{\text{DOF}}$ , and estimated rate of convergence, for the second- and fourth-order versions of each method, applied to Test Problem 3A (TP-3A). See Table 2 for more details

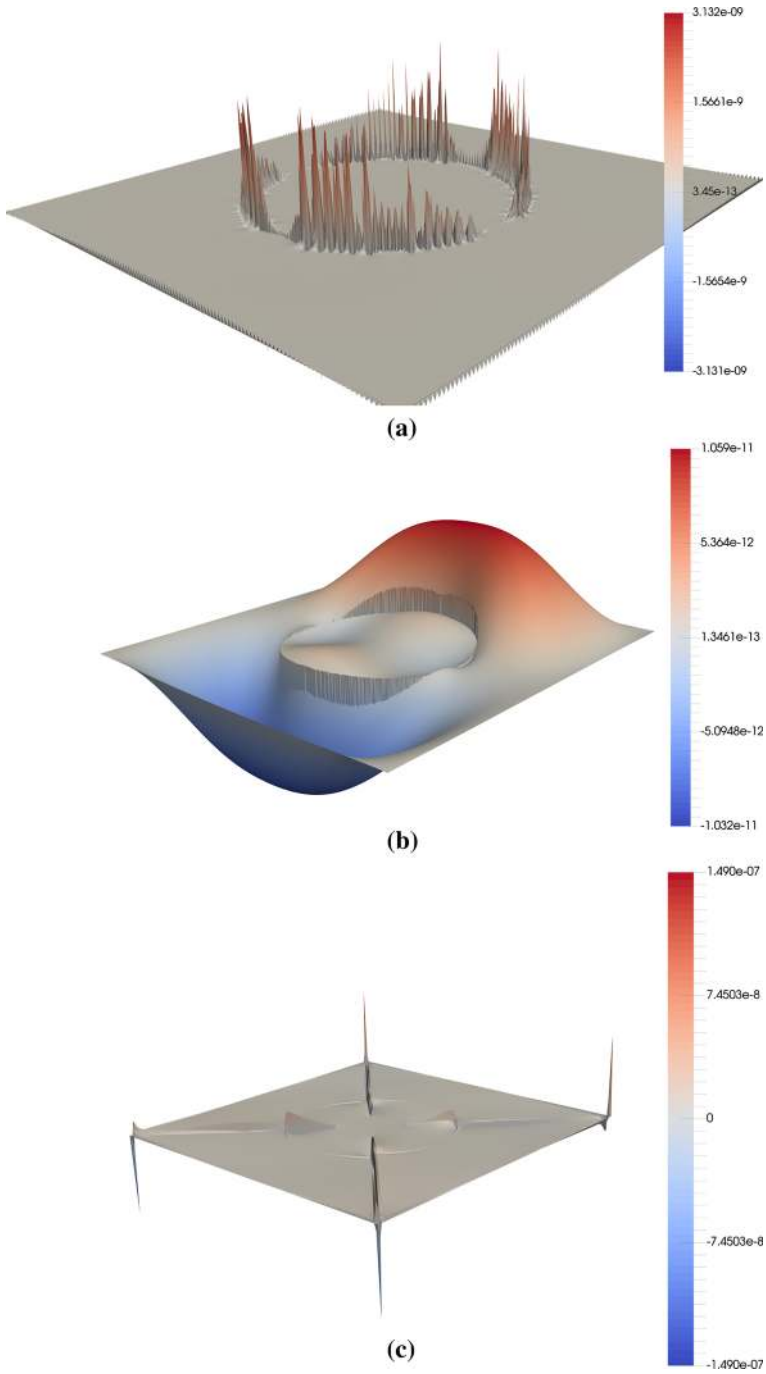
Table 2 and Fig. 7 show the numerical results for (TP-3A). This test problem has the same manufactured solution as (TP-1A), but with a time-varying diffusion coefficient. Despite this added complexity, the numerical results are the same order of accuracy, and in many cases the errors are the same up to seven digits, when compared with the results for (TP-1A). This similarity in the numerical results demonstrates that the three methods can robustly handle time-varying diffusion coefficients.

The plots of spatial error at the final time  $T = 1.0$ , shown in Fig. 8, are representative of other tests (not included in this text) on a single circular domain. The error in the cut-FEM solution presents largely at the boundary; the error in the DPM solution typically has smooth

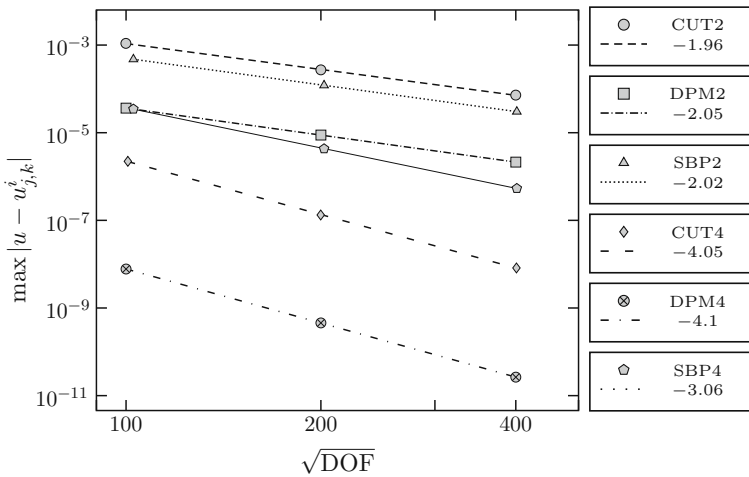


**Fig. 8** Plot of error at the final time  $T = 1.0$ , for the fourth-order versions of (a) cut-FEM, (b) DPM, and (c) SBP-SAT-FD, respectively, applied to Test Problem 3A (TP-3A)

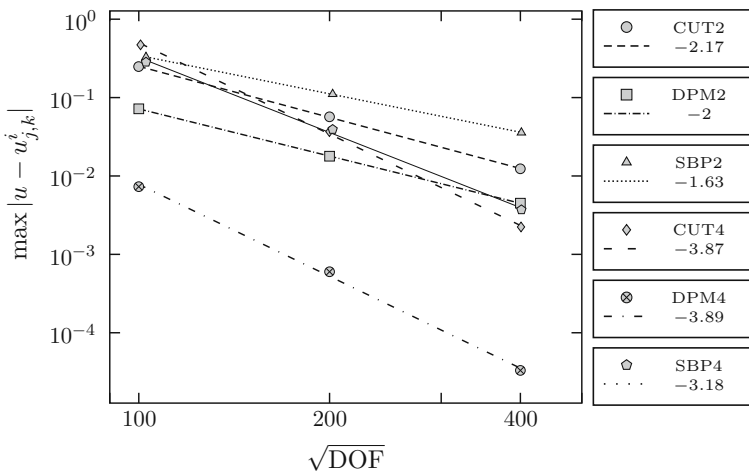




**Fig. 9** Plot of error at the final time  $T = 1.0$ , for the fourth-order versions of (a) cut-FEM, (b) DPM, and (c) SBP-SAT-FD, respectively, applied to Test Problem 2A (TP-2A)



