Title

A high-order immersed boundary discontinuous-Galerkin method for Poisson's equation with discontinuous coefficients and singular sources

Journal

International Journal for Numerical Methods in Engineering, 101(11)

ISSN

0029-5981

Authors

Brandstetter, Gerd
Govindjee, Sanjay

Publication Date

2015-03-16

DOI

10.1002/nme.4835

Peer reviewed

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2015; **101**:847–869
Published online 26 December 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.4835

A high-order immersed boundary discontinuous-Galerkin method for Poisson's equation with discontinuous coefficients and singular sources

Gerd Brandstetter and Sanjay Govindjee*,†

University of California, Berkeley, CA,USA

SUMMARY

We adopt a numerical method to solve Poisson's equation on a fixed grid with embedded boundary conditions, where we put a special focus on the accurate representation of the normal gradient on the boundary. The lack of accuracy in the gradient evaluation on the boundary is a common issue with low-order embedded boundary methods. Whereas a direct evaluation of the gradient is preferable, one typically uses post-processing techniques to improve the quality of the gradient. Here, we adopt a new method based on the discontinuous-Galerkin (DG) finite element method, inspired by the recent work of [A.J. Lew and G.C. Buscaglia. A discontinuous-Galerkin-based immersed boundary method. *International Journal for Numerical Methods in Engineering*, 76:427-454, 2008]. The method has been enhanced in two aspects: firstly, we approximate the boundary shape locally by higher-order geometric primitives. Secondly, we employ higher-order shape functions within intersected elements. These are derived for the various geometric features of the boundary based on analytical solutions of the underlying partial differential equation. The development includes three basic geometric features in two dimensions for the solution of Poisson's equation: a straight boundary, a circular boundary, and a boundary with a discontinuity. We demonstrate the performance of the method via analytical benchmark examples with a smooth circular boundary as well as in the presence of a singularity due to a re-entrant corner. Results are compared to a low-order extended finite element method as well as the DG method of [1]. We report improved accuracy of the gradient on the boundary by one order of magnitude, as well as improved convergence rates in the presence of a singular source. In principle, the method can be extended to three dimensions, more complicated boundary shapes, and other partial differential equations. Copyright © 2014 John Wiley & Sons, Ltd.

Received 21 March 2014; Revised 14 August 2014; Accepted 11 October 2014

KEY WORDS: immersed boundary; discontinuous-Galerkin; extended finite element method; electro-mechanics; MEMS; electrostatic pressure; high-order methods

1. INTRODUCTION

Immersed boundary methods are popular in various fields of computational mechanics. As prominent examples one can name Peskin's immersed boundary method [2, 3], boundary fitting methods [4], fictitious domain methods [5, 6], and the eXtended finite element method (X-FEM) [7, 8]. These methods have been successfully applied to the simulation of fluid-structure interaction, crack propagation, and phase transitions among others. Our primary interest is in the solution of coupled electro-mechanical problems, where we wish to study moving mechanical bodies in electric fields; the bodies may be conductors or dielectrics. Recently, fixed-grid methods [9, 10] have become popular for this task. In comparison to classical Lagrangian methods that adapt their mesh according to the bodies' motion, immersed boundary methods have the clear advantage that no elements can be distorted, as well as no re-meshing is required for possibly large motions of the bodies. The obvious

*Correspondence to: Sanjay Govindjee, University of California, Berkeley, CA, USA.
†E-mail: s_g@berkeley.edu

advantage, however, comes with some difficulties, for example: when Dirichlet type boundary conditions are given, they have to be enforced on non-conforming meshes. Bad element intersections may occur and require special treatment. And lastly, the evaluation of gradients on the boundary often lacks precision. The last point becomes especially problematic in the solution of coupled problems such as in electro-mechanics, where the normal gradient on any interface boundary determines the traction on the body surface, and thus, its accuracy is crucial to obtain physically meaningful results.

In this work, we focus on a numerical method that is designed to provide high quality gradients at the interface. In this context, we first review a low-order X-FEM, where inaccuracies of the gradient at the interface is a common issue. In the literature, there exist several post-processing techniques, which smooth the often largely oscillating gradient field (see e.g. [11]). Here we develop a method that allows one to evaluate the gradient accurately by direct differentiation of the bulk field. Among others, this has the clear advantage that the computational cost will be reduced, and one can avoid difficulties of the smoothing operation that may break down when bodies come close, contact, or singularities are involved. Our method follows, in spirit, the work of [1]. All elements intersected by a boundary will feature a special set of shape-functions that allow a strong imposition of Dirichlet boundary conditions along the immersed boundary. Those non-conforming elements are then put together in the context of a discontinuous-Galerkin (DG) method, where inter-element continuity is enforced in a weak sense in the region near the immersed boundary; regular finite elements are used everywhere else. This method has been adapted, for example, in [12] to problems in elasticity, where it showed very robust behavior in the enforcement of Dirichlet boundary conditions. In our method, we enhance this approach in two aspects: first, we use a higher-order representation of the boundary by approximating the boundary in each element locally via basic geometric primitives such as straight lines, circular curves, or wedges in two dimensions. Second, we use a special higher-order interpolation motivated by the analytical eigensolution of the underlying PDE in the neighborhood of the corresponding special boundary shape. Specifically, we concentrate on Poisson's equation in two-dimensions; however, the method's basic idea can be extended to more complicated boundary shapes and other types of PDEs in two as well as three dimensions. In comparison to the X-FEM as well as the original DG-based immersed boundary method, we achieve much better accuracy of the gradient. Moreover, we demonstrate the capability to incorporate singularities as they arise in the presence of re-entrant corners in a natural way.

The outline of the paper is as follows: in Section 2 and 3, we will state the problem and review a state-of-the-art X-FEM technique. In Section 4, we will layout the principles of the proposed high-order immersed boundary DG method (IB-DG) and our choice of boundary approximation and enrichment functions. Lastly, in Section 5, we discuss the performance of the high-order IB-DG method versus X-FEM and low-order IB-DG via numerical examples. Throughout, we focus on electrostatics and ignore deformation so as to concentrate on the performance of the immersed boundary.

2. GOVERNING EQUATIONS

We assume that we want to solve Poisson's equation in all space, which is divided into domains $\mathcal{R}, \mathcal{V}$, and $\mathcal{W}$ as pictured in Figure 1. $\mathcal{R}$ should be thought of as a body and $\mathcal{V}, \mathcal{W}$ as air. Specifically, we are interested in the solution of the electro-static BVP, with boundary conditions given along $\Gamma = \mathcal{R} \cap \mathcal{V}$. Assuming linear isotropic dielectric properties with permittivity

$$\epsilon(\boldsymbol{x}) = \begin{cases} \epsilon_{\mathcal{R}}, & \boldsymbol{x} \in \mathcal{R}, \\ \epsilon_{\mathcal{V}}, & \boldsymbol{x} \in \mathcal{V}, \\ \epsilon_{\mathcal{W}}, & \boldsymbol{x} \in \mathcal{W}, \end{cases} \tag{1}$$

we look at three typical cases.

Case 1

The body $\mathcal{R}$ is a conductor and the boundary Γ is a conducting surface where we know the potential

DOI: 10.1002/nme

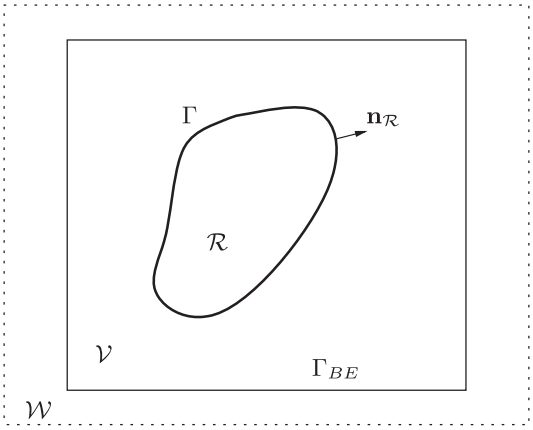


Figure 1. Problem definition and notation.

$\Phi = \bar{\Phi}$. In the absence of any volume charge, the problem reads: find Φ, such that

$$\nabla^2 \Phi = 0 \quad \text{all space}, \tag{2}$$

$$\Phi = \bar{\Phi} \quad \text{on } \Gamma, \tag{3}$$

where $\bar{\Phi}$ is any given Dirichlet boundary data along Γ.

Case 2
The body $\mathcal{R}$ is a conductor and the boundary Γ is a conducting surface where we know the total charge

$$Q = \int_{\Gamma} \sigma_f \, \mathrm{da} \tag{4}$$

on the surface. The free surface charge density $\sigma_f = [\![q]\!]$ is related to the jump of the flux $q = -\epsilon \nabla \Phi \cdot \boldsymbol{n}$ along the surface Γ with normal $\boldsymbol{n}$. In this case, we impose $\Phi = \bar{\Phi} =$ constant as a constraint and treat $\bar{\Phi}$ as an additional (scalar) unknown. The problem reads: find $[\Phi, \bar{\Phi}]$, such that

$$\nabla^2 \Phi = 0 \ \text{ all space}, \tag{5}$$

$$\Phi - \bar{\Phi} = 0 \ \text{ on } \Gamma, \tag{6}$$

$$\int_{\Gamma} [\![q]\!] \, \mathrm{da} = Q. \tag{7}$$

Case 3
The body $\mathcal{R}$ is a dielectric and the boundary Γ is a dielectric-dielectric interface. In this case the problem reads: find Φ, such that

$$\nabla^2 \Phi = 0 \ \text{ all space}, \tag{8}$$

$$[\![q]\!] = 0 \ \text{ on } \Gamma. \tag{9}$$

Again, $[\![q]\!]$ denotes the jump in the normal flux, and (9) accounts for the fact that no free surface charge is present at a dielectric-dielectric interface by assumption.

DOI: 10.1002/nme

3. EXTENDED FINITE ELEMENT METHOD (X-FEM)

For *Cases 1–3* we derive the variational form: find $\Phi \in \mathcal{P}_s$, such that

$$\int_{\mathcal{R}} \epsilon_{\mathcal{R}} \nabla \delta\Phi \cdot \nabla \Phi \, \mathrm{dv} + \int_{\mathcal{V}} \epsilon_{\mathcal{V}} \nabla \delta\Phi \cdot \nabla \Phi \, \mathrm{dv} = -\int_{\Gamma_{BE}} \delta\Phi q_{\mathcal{V}} \, \mathrm{da} \tag{10}$$

for all $\delta\Phi \in \mathcal{P}_v$ along with the requirement $\Phi = \bar{\Phi}$ on Γ for *Cases 1* and *2*. Here, the spaces $\mathcal{P}_s$ and $\mathcal{P}_v$ are suitable subspaces of H^1.

In order to solve (10) using X-FEM, one typically discretizes each domain as pictured in Figure 2(a). In this depiction, we assume the boundary Γ is discretized by linear elements, and $\mathcal{R}, \mathcal{V}$ are discretized by quadrilateral elements covering each domain of interest. The effect of $\mathcal{W}$ is modeled as a far field boundary condition along Γ_{BE} via the boundary element method (see App. A). All elements that are intersected by Γ will overlap and feature the interpolation

$$\Phi^h = \Phi^h_{\mathcal{R}} + \Phi^h_{\mathcal{V}} = \sum_i H_{\mathcal{R}} N_i \Phi_{\mathcal{R}i} + \sum_i H_{\mathcal{V}} N_i \Phi_{\mathcal{V}i}, \tag{11}$$

where N_i are the classical, finite element shape functions. The characteristic function $H_{\mathcal{R},\mathcal{V}}$ equals one in the corresponding domain and zero elsewhere. A standard bi-linear interpolation has four degrees of freedom for each element; with one intersection, we obtain eight degrees of freedom defining the eXtended or enhanced element. Note that the discontinuous shape functions will allow us to capture kinks in the potential field as pictured in Figure 2(b). Using this interpolation requires an additional constraint equation to enforce continuity along Γ.

The approximate problem then reads: find $\Phi^h_{\mathcal{R}}, \Phi^h_{\mathcal{V}} \in \mathcal{P}^h_s$, such that

$$\int_{\mathcal{R}} \epsilon_{\mathcal{R}} \nabla \delta\Phi^h_{\mathcal{R}} \cdot \nabla \Phi^h_{\mathcal{R}} \, \mathrm{dv} + \int_{\mathcal{V}} \epsilon_{\mathcal{V}} \nabla \delta\Phi^h_{\mathcal{V}} \cdot \nabla \Phi^h_{\mathcal{V}} \, \mathrm{dv} = -\int_{\Gamma_{BE}} q_{\mathcal{V}} \delta\Phi^h_{\mathcal{V}} \, \mathrm{da} \tag{12}$$

for all $\delta\Phi^h_{\mathcal{R}}, \delta\Phi^h_{\mathcal{V}} \in \mathcal{P}^h_v$ along with the requirement $\Phi^h_{\mathcal{V}} = \Phi^h_{\mathcal{R}} = \bar{\Phi}$ on Γ^h (for *Cases 1* and *2*), and $\Phi^h_{\mathcal{V}} = \Phi^h_{\mathcal{R}}$ on Γ^h (for *Case 3*). Note that the last three requirements can only be enforced in a weak sense. For simplicity, we discuss here only the constraint $\Phi^h_{\mathcal{V}} = \bar{\Phi}$ on Γ^h, the others follow in a similar fashion.

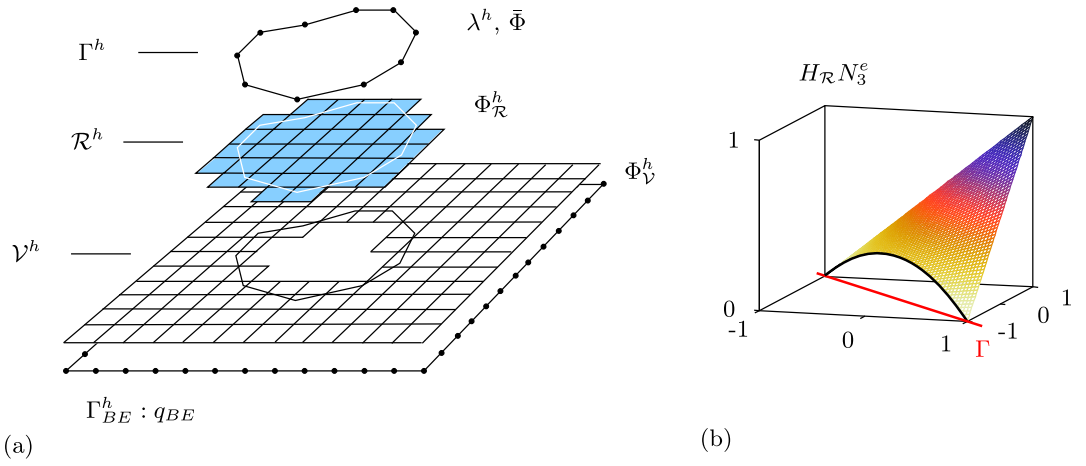


Figure 2. Schematic of the eXtended finite element method: (a) domain discretization and (b) discontinuous shape function.

DOI: 10.1002/nme

In order to enforce $\Phi_{\mathcal{V}} = \bar{\Phi}$ on Γ in a weak sense, a typical choice is the Lagrange multiplier method. Let us introduce $\lambda \in \mathcal{L}$, where $\mathcal{L} = H^{-1/2}$. One then requires stationarity of the functional

$$\Pi_{LM}(\Phi_{\mathcal{V}}, \lambda) = \int_{\Gamma} \lambda \left(\Phi_{\mathcal{V}} - \bar{\Phi}\right) \mathrm{da}. \tag{13}$$

Upon variation, we obtain: find $(\Phi_{\mathcal{V}}, \lambda) \in \mathcal{P}_s \times \mathcal{L}$, such that (10) holds, and such that

$$\int_{\Gamma} \delta\lambda \Phi_{\mathcal{V}} \, \mathrm{da} + \int_{\Gamma} \delta\Phi_{\mathcal{V}} \, \lambda \mathrm{da} = \int_{\Gamma} \delta\lambda \, \bar{\Phi} \mathrm{da} \tag{14}$$

for all $(\delta\Phi_{\mathcal{V}}, \delta\lambda) \in \mathcal{P}_v \times \mathcal{L}$. By choosing a discretization $\lambda^h \in \mathcal{L}^h$, we arrive at the discrete form: find $\left(\Phi^h_{\mathcal{V}}, \lambda^h\right) \in \mathcal{P}^h_s \times \mathcal{L}^h$, such that (12) holds, and such that

$$\int_{\Gamma} \delta\lambda^h \, \Phi^h_{\mathcal{V}} \, \mathrm{da} + \int_{\Gamma} \delta\Phi^h_{\mathcal{V}} \, \lambda^h \, \mathrm{da} = \int_{\Gamma} \delta\lambda^h \, \bar{\Phi} \mathrm{da} \tag{15}$$

for all $\left(\delta\Phi^h_{\mathcal{V}}, \delta\lambda^h\right) \in \mathcal{P}^h_v \times \mathcal{L}^h$. One can proceed in an analogous manner for constraints $\Phi^h_{\mathcal{R}} = \bar{\Phi}$ and $\Phi^h_{\mathcal{V}} = \Phi^h_{\mathcal{R}}$ on Γ^h.