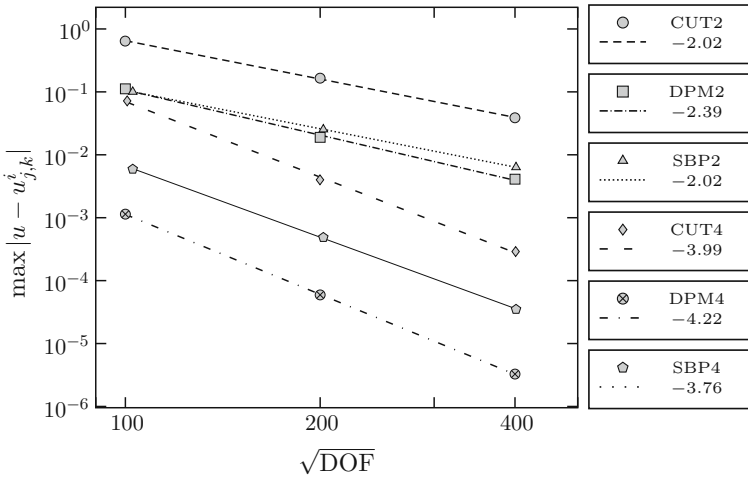
**Fig. 10** Log–log plot of absolute error (60) versus  $\sqrt{\text{DOF}}$ , and estimated rate of convergence, for the second- and fourth-order versions of each method (cut-FEM, DPM with explicit geometry, SBP–SAT–FD), applied to Test Problem 2A (TP–2A). See Table 3 for more details



**Fig. 11** Log–log plot of absolute error (60) versus  $\sqrt{\text{DOF}}$ , and estimated rate of convergence, for the second- and fourth-order versions of each method (cut-FEM, DPM with explicit geometry, SBP–SAT–FD), applied to Test Problem 2B (TP–2B). See Table 4 for more details

error, even for grid points very near  $\Gamma$ ; while the error in the SBP–SAT–FD solution is not smooth at interfaces introduced by the domain partitioning.

The plots of spatial error at the final time  $T = 1.0$  for (TP–2A) are shown in Fig. 9. These plots are fairly representative of the other composite domain tests reported herein, and also of others test problems not included in this work. As in Fig. 8, the cut-FEM has its largest error at degrees of freedom on cut (intersected) elements; the DPM has piecewise smooth error, including even grid points at the boundary/interface; and the SBP–SAT–FD has its largest error at the interfaces between computational subdomains, with particularly pronounced error at the corners of  $\Omega$ , where the grid is most stretched.



**Fig. 12** Log–log plot of absolute error (60) versus  $\sqrt{\text{DOF}}$ , and estimated rate of convergence, for the second- and fourth-order versions of each method (cut-FEM, DPM with explicit geometry, SBP–SAT–FD), applied to Test Problem 2C (TP–2C). See Table 5 for more details

Regarding the max-norm error in presented in Table 3 and Fig. 10, we see that the DPM has smaller max-norm by more than an order of magnitude. We also observe that the convergence rate of the fourth-order SBP–SAT–FD is only three. This suboptimal convergence is inline with the error plot in Fig. 9c, which shows that the error at the corners of the domain is significantly larger than elsewhere. In addition, the error is only non-smooth along the interfaces on the two diagonal lines of the domain. We have also measured the  $L_2$  error at the final time  $T = 1.0$  (not reported in this work), and fourth-order convergence is obtained.

In Table 4 and Fig. 11, we see the numerical results for (TP–2B). The analytical solution is similar to (TP–2A), though much more oscillatory—this additional challenge is manifested by an increase in error by several orders of magnitude.

In Table 5 and Fig. 12, we see the numerical results for (TP–2C), which shows that our numerical methods are robust to large jumps in diffusion coefficients, the analytical solution, and/or the flux of the true solution. Also, observe that the errors from DPM2/DPM4 (explicit geometry) and DPM2-I/DPM4-I (implicit geometry) in Tables 3, 4 and 5 are almost identical, which demonstrates the robustness and flexibility of the DPM.

## 6 Discussion

There are many possible methods (Sect. 1) for the numerical approximation of PDE posed on irregular domains, or on composite domains with interfaces. In this work, we consider three such methods, designed for the high-order accurate numerical approximation of parabolic PDEs (1, 2 or 3–9). Each implementation was written, tested, and optimized by the authors most experienced with the method—the cut-Finite Element Method (cut-FEM) by G. Ludwigsson, S. Sticks, G. Kreiss; the Difference Potentials Method (DPM) by K. R. Steffen, Q. Xia, Y. Epshteyn; and the Finite Difference Method satisfying Summation-By-Parts, with a Simultaneous Approximation Term (SBP–SAT–FD) by S. Wang, G. Kreiss. Although we consider only one type of boundary/interface (a circle), we hope that the benchmark problems

considered will be a valuable resource, and the numerical results a valuable comparison, for researchers interested in numerical methods for such problems.

The primary differences between the cut-FEM and the standard finite element method are the stabilization terms for near-boundary degrees of freedom, and the quadrature over cut (intersected) elements. Tuning the free parameters in the stabilization terms could mitigate the errors observed in Figs. 8, 9. (We have done some preliminary experiments suggesting that the errors decrease when tuning these parameters, but further investigations are required in order to guarantee robustness.) Given a level-set description of the geometry, there are robust algorithms for constructing the quadrature over cut elements. Together, these differences allow for an immersed (non-conforming) grid to be used. The theoretical base for cut-FEM is well established.

The DPM is based on the equivalence between the discrete system of Eq. (33) and the Boundary Equations with Projection (Theorem 1). The formulation outlined in Sect. 3.2 allows for an immersed (non-conforming) grid; fast  $\mathcal{O}(N \log N)$  algorithms, even for problems with general, smooth geometry; and reduces the size of the system to be solved at each time-step. The convergence theory is well-established for general, linear, elliptic boundary value problems, and we conjecture in Sect. 3.2 that this extends to the current setting. In this work, we have extended DPM to work with implicitly-defined geometries for the first time. This is a first step for solving problems where the interface moves with time.

In the finite difference framework (the SBP–SAT–FD method, in this work), the SBP property makes it possible to prove stability and convergence for high-order methods by an energy method. Combined with the SAT method to impose boundary and interface conditions, the SBP–SAT–FD method can be efficient to solve time-dependent PDE. Geometrical features are resolved by curvilinear mapping, which requires an explicit parameterization of boundaries and interfaces. High quality grid generation is important—our experiments, though not reported in this work, have shown that the error in the solution is sensitive to both the orthogonality of the grid and the grid stretching.

Similarities between the cut-FEM and the DPM (beyond the use of an immersed grid) include the thin layer of cut cells along the boundaries/interfaces (cut-FEM) and the discrete grid boundary  $\gamma$  (DPM); and the use of higher-order normal derivatives in the stabilization term (cut-FEM) and extension operator (in the Boundary Equations with Projection; DPM). A similarity between the cut-FEM and SBP–SAT–FD is the weak imposition of boundary conditions, via Nitsche’s method (cut-FEM) or the SAT method (SBP–SAT–FD). In this work, the DPM and the SBP–SAT–FD method both use an underlying finite-difference discretization, but the DPM is not restricted to this type of discretization.

Although both the cut-FEM and the DPM use higher-order normal derivatives in their treatment of the boundary/interface, the precise usage differs. For cut-FEM, it is the normal of the element interfaces cut by  $\Gamma$ , while for DPM, it is the normal of the boundary/interface  $\Gamma$ . Moreover, in the cut-FEM, stabilization terms (19) involving higher-order normal derivatives at the boundaries of cut-elements are added to the weak form of the PDE, to control the condition number of the mass and stiffness matrices, with a priori estimation of parameters to guarantee positive-definiteness of these matrices; while in the DPM, the Boundary Equations with Projection is combined with the Extension Operator (Definition 4), which incorporates higher-order normal derivatives at the boundary/interface  $\Gamma$ .

Returning to Sect. 5.3, we see (in Tables 1, 2, 3, 4, 5 and Figs. 6, 7, 8, 9, 10, 11, 12) that the expected rate of convergence for the second- and fourth-order versions of DPM and cut-FEM is achieved, while the DPM has the smallest error constant across all tests. For the SBP–SAT–FD method, expected convergence rates are obtained in some experiments. A

noticeable exception is Test Problem 2A, for which the fourth-order SBP–SAT–FD method only has a convergence rate of three.

From the error plot in Fig. 9c, we observe that the large error is localized at the four corners of the domain  $\Omega$ , where the curvilinear grid is non-orthogonal and is stretched the most (see Fig. 5b).

As seen in the error plots (Figs. 8, 9), the error for the cut-FEM and the SBP–SAT–FD has “spikes”, while for the DPM the error is smooth. A surprising observation from Fig. 9 is that conforming grids (on which the SBP–SAT–FD method is designed) do not necessarily produce more accurate solutions than immersed grids (on which the cut-FEM and the DPM are designed). Indeed, it is challenging to construct a high-quality curvilinear grid for the considered composite domain problem.

Future directions we hope to consider (in the context of new developments and also further comparisons) include: (i) parabolic problems with moving boundaries/interfaces, (ii) comparison of numerical methods for interface problems involving wave equations [12, 70, 71, 75, 78], (iii) extending our methods to consider PDEs in 3D, (iv) design of fast algorithms, and (v) design of adaptive versions of our methods.

Indeed, for (i), difficulties for the cut-FEM might be the costly construction of quadrature, while for DPM difficulties might be the accurate construction of extension operators. Regarding (iii), this has already been done for the cut-FEM and SBP–SAT–FD; while for the DPM, this is current work, with the main steps extending from 2D to 3D in a straightforward manner.

## 7 Conclusion

In this work, we propose a set of benchmark problems to test numerical methods for parabolic partial differential equations in irregular or composite domains, in the simplified geometric setting of Sect. 2, with the interface defined either explicitly or implicitly. Next, we compare and contrast three methods for the numerical approximation of such problems: the (i) cut-FEM; (ii) DPM; and (iii) SBP–SAT–FD. Brief introductions of the three numerical methods are given in Sect. 3. It is noteworthy that the DPM has, for the first time, been extended to problems with an implicitly-defined interface.

For the three methods, the numerical results in Sect. 5.3 illustrate the high-order accuracy. Similar errors (different by a constant factor) are observed at grid points away from the boundary/interface, while the observed errors near the boundary/interface vary depending upon the given method. Although we consider only test problems with circular boundary/interface, the ideas underlying the three methods can readily be extended to more general geometries.

In general, all three methods require an accurate and efficient resolution of the explicitly- or implicitly-defined irregular geometry: cut-FEM relies on accurate quadrature rules for cut elements, and a good choice of stabilization parameters; DPM relies on an accurate and efficient representation of Cauchy data using a good choice of basis functions; and SBP–SAT–FD relies on the smooth parametrization to generate a high-quality curvilinear grid.

**Acknowledgements** The authors are grateful to the anonymous referees for their valuable remarks and questions, which led to significant improvements to the manuscript. We gratefully acknowledge the support of the Swedish Research Council (Grant No. 2014-6088); the Swedish Foundation for International Cooperation in Research and Higher Education (Grant No. STINT-IB2016-6512); Uppsala University, Department of Information Technology; and the University of Utah, Department of Mathematics. Y. Epshteyn, K. R. Steffen, and Q. Xia also acknowledge partial support of Simons Foundation Grant No. 415673.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## 8 Appendix (DPM)

Let us now expand some details presented in the brief introduction to the Difference Potentials Method (Sect. 3.2).

### 8.1 Fully-Discrete Formulation of (3, 4)

The fully-discrete, finite-difference discretization introduced in (33) is

$$L_{\Delta t, h}^s u_s^{i+1} = F_s^{i+1}, \quad (x_j, y_k) \in M_s^+. \tag{62}$$

The general form of the operator is  $L_{\Delta t, h}^s = \lambda_s(t^{i+1})\Delta_h - \sigma I$ , where  $\sigma = \frac{3}{2\Delta t}$  for second-order (BDF2–DPM2),  $\sigma = \frac{25}{12\Delta t}$  for fourth-order (BDF4–DPM4), and  $\Delta_h$  is either a standard five- or nine-point Laplacian. For the nine-point Laplacian, we have

$$\begin{aligned} \Delta_h u_{j,k} = & \frac{1}{12h^2} (-u_{j-2,k} + 16u_{j-1,k} + 16u_{j+1,k} - u_{j+2,k} - u_{j,k-2} \\ & + 16u_{j,k-1} + 16u_{j,k+1} - u_{j,k+2} - 60u_{j,k}) \end{aligned} \tag{63}$$

for points sufficiently far away from the boundary of the auxiliary domain  $\Omega_s^0$ . For points that are close to the boundary, we use a modified, fourth-order stencil. For example, at the southwest corner, we take

$$\begin{aligned} \Delta_h u_{1,1} = & \frac{1}{12h^2} (10u_{0,1} - 4u_{2,1} + 14u_{3,1} - 6u_{4,1} + u_{5,1} \\ & + 10u_{1,0} - 4u_{1,2} + 14u_{1,3} - 6u_{1,4} + u_{1,5} - 30u_{1,1}), \end{aligned} \tag{64}$$

where  $u_{0,1}$  and  $u_{1,0}$  will be from the boundary condition (7).