We remark that by a standard localization argument, from (12) to (15) one can show that the Lagrange multiplier equals the normal flux on the boundary:

$$\lambda^h = -\epsilon_{\mathcal{V}} \nabla \Phi^h_{\mathcal{V}} \cdot \boldsymbol{n}_{\mathcal{V}}. \tag{16}$$

As will be assessed in Section 5, this presents an interesting alternative to the direct evaluation of the normal gradient on the boundary.

Note that here, $\bar{\Phi}$ is any given potential, and in the case of a conducting body, $\bar{\Phi}$ will be constant on the body. For a typical electro-static problem, however, it might occur that the voltage (=potential) is not controlled, but rather the total electrical charge Q on a conductor is specified, and one must calculate the corresponding potential as a (scalar) unknown– the so-called floating potential problem. In this case we modify (15) to: find $\left(\Phi^h_{\mathcal{V}}, \lambda^h, \bar{\Phi}\right) \in \mathcal{P}^h_s \times \mathcal{L}^h \times \mathbb{R}$, such that (12) holds, and such that

$$\bar{\Pi}_{LM}\left(\Phi^h_{\mathcal{V}}, \lambda^h, \bar{\Phi}\right) = \int_{\Gamma} \lambda^h \left(\Phi^h_{\mathcal{V}} - \bar{\Phi}\right) \mathrm{da} + Q\bar{\Phi} \tag{17}$$

is rendered stationary. The Lagrange multiplier λ^h can still be interpreted as the normal flux on the boundary, which is essentially the surface charge distribution (Section 2). Upon variation of (17), we see that

$$Q = \int_{\Gamma} \lambda^h \, \mathrm{da}, \tag{18}$$

which is consistent with the constraint Equation (7) for the given charge load.

Note that the Lagrange multiplier space $\mathcal{L}^h$ is not arbitrary but has to satisfy the *inf-sup* condition to ensure stability [13]. Moreover, the use of Lagrange multipliers delivers a non-positive system and additional degrees of freedom are introduced. There are many studies that deal with these issues and propose solutions on how to choose $\mathcal{L}^h$ [14–16]; further alternative formulations such as Nitsche's method or stabilized Lagrange multipliers, respectively, have also been advocated [17–25]. For our purpose of benchmark testing, the classic Lagrange multiplier approach works nicely as we can control $\mathcal{L}^h$ *a-priori*.

We want to draw special attention to three short comings of the presented X-FEM method and related technologies, when utilizing a low-order interpolation such as the bi-linear interpolation. First, as will be demonstrated in Section 5, in the presence of a corner or any more complicated

DOI: 10.1002/nme

geometry, one will not be able to interpolate the field exactly along the immersed boundary. Second, the bi-linear interpolation obviously does not account for any possible singularity in the gradient, as arises, for example, at a re-entrant corner. And third, the evaluation of the gradient on the boundary will in general be very inaccurate and highly oscillatory even for smooth boundaries, depending on where the element is cut. These issues can be addressed by mesh-refinement, but this is certainly not in the spirit of embedded boundary methods that were developed to precisely avoid this. A higher-order X-FEM technique (see e.g. [26]) may show some improvement related to these concerns, but we found the approach based on the DG FEM as presented by [1] more natural to extend for our specific demands.

4. HIGH-ORDER IMMERSED BOUNDARY DISCONTINUOUS-GALERKIN METHOD

In this work, we propose a new immersed boundary method based on the DG FEM. The DG approach has been used by [1] and [27] recently in a similar context. However, to our knowledge, no studies have demonstrated yet the use of higher-order approximations to the boundary shape or interpolation space.

The basic idea is pictured in Figure 3: all elements that are *not* intersected by the Γ-boundary utilize standard conforming finite elements. In our examples, we will use a bi-linear interpolation. All elements that are intersected by the Γ-boundary utilize a special interpolation that is element-wise dependant on the shape and location of Γ. All intersected elements are by default non-conforming, and continuity across element boundaries and to the standard FE domain is enforced in the DG

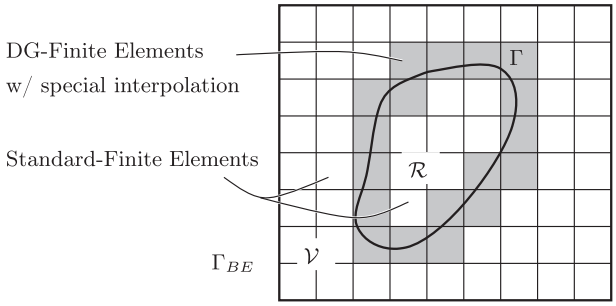


Figure 3. Schematic of the high-order immersed boundary discontinuous-Galerkin method: Γ-boundary, intersected DG-finite elements, standard finite elements, and Γ_{BE}.

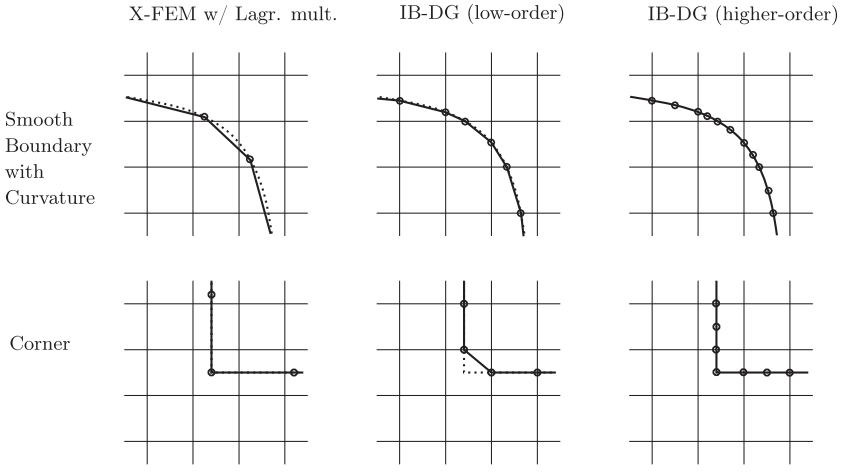


Figure 4. Schematic of various Γ-approximations: (left) eXtended finite element method (X-FEM) with Lagrange multiplier; (middle) low-order immersed boundary discontinuous-Galerkin (IB-DG); (right) higher-order IB-DG.

DOI: 10.1002/nme

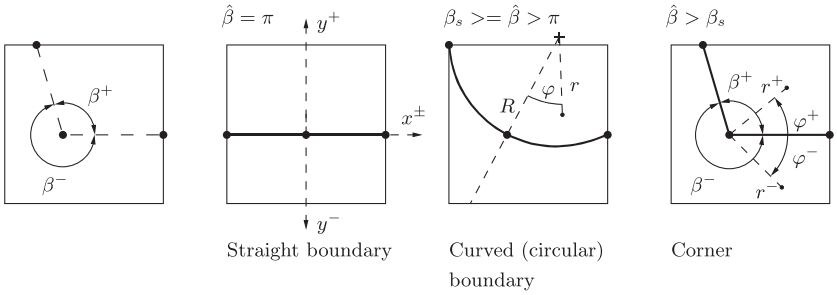


Figure 5. Schematic of basic geometric primitives used for higher-order Γ-approximation.

context. At this point, we want to review various Γ-approximations as pictured in Figure 4. Note that the approximation of the boundary is in general independent of the field interpolation. Due to the convenience in the integration, one typically uses piecewise linear patches in standard (low-order) X-FEM or low-order IB-DG methods. As can be readily observed, this approach will lead to inaccuracies in the Γ-approximation, which guide us to a higher-order approximation, built of nonlinear geometric primitives. We locally approach the Γ-boundary as sketched in Figure 5: for each intersected element we calculate three *control-points*, each lying on Γ. Depending on the angle $\hat{\beta} = \max\{\beta^+, \beta^-\}$, we propose an automatic heuristic switch based on a user-defined parameter β_s.

- If $\hat{\beta} = \pi$, approximate Γ by a *straight line* through all three control-points.
- If $\beta_s \geqslant \hat{\beta} > \pi$, approximate Γ by a *circular curve* through all three control-points.
- If $\hat{\beta} > \beta_s$, approximate Γ by a *wedge* with vertex at the mid-control-point.

In our later examples, we set $\beta_s = 1.3\,\pi$, but this can be adjusted by the user's need. Instead of an automatic switch, this can also be performed by a user decision– for example, by flagging certain nodes along the boundary as singular corners, and moreover, one may utilize more complicated shapes from a user defined library to approximate the boundary at the required accuracy.