Next, the right-hand side of (33) for BDF2–DPM2 is given by

$$F_s^{i+1} = -f_s^{i+1} - \frac{\sigma}{3}(4u_s^i - u_s^{i-1}), \tag{65}$$

and for BDF4–DPM4 by

$$F_s^{i+1} = -f_s^{i+1} - \frac{\sigma}{25}(48u_s^i - 36u_s^{i-1} + 16u_s^{i-2} - 3u_s^{i-3}). \tag{66}$$

Lastly, the initialization at  $t = 0$  is done using the exact solutions for the terms  $u_s^0, u_s^{-1}, u_s^{-2}$ , and  $u_s^{-3}$ . Another possibility would be the use of lower-order BDF methods, then advancing to higher-order methods once enough stages are established. No significant differences were observed between the two approaches.

### 8.2 Equation-Based Extension

Let us now expand the discussion surrounding (39–41) leading up to Definition 4 of the Extension Operator (42).

An important step in this discussion is to recast the original PDE (3, 4) into a curvilinear form, for points  $(x, y)$  in the vicinity of  $\Gamma$ . Following the notation [58], let us first introduce

the coordinate system  $(d, \vartheta)$  for points in the vicinity of  $\Gamma$ . Recall from Definition 42 that  $d$  is the distance in the normal direction from a given point to its orthogonal projection on  $\Gamma$ , while  $\vartheta$  is the arclength along  $\Gamma$  from some reference point to the orthogonal projection. In this coordinate system, the PDE (3, 4) becomes

$$\frac{\partial u_s}{\partial t} - \lambda_s \left( \frac{1}{H_\vartheta} \left[ \frac{\partial}{\partial n} \left( H_\vartheta \frac{\partial u_s}{\partial n} \right) + \frac{\partial}{\partial \vartheta} \left( \frac{1}{H_\vartheta} \frac{\partial u_s}{\partial \vartheta} \right) \right] \right) = f_s, \tag{67}$$

where where  $H_\vartheta = 1 - d\kappa$  is the Lamé coefficient, and  $\kappa$  is the signed curvature along the interface  $\Gamma$ .

From (67), a straightforward calculation gives the second-order normal derivative  $\frac{\partial^2 u_s}{\partial n^2}$  (used in the calculation of (41)), which is

$$\frac{\partial^2 u_s}{\partial n^2} = \frac{1}{\lambda_s} \left( \frac{\partial u_s}{\partial t} - f_s \right) - \frac{\partial^2 u_s}{\partial \vartheta^2} + \kappa \frac{\partial u_s}{\partial n}. \tag{68}$$

For the fourth-order numerical method, which uses an Extension Operator with  $p = 4$ , we also need the third- and fourth-order normal derivatives, which we state now. Differentiating (68) with respect to  $n$ , we see that

$$\frac{\partial^3 u_s}{\partial n^3} = \frac{1}{\lambda_s} \left( \frac{\partial^2 u_s}{\partial t \partial n} - \frac{\partial f_s}{\partial n} \right) - \frac{\partial^3 u_s}{\partial n \partial \vartheta^2} + \kappa \frac{\partial^2 u_s}{\partial n^2} \tag{69}$$

and

$$\begin{aligned} \frac{\partial^4 u_s}{\partial n^4} &= \frac{1}{\lambda_s^2} \left( \frac{\partial^2 u_s}{\partial t^2} - \frac{\partial f_s}{\partial t} \right) \\ &\dots + \frac{1}{\lambda_s} \left( -2 \frac{\partial^3 u_s}{\partial t \partial \vartheta^2} + \kappa \frac{\partial^2 u_s}{\partial n \partial t} - \frac{\partial^2 f_s}{\partial n^2} + \frac{\partial^2 f_s}{\partial \vartheta^2} + \frac{\partial^4 u_s}{\partial \vartheta^4} - \kappa \frac{\partial^3 u_s}{\partial n \partial \vartheta^2} \right) + \kappa \frac{\partial^3 u_s}{\partial n^3}. \end{aligned} \tag{70}$$

Next, let us follow-up on comments made in the text following (41). There, it was pointed out that the unknown Dirichlet and Neumann data  $(u_s, \frac{\partial u_s}{\partial n})$  are the only data required for the Extension Operator (42) with  $p = 2$ . Moreover, it was pointed out that this is also true for the Extension Operator when  $p = 4$ . The reasoning is as follows.

- The time derivatives  $\frac{\partial u_s}{\partial t}, \frac{\partial^2 u_s}{\partial t^2}, \frac{\partial f_s}{\partial t}, \frac{\partial^2 u_s}{\partial n \partial t},$  and  $\frac{\partial^3 u_s}{\partial t \partial \vartheta^2}$  can be approximated by the backward difference formula. In BDF2–DPM2,

$$\frac{\partial u_s^{i+1}}{\partial t} \approx \frac{3u_s^{i+1} - 4u_s^i + u_s^{i-1}}{2\Delta t} \quad \text{and} \tag{71}$$

$$\frac{\partial^2 u_s^{i+1}}{\partial t^2} \approx \frac{2u_s^{i+1} - 5u_s^i + 4u_s^{i-1} - u_s^{i-2}}{\Delta t^2}, \tag{72}$$

while in BDF4–DPM4,

$$\frac{\partial u_s^{i+1}}{\partial t} \approx \frac{25u_s^{i+1} - 48u_s^i + 36u_s^{i-1} - 16u_s^{i-2} + 3u_s^{i-3}}{12\Delta t} \quad \text{and} \tag{73}$$

$$\frac{\partial^2 u_s^{i+1}}{\partial t^2} \approx \frac{1}{\Delta t^2} \left( \frac{15}{4}u_s^{i+1} - \frac{77}{6}u_s^i + \frac{107}{6}u_s^{i-1} - 13u_s^{i-2} + \frac{61}{12}u_s^{i-3} - \frac{5}{6}u_s^{i-4} \right). \tag{74}$$