As mentioned before, by default, we use a low-order interpolation for all elements that are *not* intersected by Γ. For all elements intersected by Γ, we switch to a higher-order approximation that follows the boundary shape locally. For the three basic shapes we developed so far, we propose the following interpolations.

4.1. Straight boundary

In the case of a straight boundary, for each side of the element one can use a local Cartesian coordinate system $\{x, y\}$ (Figure 5), and approximate the solution by polynomial spaces. In accordance with [1], this can be performed by a linear (low-order) space

$$\Phi^h \in \text{span}\{1, x, y^\pm\}, \tag{19}$$

featuring four degrees of freedom per element. Note that this approximation will be used in the low-order IB-DG implementation for comparison in the next section. In our higher-order formulation, we will use a quadratic space

$$\Phi^h \in \text{span}\{1, x, x^2, y^\pm, (y^2)^\pm, xy^\pm\}, \tag{20}$$

resulting in nine degrees of freedom per element. This enables a more accurate gradient interpolation and, moreover, avoids locking of the solution when non-constant gradients occur in an element.

In this notation, we note that all modes labeled $\{(.)^\pm\} = \{(.)^+, (.)^-\}$ have to be counted twice as they are independently used to interpolate the field in each domain. Note that the interpolation space is designed to follow the boundary shape, which enables one to specify Dirichlet type boundary

DOI: 10.1002/nme

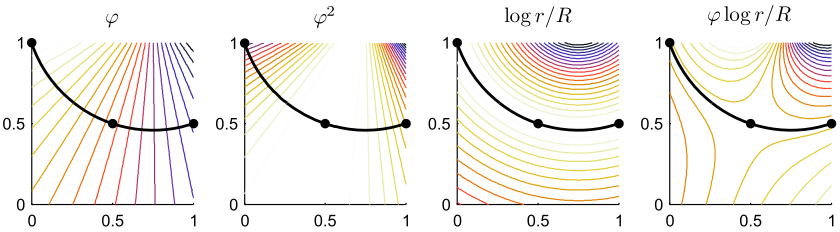


Figure 6. Shape functions for circular boundary.

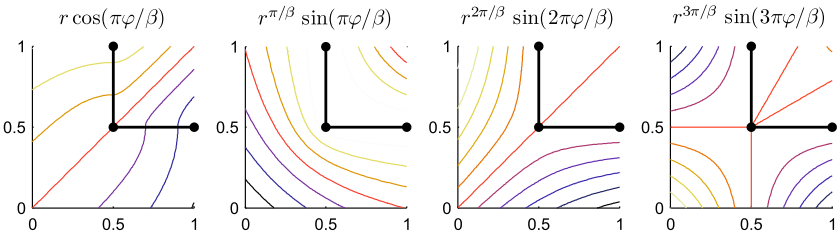


Figure 7. Shape functions for corner element.

conditions in a strong sense. This is one major difference to the X-FEM method, where Dirichlet boundary conditions can only be enforced in a weak sense.

4.2. *Circular boundary*

In the case of a circular boundary approximation, we introduce the polar coordinate system $\{r, \varphi\}$ as shown in Figure 5 and propose the shape functions

$$\Phi^h \in \text{span}\{1, \varphi, \varphi^2, \log(r/R)^{\pm}, \varphi \log(r/R)^{\pm}\}, \tag{21}$$

as pictured in Figure 6 (viz. seven degrees of freedom). These functions are motivated by the analytical eigensolution of Poisson's equation near a circular boundary in two dimensions (see e.g. [28, §9.4]).

4.3. *Corner element*

Lastly, we propose a corner interpolation using a polar coordinate system $\{r, \varphi\}$ as pictured in Figure 5. Different to the circular boundary case, the coordinate center is now at the singular corner location. We assume

$$\Phi^h \in \text{span}\left\{1, r\cos(\pi\varphi/\beta), r^2\cos^2(\pi\varphi/\beta), r^{m\pi/\beta}\sin(m\pi\varphi/\beta)^{\pm}\right\}_{m=1}^{N}, \tag{22}$$

again motivated by near field solutions to Poisson's equation [28, §9.4]; see Figure 7. Note that $m = 1$ represents the singularity in the gradient due to a re-entrant corner; modes $m > 1$ represent higher-order series expansions of the exact solution. We found the choice $N = 3$ (nine degrees of freedom) sufficiently accurate for our numerical experiments in Section 5. Higher choices of N will give more accurate results but potentially lead to instabilities.

4.4. *Inter-element continuity*

Because all intersected elements have locally defined solution parameters, one has to enforce continuity along the element boundaries. For simplicity, we employ a DG method with internal penalties (IP-DG) [29–31]. Alternatively one could utilize other DG methods such as Bassi-Rebay [32], the local DG [33], or the compact DG method [34]. These approaches improve upon IP-DG but for our purpose of developing a proper boundary representation they are unneeded and we opt for the simplicity afforded by IP-DG.

DOI: 10.1002/nme

The overall problem then reads: find $\Phi^h \in \mathcal{P}_s^h$, such that (12) holds and

$$\Pi^{DG}(\Phi^h) = \sum_e \left\{ \int_{\Gamma_e} \langle q^h \rangle [\![\Phi^h]\!] \mathrm{da} - \frac{\alpha}{h_e} \int_{\Gamma_e} [\![\Phi^h]\!]^2 \, \mathrm{da} \right\} \rightarrow \text{ stat.} \tag{23}$$

Here

$$[\![\Phi^h]\!] = \Phi^{h+} - \Phi^{h-}, \tag{24}$$

$$\langle q^h \rangle = \frac{1}{2}(q^{h+} + q^{h-}) \tag{25}$$

denote the jump and average of field or flux, respectively, across each element boundary Γ_e. The sum goes over all boundaries of intersected elements. The stability parameter α in this form is scaled by the local area-measure h_e of the element boundary. After the typical variation, together with (12) we arrive at a linear system, where with ease we can strongly enforce Dirichlet type boundary conditions along Γ. Note that α in this work is a user-defined parameter that must be chosen high enough in order to satisfy coercivity of the weak form, and low enough to retain accuracy. Recent work by [35, 36] or [37] establish methods to estimate α based on the solution of a local eigen-value problem. For our purposes of benchmark testing we manually optimize α for the given examples.

4.5. Charge loading

For the charge loading case, let us denote $\Phi^h = \sum_{j=1}^m P_j \Phi_j$ with $P_j \in \mathcal{P}_s^h$ and the expansion coefficients Φ_j. Using a Galerkin discretization, one can write (12) and (23) in the algebraic form: find $[\Phi_j]$ such that

$$\sum_j K_{ij} \Phi_j = q_i \quad \forall i = 1, \ldots, m, \tag{26}$$

where K_{ij} are the coefficients of the electrical stiffness, and q_i the equivalent fluxes for each degree of freedom. Let us now denote the set

$$\mathcal{J} = \{ j \mid P_j(\boldsymbol{x}) \neq 0, \forall \boldsymbol{x} \in \Gamma \}, \tag{27}$$

which are the degrees of freedom in the intersected elements that are used to interpolate $\Phi^h = \bar{\Phi}$ along Γ. We now split $\mathcal{J}$ into $\mathcal{J} = \mathcal{J}_0 \cup \mathcal{J}_n$, with $\mathcal{J}_0 \cap \mathcal{J}_n = \emptyset$, where $\mathcal{J}_0$ is the set of all constant modes and $\mathcal{J}_n$ is the set of all higher modes. For charge loading on a conducting surface we then require

$$\Phi_j = \bar{\Phi}, \forall j \in \mathcal{J}_0, \quad \Phi_j = 0, \forall j \in \mathcal{J}_n, \quad \sum_{j \in \mathcal{J}_0} q_j = Q, \tag{28}$$

which are the equivalent forms to (17) and (18). From (26) we then derive: find $[\Phi_j, \bar{\Phi}]$, $j \notin \mathcal{J}$, such that

$$\sum_{j \notin \mathcal{J}} K_{ij} \Phi_j + \sum_{j \in \mathcal{J}_0} K_{ij} \bar{\Phi} = q_i, \quad \forall i \notin \mathcal{J}, \tag{29}$$

$$\sum_{i \in \mathcal{J}_0} \sum_{j \notin \mathcal{J}} K_{ij} \Phi_j + \sum_{i \in \mathcal{J}_0} \sum_{j \in \mathcal{J}_0} K_{ij} \bar{\Phi} = Q. \tag{30}$$

for any given equivalent nodal fluxes q_i and total charge Q.

DOI: 10.1002/nme

4.6. Integration

Before we proceed to the numerical examples, we point out some further details of the implementation. For elements intersected by a smooth boundary, we employ standard Gauss integration procedures via tessellation [7]. In elements featuring a sharp corner, we have to integrate a singular function of the form

$$\int_{\Delta_I} \epsilon(\pi/\beta)^2 r^{-\gamma} \mathrm{da}, \tag{31}$$

where $\gamma = 2 - 2\pi/\beta$ is the order of the singularity. As pictured in Figure 8 (right), we divide elements by a Delaunay-triangularization into triangles Δ_I, such that the singularity is at one vertex. Following [38], we then use a generalized Duffy-trick to integrate each triangle with a proper Gauss-rule that respects the order of the singularity depending on the angle $\beta^{\pm}$.

4.7. Singular corner enrichment radius

In accordance with [39] and [40], we observe that optimal convergence during mesh-refinement for the singular enrichment can only be achieved by enhancing all elements surrounding the singular point within a certain radius r_E (Figure 9). For each such element, we refer to the same source of singularity, from which we measure the local coordinates $\{r, \varphi\}$ and from which we take the angle $\beta^{\pm}$.

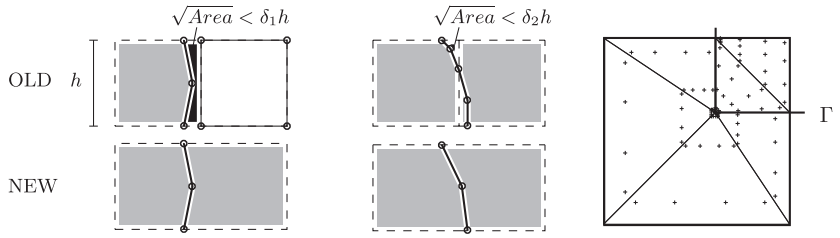


Figure 8. Immersed boundary discontinuous-Galerkin (IB-DG): (left) element extensions; (right) Gauss integration of singular functions via generalized Duffy-trick.

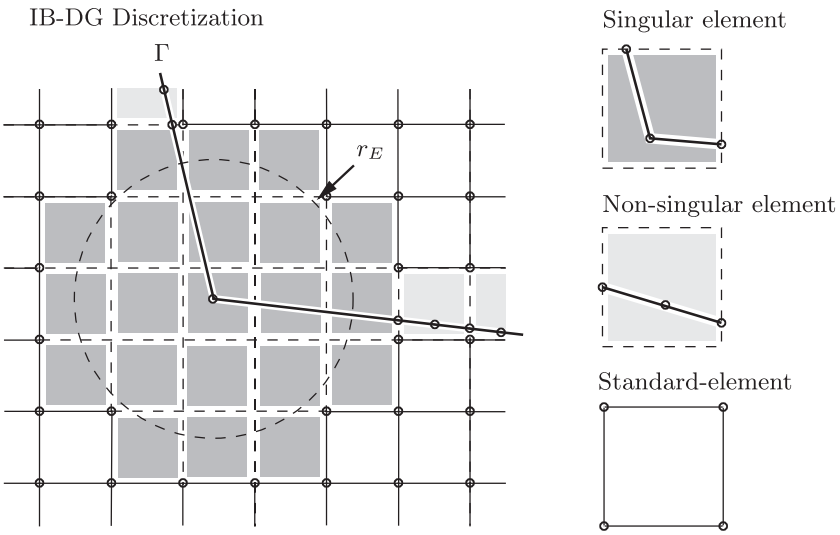


Figure 9. Singular element enhancement within enrichment radius r_E.

DOI: 10.1002/nme

4.8. Element extensions

Because one typically cannot control the boundary location, for example if one has moving bodies, bad element intersections may lead to ill-conditioning. To alleviate this problem, we follow a procedure similar to what was proposed in [41]. In two dimensions, our strategy depends on how the elements are intersected. When two opposite sides are intersected (Figure 8, left) and $\sqrt{Area} < \delta_1 h$, we extend the element to the next neighbor. When two adjacent sides are intersected (Figure 8, middle) and $\sqrt{Area} < \delta_2 h$, we merge two intersected elements that share a common edge. In rare cases, it may occur that such merging will lead to a successive combining of elements into a very large element. For a structured mesh as pictured here, this can be avoided by consistently merging elements in only one coordinate direction. With unstructured meshes, one would have to consider a strategy that avoids such combinations.

4.9. Element identification

In order to identify the various element-types, a Level-set function is calculated for each solution iteration. In our examples, we compute the signed-distance to the boundary Γ at each nodal location of the computational domain $\Omega^h = \mathcal{R}^h \cup \mathcal{V}^h$. This enables an identification of all intersected elements, for which we identify the element control-points as pictured in Figure 5. Subsequently we check for singular elements and bad element intersections depending on the parameters r_E, δ_1, δ_2. Note again that all identified elements, that track the interface, feature a higher-order locally defined interpolation, whereas the remaining elements utilize a standard bi-linear interpolation as pictured in Figure 9. This is a major advantage of the IB-DG method versus X-FEM. Once the basic framework is implemented, one can easily define new elements and combine various element-types of different orders to obtain an optimal and efficient interpolation space for the problem at hand.

5. NUMERICAL EXAMPLES

We now look at several examples where a straight forward analytical solution is accessible. In particular, we will validate the accuracy and convergence during $h-$refinement. To this end we introduce the relative L2-error norms

$$||\Phi - \Phi^h||_\Omega / ||\Phi||_\Omega = \sqrt{\int_\Omega \left(\Phi^h - \Phi\right)^2 \mathrm{d}\Omega} \Big/ \sqrt{\int_\Omega \Phi^2 \mathrm{d}\Omega}, \tag{32}$$

for the bulk field in the computational domain Ω and

$$||\nabla_n \Phi - \nabla_n \Phi^h||_\Gamma / ||\nabla_n \Phi||_\Gamma = \sqrt{\int_\Gamma \left(\nabla_n \Phi^h - \nabla_n \Phi\right)^2 \mathrm{d}\Gamma} \Big/ \sqrt{\int_\Gamma \nabla_n \Phi^2 \mathrm{d}\Gamma}, \tag{33}$$

for the normal gradient along Γ.

5.1. Two cylinders

In this example, we assume two cylinders, which are separated by a distance c, and one is kept at a fixed potential $\Phi = \Phi_0$, while the other is kept at $\Phi = -\Phi_0$ (Figure 10). The analytical solution is given by (see e.g. [42, p.15])

$$\Phi(\boldsymbol{x}) = \Phi_0 \log \frac{r_2(\boldsymbol{x})}{r_1(\boldsymbol{x})} \Big/ \log \frac{a}{d}, \tag{34}$$

where the cylinder radius a, r_1, r_2 are pictured in Figure 10 and $d = c/2 - \sqrt{0.25c^2 - a^2}$.

DOI: 10.1002/nme

For the numerical example we consider $\epsilon_{\mathcal{V}} = \epsilon_{\mathcal{W}} = 1, \Phi_0 = 300, a = 0.1$ and $c = 0.5$. The background mesh covers the domain $\Omega = [0, 1]^2$ and has been refined from 25×25 to 200×200 elements; that is, the element size $h = 1/25, \ldots, 1/200$. For ease of implementation, we discretize the cylinder surface (Γ^h) by N piecewise linear surface patches. Because we adopt the Lagrange multiplier space according to the surface discretization for X-FEM via Lagrange multipliers, N cannot be chosen arbitrarily in this case because the inf-sup condition must be satisfied. We found the best possible results by using 10–80 linear elements for each cylinder (Figure 10), which corresponds to a ratio $l/h \approx 1.6$. In the case of the IB-DG method, there is no constraint on the surface discretization, and we used $N = 500$ to obtain an accurate representation of Γ. Following a sensitivity study as discussed later, for the low-order IB-DG we choose $\alpha = 100$. For the higher-order IB-DG we choose $\alpha = 150$ and in addition use element extensions with $\delta_1 = 0.6, \delta_2 = 1$. No radius r_E needs to be specified in this example because no singularities are present.