- The derivatives in terms of arclength  $\vartheta$  can be computed from  $u_s$  or  $\frac{\partial u_s}{\partial n}$ . For example, denoting  $u_s = \sum_{v=1}^{\mathcal{N}^0} c_{1,v}^{s,i+1} \phi_v(\vartheta)$  (using notation following from (46)), then it comes handy that

$$\frac{\partial u_s}{\partial \vartheta} = \sum_{v=1}^{\mathcal{N}^0} c_{1,v}^{s,i+1} \phi'_v(\vartheta), \quad \frac{\partial^2 u_s}{\partial \vartheta^2} = \sum_{v=1}^{\mathcal{N}^0} c_{1,v}^{s,i+1} \phi''_v(\vartheta), \quad \frac{\partial^4 u_s}{\partial \vartheta^4} = \sum_{v=1}^{\mathcal{N}^0} c_{1,v}^{s,i+1} \phi_v^{(4)}(\vartheta). \tag{75}$$

### 8.3 The System of Equations at Each Time Step

With the Cauchy data  $u_{s,\Gamma}^{i+1}$  and Extension Operator  $\text{Ex}_s u_{s,\Gamma}^{i+1}$  from  $\Gamma$  to  $\gamma_s$  introduced in Definition 4, and the spectral representation introduced in (46), we now give a sketch of the linear system for the coefficients  $(c_{1,v}^{s,i+1})_{v=1}^{\mathcal{N}^0}$  and  $(c_{2,v}^{s,i+1})_{v=1}^{\mathcal{N}^1}$ , and moreover the approximation of the solution  $u_s(x, y, t^{i+1})$  at  $(x_j, y_k) \in N_s^+$ .

Indeed, substituting  $\text{Ex}_s \tilde{u}_{s,\Gamma}^{i+1}$  (42, 46) into the BEP (38), the resulting linear systems are

$$\sum_{k=0}^1 \sum_{v=1}^{\mathcal{N}^k} \left( c_{k+1,v}^{s,i+1} \text{Ex}_s \Phi_v^k - c_{k+1,v}^{s,i+1} P_{\gamma_s}^{i+1} \text{Ex}_s \Phi_v^k \right) = \text{Tr}_{\gamma_s} [G_{\Delta t,h}^{i+1} F_s^{i+1}]. \tag{76}$$

This can be further elucidated by introducing the vector of unknowns

$$\mathbf{c}_s^{i+1} = \left[ \underbrace{c_{1,1}^{s,i+1} \quad c_{1,2}^{s,i+1} \quad \dots \quad c_{1,\mathcal{N}^0}^{s,i+1}}_{\mathbf{c}_{s,1}^{i+1}} \quad \underbrace{c_{2,1}^{s,i+1} \quad c_{2,2}^{s,i+1} \quad \dots \quad c_{2,\mathcal{N}^1}^{s,i+1}}_{\mathbf{c}_{s,2}^{i+1}} \right]^\top \tag{77}$$

(so that  $\mathbf{c}_s^{i+1} = [\mathbf{c}_{s,1}^{i+1}, \mathbf{c}_{s,2}^{i+1}]^\top$ ), and the matrix

$$A_s = \left[ \underbrace{\left( (I - P_{\gamma_s}^{i+1}) \text{Ex}_s \Phi_1^0, \quad (I - P_{\gamma_s}^{i+1}) \text{Ex}_s \Phi_2^0, \quad \dots \quad (I - P_{\gamma_s}^{i+1}) \text{Ex}_s \Phi_{\mathcal{N}^0}^0 \right)}_{A_{s,1}}, \right. \\ \left. \underbrace{\left( (I - P_{\gamma_s}^{i+1}) \text{Ex}_s \Phi_1^1, \quad (I - P_{\gamma_s}^{i+1}) \text{Ex}_s \Phi_2^1, \quad \dots \quad (I - P_{\gamma_s}^{i+1}) \text{Ex}_s \Phi_{\mathcal{N}^1}^1 \right)}_{A_{s,2}} \right]. \tag{78}$$

Then, the full system of Eq. (76) is

$$A \begin{bmatrix} \mathbf{c}_1^{i+1} \\ \mathbf{c}_2^{i+1} \end{bmatrix} = \begin{bmatrix} \text{Tr}_{\gamma_1} [G_{\Delta t,h}^{i+1} F_1^{i+1}] \\ \text{Tr}_{\gamma_2} [G_{\Delta t,h}^{i+1} F_2^{i+1}] \end{bmatrix}, \quad \text{with } A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}. \tag{79}$$

However, note that  $\mathbf{c}_1^{i+1}$  and  $\mathbf{c}_2^{i+1}$  are related by the interface conditions (8, 9), so that the number of unknowns in (79) is equal the dimension of either  $\mathbf{c}_1^{i+1}$  or  $\mathbf{c}_2^{i+1}$ , depending on which one is considered the independent unknown. Therefore, the dimension of  $A$  is  $(|\gamma_1| + |\gamma_2|) \times (\mathcal{N}^0 + \mathcal{N}^1)$ , where  $\mathcal{N}^0 + \mathcal{N}^1$  is the dimension of  $\mathbf{c}_1^{i+1}$  or  $\mathbf{c}_2^{i+1}$  (whichever is the independent unknown).

*Remark 8* The independent unknown ( $\mathbf{c}_1^{i+1}$  or  $\mathbf{c}_2^{i+1}$ ) is chosen so that the finite-dimensional, spectral representation (46) of the Cauchy data  $u_{s,\Gamma}^{i+1}$  accurately resolves the Cauchy data with a small number of basis functions, in the consideration of both accuracy and computational efficiency. For (TP–2A) and (TP–2B), we choose  $\mathbf{c}_2^{i+1}$  as the independent unknown, while



for (TP–2C) we choose  $\mathbf{e}_1^{i+1}$ . With these choices for the independent unknown, we have  $\mathcal{N}^0 = \mathcal{N}^1 = 1$  for the three considered test problems.

Since each column involves the Difference Potentials operator  $P_{\gamma_s}^{i+1}$  applied to a vector  $\mathbf{e}_{\gamma_s}^k \Phi_{\gamma_s}^k$ , each column is therefore constructed *via* one solution of the Auxiliary Problem (Definition 1). However, the Auxiliary Problems are posed on the computationally simple Auxiliary Domains, and can be computed using a fast FFT- or multigrid-based algorithm, which can significantly reduce the computational cost. Moreover, if  $\lambda_s(t) \equiv \lambda_s$  is constant, then  $A$  can be computed and inverted once (as a pre-processing step), thus significantly reducing computational cost for long-time simulations.

## References

1. Abarbanel, S., Ditkowski, A.: Asymptotically stable fourth-order accurate schemes for the diffusion equation on complex shapes. *J. Comput. Phys.* **133**, 279–288 (1997). <https://doi.org/10.1006/jcph.1997.5653>
2. Adams, L., Li, Z.: The immersed interface/multigrid methods for interface problems. *SIAM J. Sci. Comput.* **24**(2), 463–479 (2002). <https://doi.org/10.1137/S1064827501389849>
3. Albright, J.: Numerical Methods Based on Difference Potentials for Models with Material Interfaces. Ph.D. thesis, University of Utah (2016)
4. Albright, J., Epshteyn, Y., Medvinsky, M., Xia, Q.: High-order numerical schemes based on difference potentials for 2D elliptic problems with material interfaces. *Appl. Numer. Math.* **111**, 64–91 (2017). <https://doi.org/10.1016/j.apnum.2016.08.017>
5. Albright, J., Epshteyn, Y., Steffen, K.R.: High-order accurate difference potentials methods for parabolic problems. *Appl. Numer. Math.* **93**, 87–106 (2015). <https://doi.org/10.1016/j.apnum.2014.08.002>
6. Albright, J., Epshteyn, Y., Xia, Q.: High-order accurate methods based on difference potentials for 2D parabolic interface models. *Commun. Math. Sci.* **15**(4), 985–1019 (2017). <https://doi.org/10.4310/CMS.2017.v15.n4.a4>
7. Almqvist, M., Karasalo, I., Mattsson, K.: Atmospheric sound propagation over large-scale irregular terrain. *J. Sci. Comput.* **61**, 369–397 (2014). <https://doi.org/10.1007/s10915-014-9830-4>
8. Appelö, D., Petersson, N.A.: A stable finite difference method for the elastic wave equation on complex geometries with free surfaces. *Commun. Comput. Phys.* **5**(1), 84–107 (2009)
9. Bedrossian, J., Von Brecht, J.H., Zhu, S., Sifakis, E., Teran, J.M.: A second order virtual node method for elliptic problems with interfaces and irregular domains. *J. Comput. Phys.* **229**(18), 6405–6426 (2010). <https://doi.org/10.1016/j.jcp.2010.05.002>
10. Berg, J., Nordström, J.: Spectral analysis of the continuous and discretized heat and advection equation on single and multiple domains. *Appl. Numer. Math.* **62**, 1620–1638 (2012). <https://doi.org/10.1016/j.apnum.2012.05.002>
11. Bouchon, M., Campillo, M., Gaffet, S.: A boundary integral equation-discrete wavenumber representation method to study wave propagation in multilayered media having irregular interfaces. *Geophysics* **54**(9), 1134–1140 (1989). <https://doi.org/10.1190/1.1442748>
12. Britt, S., Tsynkov, S., Turkel, E.: Numerical solution of the wave equation with variable wave speed on nonconforming domains by high-order difference potentials. *J. Comput. Phys.* (2017). <https://doi.org/10.1016/j.jcp.2017.10.049>
13. Burman, E., Claus, S., Hansbo, P., Larson, M.G., Massing, A.: CutFEM: Discretizing geometry and partial differential equations. *Int. J. Numer. Methods Eng.* **104**(7), 472–501 (2015). <https://doi.org/10.1002/nme.4823>
14. Burman, E., Hansbo, P.: Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. *Appl. Numer. Math.* **62**(4), 328–341 (2012). <https://doi.org/10.1016/j.apnum.2011.01.008>
15. Burman, E., Hansbo, P., Larson, M.G., Zahedi, S.: Cut finite element methods for coupled bulksurface problems. *Numer. Math.* **133**(2), 203–231 (2016). <https://doi.org/10.1007/s00211-015-0744-3>
16. Carpenter, M.H., Gottlieb, D., Abarbanel, S.: Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes. *J. Comput. Phys.* **111**, 220–236 (1994). <https://doi.org/10.1006/jcph.1994.1057>

17. Carpenter, M.H., Nordström, J., Gottlieb, D.: Revisiting and extending interface penalties for multi-domain summation-by-parts operators. *J. Sci. Comput.* **45**(1–3), 118–150 (2009). <https://doi.org/10.1007/s10915-009-9301-5>
18. Coco, A., Russo, G.: Second order multigrid methods for elliptic problems with discontinuous coefficients on an arbitrary interface, I: one dimensional problems. *Numer. Math. Theory Methods Appl.* **5**(01), 19–42 (2012). <https://doi.org/10.4208/nmtma.2011.m12si02>
19. Crockett, R., Colella, P., Graves, D.T.: A Cartesian grid embedded boundary method for solving the Poisson and heat equations with discontinuous coefficients in three dimensions. *J. Comput. Phys.* **230**(7), 2451–2469 (2011). <https://doi.org/10.1016/j.jcp.2010.12.017>
20. Del Rey Fernández, D.C., Boom, P.D., Zingg, D.W.: A generalized framework for nodal first derivative summation-by-parts operators. *J. Comput. Phys.* **266**, 214–239 (2014). <https://doi.org/10.1016/j.jcp.2014.01.038>
21. Del Rey Fernández, D.C., Hicken, J.E., Zingg, D.W.: Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. *Comput. Fluids* **95**, 171–196 (2014). <https://doi.org/10.1016/j.compfluid.2014.02.016>
22. Demirdžić, I., Muzaferija, S.: Finite volume method for stress analysis in complex domains. *Int. J. Numer. Methods. Eng.* **37**(21), 3751–3766 (1994). <https://doi.org/10.1002/nme.1620372110>
23. Ditkowski, A., Harness, Y.: High-order embedded finite difference schemes for initial boundary value problems on time dependent irregular domains. *J. Sci. Comput.* **39**, 394–440 (2009). <https://doi.org/10.1007/s10915-009-9277-1>
24. Duru, K., Virta, K.: Stable and high order accurate difference methods for the elastic wave equation in discontinuous media. *J. Comput. Phys.* **279**, 37–62 (2014). <https://doi.org/10.1016/j.jcp.2014.08.046>
25. Epshteyn, Y.: Upwind-difference potentials method for Patlak–Keller–Segel chemotaxis model. *J. Sci. Comput.* **53**(3), 689–713 (2012). <https://doi.org/10.1007/s10915-012-9599-2>
26. Epshteyn, Y.: Algorithms composition approach based on difference potentials method for parabolic problems. *Commun. Math. Sci.* **12**(4), 723–755 (2014). <https://doi.org/10.4310/cms.2014.v12.n4.a7>
27. Epshteyn, Y., Medvinsky, M.: On the solution of the elliptic interface problems by difference potentials method. In: Kirby, R.M., Berzins, M., Hesthaven, J.S. (eds.) *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, pp. 197–205. Springer (2015). [https://doi.org/10.1007/978-3-319-19800-2\\_16](https://doi.org/10.1007/978-3-319-19800-2_16)
28. Epshteyn, Y., Phippen, S.: High-order difference potentials methods for 1D elliptic type models. *Appl. Numer. Math.* **93**, 69–86 (2015). <https://doi.org/10.1016/j.apnum.2014.02.005>
29. Epshteyn, Y., Sofronov, I., Tsynkov, S.: Professor V. S. Ryaben'kii. On the occasion of the 90-th birthday. *Appl. Numer. Math.* **93**(Supplement C), 1–2 (2015). <https://doi.org/10.1016/j.apnum.2015.02.001>
30. Fadlun, E., Verzicco, R., Orlandi, P., Mohd-Yusof, J.: Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *J. Comput. Phys.* **161**(1), 35–60 (2000). <https://doi.org/10.1006/jcph.2000.6484>
31. Fedkiw, R.P., Aslam, T., Merriman, B., Osher, S.: A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys.* **152**(2), 457–492 (1999). <https://doi.org/10.1006/jcph.1999.6236>
32. Gibou, F., Fedkiw, R.: A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem. *J. Comput. Phys.* **202**(2), 577–601 (2005). <https://doi.org/10.1016/j.jcp.2004.07.018>
33. Godunov, S.K., Zhukov, V.T., Lazarev, M.I., Sofronov, I.L., Turchaninov, V.I., Kholodov, A.S., Tsynkov, S.V., Chetverushkin, B.N., Epshteyn, Y.Y.: Viktor Solomonovich Ryaben'kii and his school (on his 90th birthday). *Russ. Math. Surv.* **70**(6), 1183 (2015). <https://doi.org/10.1070/RM2015v070n06ABEH004981>
34. Gong, J., Xuan, L., Ming, P., Zhang, W.: An unstructured finite-volume method for transient heat conduction analysis of multilayer functionally graded materials with mixed grids. *Numer. Heat Transfer, Part B* **63**(3), 222–247 (2013). <https://doi.org/10.1080/10407790.2013.751251>
35. Guittet, A., Lepilliez, M., Tanguy, S., Gibou, F.: Solving elliptic problems with discontinuities on irregular domains—the Voronoi interface method. *J. Comput. Phys.* **298**, 747–765 (2015). <https://doi.org/10.1016/j.jcp.2015.06.026>
36. Guittet, A., Poignard, C., Gibou, F.: A Voronoi interface approach to cell aggregate electropermeabilization. *J. Comput. Phys.* **332**, 143–159 (2017). <https://doi.org/10.1016/j.jcp.2016.11.048>
37. Hansbo, A., Hansbo, P.: An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Comput. Methods Appl. Mech. Eng.* **191**(47), 5537–5552 (2002). [https://doi.org/10.1016/S0045-7825\(02\)00524-8](https://doi.org/10.1016/S0045-7825(02)00524-8)
38. Hansbo, P., Larson, M.G., Zahedi, S.: A cut finite element method for a Stokes interface problem. *Appl. Numer. Math.* **85**, 90–114 (2014). <https://doi.org/10.1016/j.apnum.2014.06.009>