Looking at typical results in Figure 11(a-c), we readily observe two advantages of the higher-order IB-DG method: first, the potential field follows the surface discretization much more accurately than in the case of a low-order IB-DG method or X-FEM. Second, the constraint $\Phi = \bar{\Phi}$ is enforced

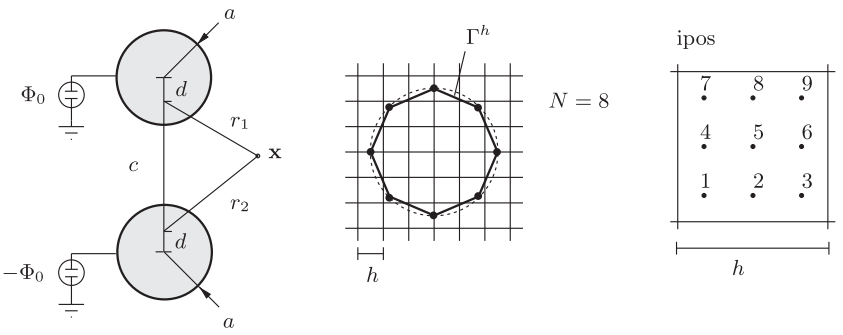
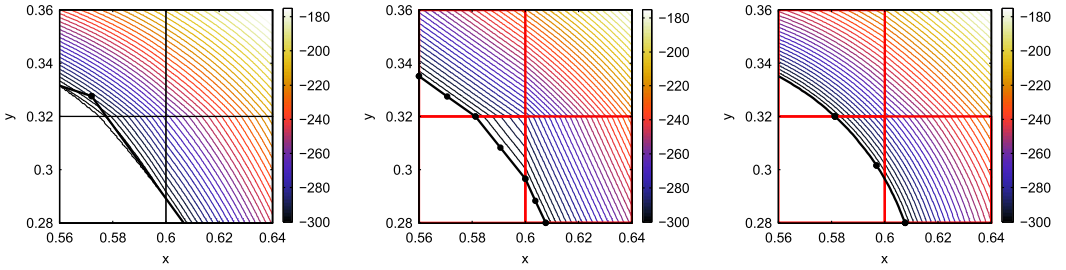
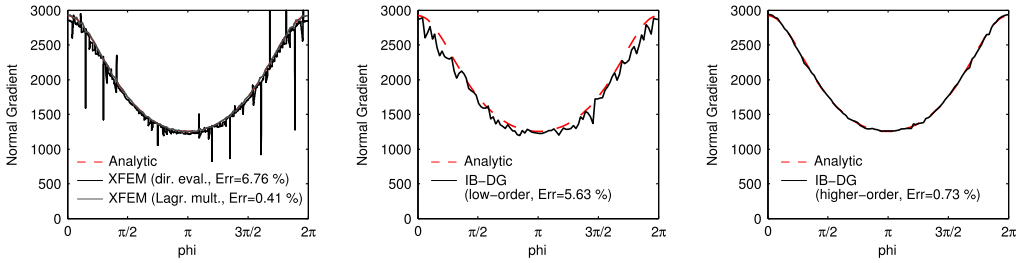


Figure 10. Two-cylinders example: schematic, discretization and various cylinder center positions (ipos).



(a) Potential contour: XFEM (h=1/25). (b) Potential contour: IB-DG (low-order, h=1/25). (c) Potential contour: IB-DG (higher-order, h=1/25).



(d) Normal gradient: XFEM (h=1/200). (e) Normal gradient: IB-DG (low-order, h=1/200). (f) Normal gradient: IB-DG (higher-order, h=1/200).

Figure 11. Two-cylinders example: (top) detailed potential contour, Γ-boundary and background mesh with control-points; (bottom) normal gradient along cylinder surface.

DOI: 10.1002/nme

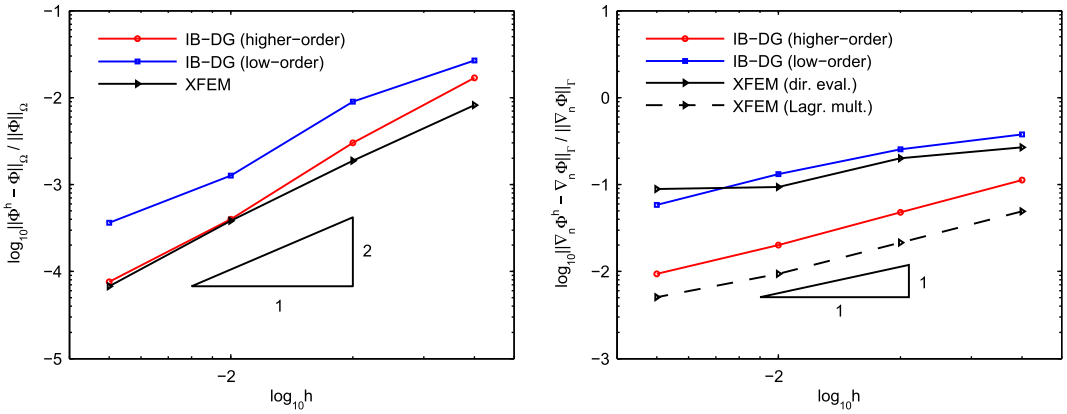


Figure 12. Two-cylinders example, maximum L2-error convergence: (left) bulk field; (right) normal gradient.

exactly on the boundary, which cannot be guaranteed by X-FEM. Looking at Figure 11(d-f), the advantage of the higher-order enhancement becomes even more obvious. Whereas X-FEM and low-order IB-DG give very poor quality of the gradient along Γ, the error for higher-order IB-DG is noticeably better. Inaccuracies in the gradient as arising in Figure 11(d) are a well known issue to the X-FEM community [11]. Methods exist to reconstruct more accurate gradients by post-processing steps, but this is not necessary for the proposed higher-order IB-DG method. Note that the Lagrange multiplier in this example does give a very accurate representation of the normal flux on the boundary, and only small oscillations occur. However, this is a best case scenario as we optimized $\mathcal{L}^h$, and any other choice will easily give much worse results with possibly large oscillations. Moreover, as will be observed in the next example, the standard Lagrange multiplier approach will fail whenever singularities in the gradient field are involved. The convergence during h-refinement of the field Φ and the normal gradient field is shown in Figure 12. All methods show second and first order convergence of the field and the normal gradient error, respectively. The gradient approximation of the higher-order IB-DG method and the Lagrange-multiplier are about one order of magnitude more accurate than the low-order IB-DG method and X-FEM, but the rates are the same.

5.2. *Sensitivity (interface location)*

In a general application of the method, the elements can be intersected by the boundary in any possible way. Thus, we test over a certain range of configurations and report in our convergence plots worst case scenarios as an upper bound on the error. To give a full picture, Figure 13(left) shows the error as we vary the cylinder center as pictured in Figure 10. We observe very little variation of the error, which has also been confirmed in the examples that follow.

5.3. *Sensitivity* (α)

As indicated previously, one needs to select the stabilization parameter α. In order to pick α for the IB-DG methods, we plot the field- and gradient-error in Figure 13(center and right). The optimum α depends on the mesh refinement h. In this and the upcoming examples, we choose a minimum α such that optimum convergence during h-refinement was achieved over the range of h-values examined. For any given discretization h, however, other choices of α may give more accurate results in terms of the error constant. This manual procedure ensures that we observe full rates of convergence. However, in a general setting one should employ an automatic parameter selection scheme. We also note that the sensitivity with respect to α is essentially independent of the degree of intersection, ipos.

DOI: 10.1002/nme

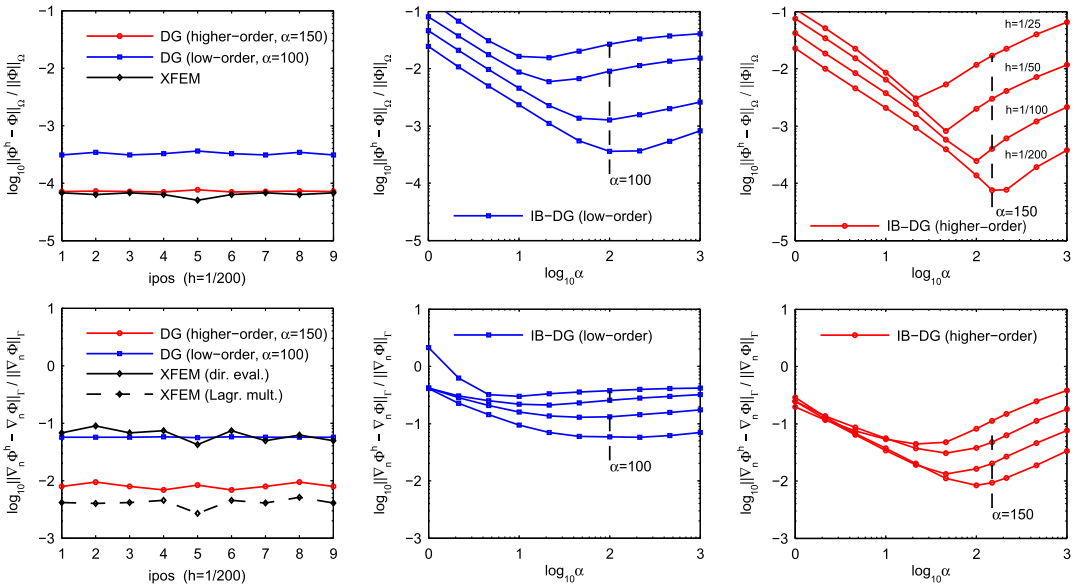


Figure 13. Two-cylinders example: sensitivity of bulk- and gradient-error (left) with respect to interface location (ipos); (center and right) with respect to stability parameter (α) for various h-values as indicated in upper right figure.

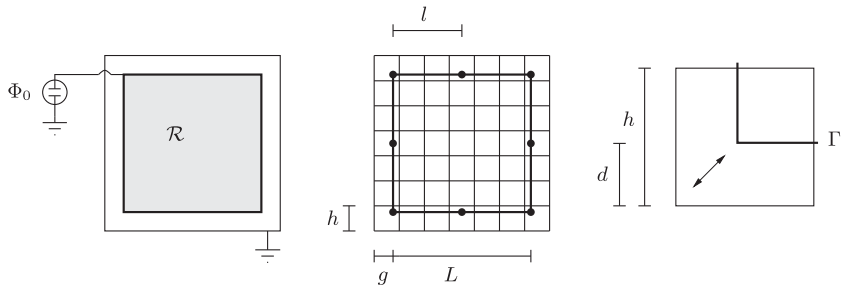


Figure 14. Rectangular-corner example: schematic, discretization and varying interface location.

5.4. *Rectangular corner*

For the second example, we assume a rectangular body within a box, where the potential at the boundary Γ is held at $\Phi_0 = 300$ and at the border of the box is set to zero (Figure 14). We denote the length of the rectangular body as L and the gap between Γ and the outer box as g. When the gap-to-length ratio $g/L << 1$, we set the origin of a Cartesian coordinate system (x, y) at the lower left corner of the outer box, and one can find the analytical solution near this corner as pointed out in [43, p.21] via a conformal mapping $f : \tilde{z} \to z$, with $z, \tilde{z} \in \mathbb{C}$. Here $z = x + iy$ represents the coordinate location in the physical space, whereas $\tilde{z} = \tilde{r}\cos(\tilde{\varphi}) + i\tilde{r}\sin(\tilde{\varphi})$ follows rays of corresponding potentials $\Phi = \Phi_0\tilde{\varphi}/\pi$ for any fixed angle $\tilde{\varphi} \in [0, \pi]$. The mapping f is given by

$$f(\tilde{z}) = \frac{2g}{\pi}\left[\arctan\sqrt{\frac{\tilde{z}-1}{\tilde{z}+1}} + \frac{1}{2}\ln\frac{1+\sqrt{\frac{\tilde{z}-1}{\tilde{z}+1}}}{1-\sqrt{\frac{\tilde{z}-1}{\tilde{z}+1}}}\right], \tag{35}$$

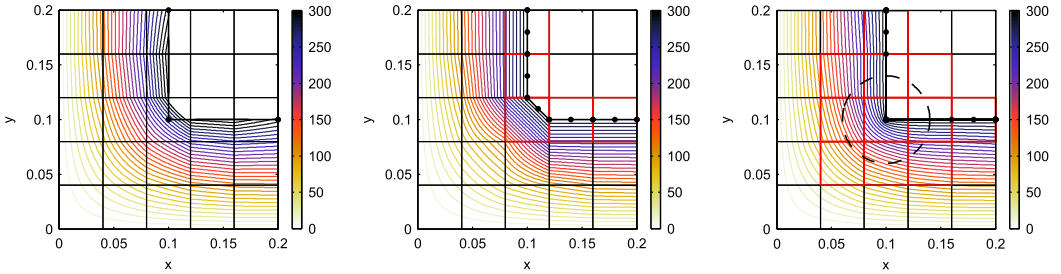
DOI: 10.1002/nme

and we find the solution at $z = (x, y)$ formally by taking the inverse $\tilde{z} = f^{-1}(z)$. The normal gradient then is

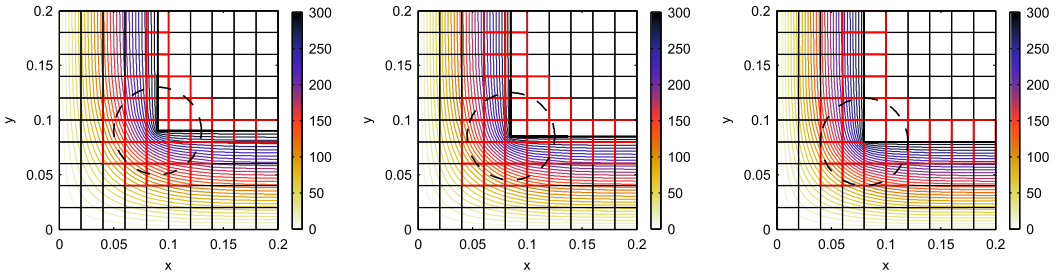
$$\nabla_n \Phi = \frac{\Phi_0}{g} \sqrt{\frac{\tilde{z} - 1}{\tilde{z} + 1}}. \tag{36}$$

For the numerical example we consider $\epsilon_{\mathcal{V}} = 1, \Phi_0 = 300$, and vary L to change the interface location. The background mesh covers the domain $[0, 1]^2$, and has been refined from 25×25 to 200×200 elements; that is, the element size $h = 1/25, \ldots, 1/200$. Again, for X-FEM via Lagrange multipliers, the corner surface discretization (Γ^h) is not arbitrary and has been optimized to 8–64 linear elements per side ($l/h \approx 2.5$). For the low-order DG we choose $\alpha = 20$, whereas for the higher-order DG we choose $\alpha = 1000, \delta_1 = 0.6, \delta_2 = 1$. Due to the presence of a singularity, in the higher-order IB-DG we use a geometric enrichment around the singularity with $r_E = 0.04$. We chose r_E to sufficiently cover the effective radius of the singularity as observed in the numerical examples.

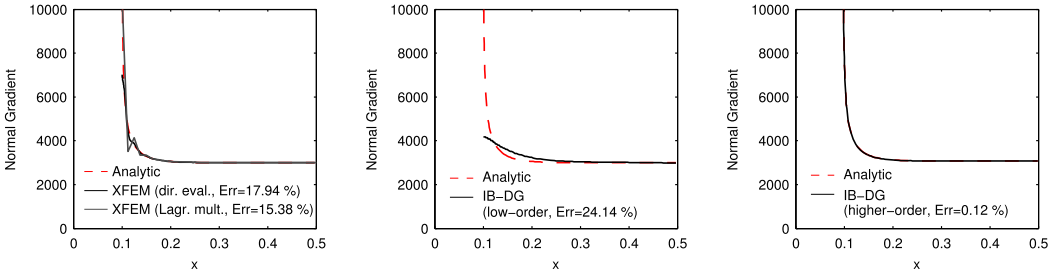
Looking at typical results in Figure 15, we make similar observations as in the previous example but even more distinct. The low-order X-FEM and IB-DG clearly fail to interpolate the poten-



(a) Potential contour: XFEM (h=1/25). (b) Potential contour: IB-DG (low-order, h=1/25). (c) Potential contour: IB-DG (higher-order, h=1/25).



(d) Potential contour: IB-DG (higher-order, h=1/50, d/h=0.5). (e) Potential contour: IB-DG (higher-order, h=1/50, d/h=0.25). (f) Potential contour: IB-DG (higher-order, h=1/50, d/h=0).



(g) Normal gradient: XFEM (h=1/200). (h) Normal gradient: IB-DG (low-order, h=1/200). (i) Normal gradient: IB-DG (higher-order, h=1/200).

Figure 15. Corner example: (top-middle) resulting potential contour, Γ-boundary and background mesh; (bottom) normal gradient along Γ-boundary.

DOI: 10.1002/nme

tial around the corner [Figure 15(a,b)] and approximate the gradient very poorly even for high refinements [Figure 15(g,h)]. On the other hand, the higher-order IB-DG approach shows excellent performance as seen in Figure 15(c,i).

In order to perform a convergence study for h-refinement, at each refinement we vary the ratio $d/h \in [0, \ldots, 1]$ as pictured in Figure 14. Three cases are plotted for the higher-order IB-DG in Figure 15(d-f). As Γ comes closer to the element boundary, our method will automatically extend the DG element region (red element boundaries) and merge neighboring elements. The upper bounds on the error for the bulk- and gradient-fields are shown in Figure 16. The higher-order IB-DG method clearly outperforms the low-order approaches. We observe second order convergence in the bulk field as well as the gradient field for the higher-order IB-DG, whereas the low-order methods lock with respect to the surface gradient error. Note especially that despite the previous example with a circular boundary, the Lagrange multiplier is unable to deliver accurate results in this case. Moreover, any post-processing will have difficulties to reconstruct the singular gradients around the corner from a low-order bulk field interpolation.