39. Hellrung, J.L., Wang, L., Sifakis, E., Teran, J.M.: A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions. *J. Comput. Phys.* **231**(4), 2015–2048 (2012). <https://doi.org/10.1016/j.jcp.2011.11.023>
40. Hesthaven, J.S., Warburton, T.: *Nodal Discontinuous Galerkin Methods*. Springer, Berlin (2008). <https://doi.org/10.1007/978-0-387-72067-8>
41. Johansen, H., Colella, P.: A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. *J. Comput. Phys.* **147**(1), 60–85 (1998). <https://doi.org/10.1006/jcph.1998.5965>
42. Kim, J., Kim, D., Choi, H.: An immersed-boundary finite-volume method for simulations of flow in complex geometries. *J. Comput. Phys.* **171**(1), 132–150 (2001). <https://doi.org/10.1006/jcph.2001.6778>
43. Knupp, P., Steinberg, S.: *Fundamentals of Grid Generation*. CRC Press, Boca Raton (1993)
44. Kozydon, J.E., Warburton, L.C.: Stable coupling of nonconforming, high-order finite difference methods. *SIAM J. Sci. Comput.* **38**, A923–A952 (2016). <https://doi.org/10.1137/15M1022823>
45. Kreiss, H.O., Scherer, G.: Finite element and finite difference methods for hyperbolic partial differential equations. In: *Mathematical Aspects of Finite Elements in Partial Differential Equations*, Symposium Proceedings pp. 195–212 (1974). <https://doi.org/10.1016/B978-0-12-208350-1.50012-1>
46. LeVeque, R.J., Li, Z.: The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.* **31**(4), 1019–1044 (1994). <https://doi.org/10.1137/0731054>
47. LeVeque, R.J., Li, Z.: Immersed interface methods for Stokes flow with elastic boundaries or surface tension. *SIAM J. Sci. Comput.* **18**(3), 709–735 (1997). <https://doi.org/10.1137/S1064827595282532>
48. Li, Z., Ito, K.: The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains. *Front. Appl. Math. Soc. Ind. Appl. Math.* (2006). <https://doi.org/10.1137/1.9780898717464>
49. Linnick, M.N., Fasel, H.F.: A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *J. Comput. Phys.* **204**(1), 157–192 (2005). <https://doi.org/10.1016/j.jcp.2004.09.017>
50. Liu, T., Khoo, B., Yeo, K.: Ghost fluid method for strong shock impacting on material interface. *J. Comput. Phys.* **190**(2), 651–681 (2003). [https://doi.org/10.1016/S0021-9991\(03\)00301-2](https://doi.org/10.1016/S0021-9991(03)00301-2)
51. Liu, X.D., Sideris, T.: Convergence of the ghost fluid method for elliptic equations with interfaces. *Math. Comput.* **72**(244), 1731–1746 (2003). <https://doi.org/10.1090/S0025-5718-03-01525-4>
52. Magura, S., Petropavlovsky, S., Tsynkov, S., Turkel, E.: High-order numerical solution of the Helmholtz equation for domains with reentrant corners. *Appl. Numer. Math.* **118**, 87–116 (2017). <https://doi.org/10.1016/j.apnum.2017.02.013>
53. Massing, A., Larson, M.G., Logg, A., Rognes, M.E.: A stabilized Nitsche fictitious domain method for the Stokes problem. *J. Sci. Comput.* **61**(3), 604–628 (2014). <https://doi.org/10.1007/s10915-014-9838-9>
54. Mattsson, K.: Summation by parts operators for finite difference approximations of second-derivatives with variable coefficient. *J. Sci. Comput.* **51**, 650–682 (2012). <https://doi.org/10.1007/s10915-011-9525-z>
55. Mattsson, K., Carpenter, M.H.: Stable and accurate interpolation operators for high-order multiblock finite difference methods. *SIAM J. Sci. Comput.* **32**, 2298–2320 (2010). <https://doi.org/10.1137/090750068>
56. Mayo, A.: The fast solution of Poisson's and the biharmonic equations on irregular regions. *SIAM J. Sci. Comput.* **21**(2), 285–299 (1984). <https://doi.org/10.1137/0721021>
57. McCorquodale, P., Colella, P., Johansen, H.: A Cartesian grid embedded boundary method for the heat equation on irregular domains. *J. Comput. Phys.* **173**(2), 620–635 (2001). <https://doi.org/10.1006/jcph.2001.6900>
58. Medvinsky, M., Tsynkov, S., Turkel, E.: The method of difference potentials for the Helmholtz equation using compact high order schemes. *J. Sci. Comput.* **53**(1), 150–193 (2012). <https://doi.org/10.1007/s10915-012-9602-y>
59. Medvinsky, M., Tsynkov, S., Turkel, E.: Solving the Helmholtz equation for general smooth geometry using simple grids. *Wave Motion* **62**, 75–97 (2016). <https://doi.org/10.1016/j.wavemoti.2015.12.004>
60. Nitsche, J.: Über ein variationsprinzip zur lösung von Dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg* **36**(9), 9–15 (1971). <https://doi.org/10.1007/BF02995904>
61. Peskin, C.S.: The immersed boundary method. *Acta Numer.* **11**, 479–517 (2002). <https://doi.org/10.1017/S0962492902000077>
62. Reznik, A.A.: Approximation of surface potentials of elliptic operators by difference potentials. *Dokl. Akad. Nauk SSSR* **263**, 1318–1321 (1982)
63. Reznik, A.A.: *Approximation of the Surface Potentials of Elliptic Operators by Difference Potentials and the Solution of Boundary Value Problems*. Ph.D. thesis, Moscow Institute for Physics and Technology (1983)
64. Ryaben'kiĭ, V.S.: Boundary equations with projectors. *Uspekhi Mat. Nauk* **40**(2(242)), 121–149 (1985)

65. Ryaben'kii, V.S.: Method of Difference Potentials and Its Applications. Springer, Berlin (2002). <https://doi.org/10.1007/978-3-642-56344-7>
66. Ryaben'kiĭ, V.S.: Difference potentials analogous to Cauchy integrals. *Uspekhi Mat. Nauk* **67**(3(405)), 147–172 (2012)
67. Ryaben'kii, V.S., Turchaninov, V.I., Epshteyn, Y.Y.: Algorithm composition scheme for problems in composite domains based on the difference potential method. *Comp. Math. Math. Phys.* **46**(10), 1768–1784 (2006). <https://doi.org/10.1134/s0965542506100137>
68. Saye, R.I.: High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles. *SIAM J. Sci. Comput.* **37**(2), A993–A1019 (2015). <https://doi.org/10.1137/140966290>
69. Sethian, J.A., Wiegmann, A.: Structural boundary design via level set and immersed interface methods. *J. Comput. Phys.* **163**(2), 489–528 (2000). <https://doi.org/10.1006/jcph.2000.6581>
70. Sticko, S.: Towards Higher Order Immersed Finite Elements for the Wave Equation. Licentiate thesis, Uppsala University, Division of Scientific Computing (2016)
71. Sticko, S., Kreiss, G.: A stabilized Nitsche cut element method for the wave equation. *Comput. Methods Appl. Mech. Eng.* **309**, 364–387 (2016). <https://doi.org/10.1016/j.cma.2016.06.001>
72. Strand, B.: Summation by parts for finite difference approximations for  $d/dx$ . *J. Comput. Phys.* **110**, 47–67 (1994). <https://doi.org/10.1006/jcph.1994.1005>
73. Svård, M., Nordström, J.: Review of summation-by-parts schemes for initial-boundary-value problems. *J. Comput. Phys.* **268**, 17–38 (2014). <https://doi.org/10.1016/j.jcp.2014.02.031>
74. Tseng, Y.H., Ferziger, J.H.: A ghost-cell immersed boundary method for flow in complex geometry. *J. Comput. Phys.* **192**(2), 593–623 (2003). <https://doi.org/10.1016/j.jcp.2003.07.024>
75. Virta, K., Mattsson, K.: Acoustic wave propagation in complicated geometries and heterogeneous media. *J. Sci. Comput.* **61**, 90–118 (2014). <https://doi.org/10.1007/s10915-014-9817-1>
76. Wadbro, E., Zahedi, S., Kreiss, G., Berggren, M.: A uniformly well-conditioned, unfitted Nitsche method for interface problems. *BIT Numer. Math.* **53**(3), 791–820 (2013). <https://doi.org/10.1007/s10543-012-0417-x>
77. Wang, S., Kreiss, G.: Convergence of summation-by-parts finite difference methods for the wave equation. *J. Sci. Comput.* **71**(1), 219–245 (2017). <https://doi.org/10.1007/s10915-016-0297-3>
78. Wang, S., Virta, K., Kreiss, G.: High order finite difference methods for the wave equation with non-conforming grid interfaces. *J. Sci. Comput.* **68**(3), 1002–1028 (2016). <https://doi.org/10.1007/s10915-016-0165-1>
79. Wang, Y., Zhou, H., Yuan, S., Ye, Y.: A fourth order accuracy summation-by-parts finite difference scheme for acoustic reverse time migration in boundary-conforming grids. *J. Appl. Geophys.* **136**, 498–512 (2017). <https://doi.org/10.1016/j.jappgeo.2016.12.002>
80. Woodward, W.H.: On the Application of the Method of Difference Potentials to Linear Elastic Fracture Mechanics. Ph.D. thesis, The University of Manchester (2015)
81. Woodward, W.H., Utyuzhnikov, S., Massin, P.: On the application of the method of difference potentials to linear elastic fracture mechanics. *Int. J. Numer. Methods Eng.* **103**(10), 703–736 (2015). <https://doi.org/10.1002/nme.4903>
82. Xia, K., Zhan, M., Wei, G.W.: MIB method for elliptic equations with multi-material interfaces. *J. Comput. Phys.* **230**(12), 4588–4615 (2011). <https://doi.org/10.1016/j.jcp.2011.02.037>
83. Ye, T., Mittal, R., Udaykumar, H., Shyy, W.: An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *J. Comput. Phys.* **156**(2), 209–240 (1999). <https://doi.org/10.1006/jcph.1999.6356>
84. Yu, S., Wei, G.: Three-dimensional matched interface and boundary (MIB) method for treating geometric singularities. *J. Comput. Phys.* **227**(1), 602–632 (2007). <https://doi.org/10.1016/j.jcp.2007.08.003>
85. Yu, S., Zhou, Y., Wei, G.W.: Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces. *J. Comput. Phys.* **224**(2), 729–756 (2007). <https://doi.org/10.1016/j.jcp.2006.10.030>
86. Zhou, Y., Zhao, S., Feig, M., Wei, G.W.: High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. *J. Comput. Phys.* **213**(1), 1–30 (2006). <https://doi.org/10.1016/j.jcp.2005.07.022>