5.5. *Charge loading*

We next test the case where a total charge Q is imposed on a conductor. We take the geometry from the rectangular boundary given in the previous example. From the analytical approximation, we calculate

$$Q = -\int_\Gamma \epsilon_\mathcal{V}(\nabla\Phi \cdot \mathbf{n})\,\mathrm{da} = -8\epsilon_\mathcal{V}\Phi_0/\pi \int_{\tilde{z}_0}^{-1} \tilde{z}^{-1}\,\mathrm{da}, \tag{37}$$

where we find $\tilde{z}_0 = f^{-1}(z_0)$, with $z_0 = (0.5, g)$ and the mapping f is given by (35). We tabulate typical values for Q at $\Phi_0 = 300$ and various g in Table I.

For the numerical test, we use the same parameters as from the previous example, except we now employ Q as a given load and solve consequently for Φ^h, $\bar{\Phi}$ via (17) in the X-FEM context or (29) and (30) for the IB-DG methods. We plot the convergence in Figure 17 and observe similar results as reported in the previous example, where our high-order IB-DG outperforms the other methods especially with respect to the normal gradient error.

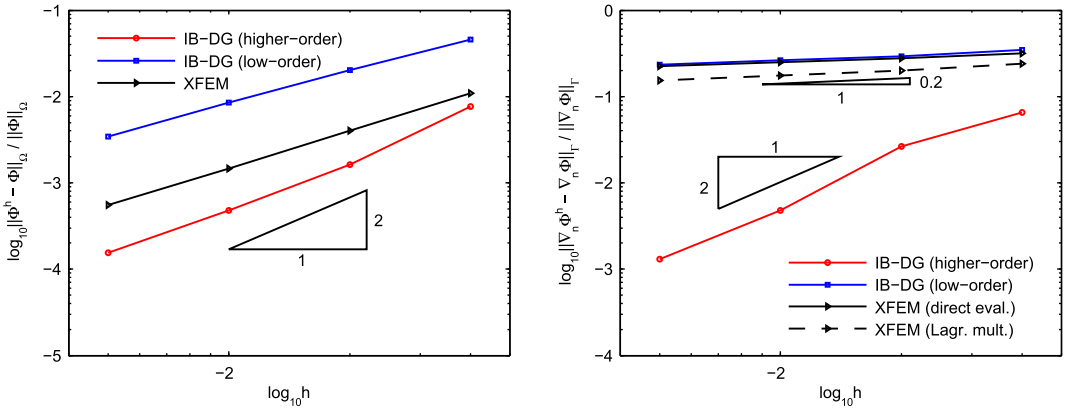


Figure 16. Corner example, voltage loading, maximum L2-error convergence: (left) bulk field; (right) normal gradient.

Table I. Charge loading example: Typical $\tilde{z}_0$ and Q for various gaps g at $\bar{\Phi} = 300$.

g	0.0900	0.0925	0.095	0.0975	0.1000
$\tilde{z}_0$	$-3.9505 \cdot 10^6$	$-2.4648 \cdot 10^6$	$-1.5766 \cdot 10^6$	$-1.0318 \cdot 10^6$	$-6.8970 \cdot 10^5$
Q	$-1.1604 \cdot 10^4$	$-1.1243 \cdot 10^4$	$-1.0902 \cdot 10^4$	$-1.0578 \cdot 10^4$	$-1.0270 \cdot 10^4$

DOI: 10.1002/nme

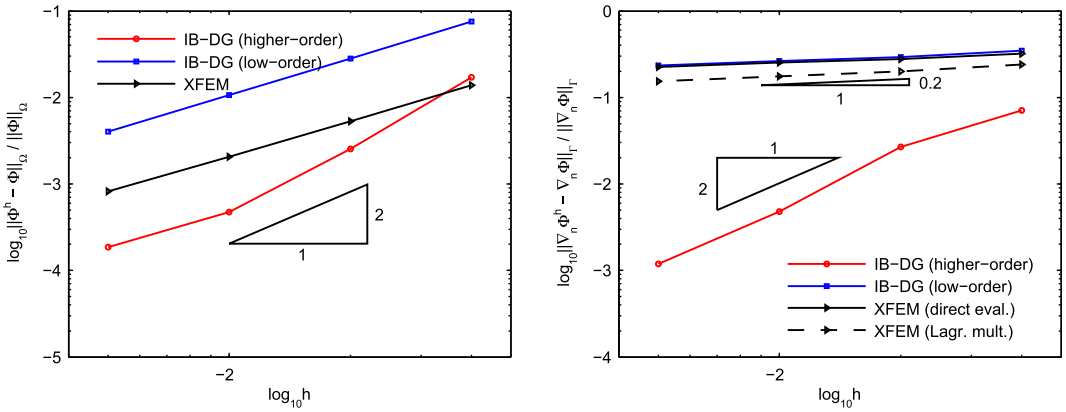


Figure 17. Corner example, charge loading, maximum L2-error convergence: (left) bulk field; (right) normal gradient.

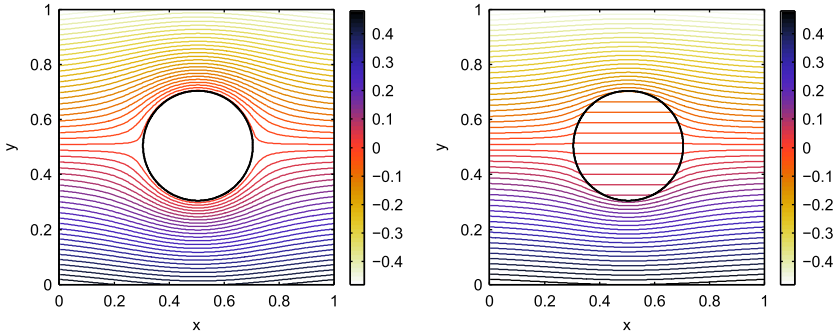


Figure 18. Dielectric-interface example: resulting potential contour immersed boundary discontinuous-Galerkin (IB-DG) (higher-order, h=1/100) for (left) $\epsilon_{\mathcal{R}} = 10^6$ and (right) $\epsilon_{\mathcal{R}} = 3$.

5.6. *Dielectric interface*

For our last example, we test the capability to calculate penetrating fields as well as a discontinuous material permittivity across dielectric-dielectric interfaces. To this end we assume a cylinder with radius R and permittivity $\epsilon_{\mathcal{R}}$ placed in a uniform e-field of strength E_0 in the surrounding infinite space with permittivity $\epsilon_{\mathcal{V}}$. Using a polar coordinate system $\{r, \varphi\}$ with origin at the cylinder center, the analytical solution is given by

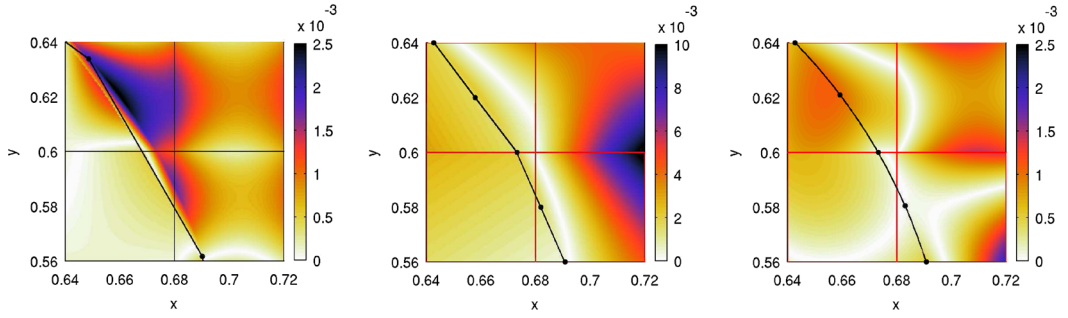
$$\Phi = \begin{cases} -\frac{2\epsilon_{\mathcal{V}}}{\epsilon_{\mathcal{R}}+\epsilon_{\mathcal{V}}} E_0 r \sin\varphi, & \text{if } r < R, \\ -E_0 r \sin\varphi + \frac{\epsilon_{\mathcal{R}}-\epsilon_{\mathcal{V}}}{\epsilon_{\mathcal{R}}+\epsilon_{\mathcal{V}}} E_0 \frac{R^2}{r} \sin\varphi, & \text{if } r \geqslant R, \end{cases} \tag{38}$$

which features a constant electrical field in the interior of the cylinder. For the numerical example we consider the radius $R = 0.2$, permittivity $\epsilon_{\mathcal{V}} = 1$, and e-field strength $E_0 = 1$. We test two different scenarios $\epsilon_{\mathcal{R}} = 10^6$ and $\epsilon_{\mathcal{R}} = 3$. The background mesh has been refined from 25×25 to 200×200 elements. For the Lagrange multiplier space we use $N = 15, \ldots, 120$ linear elements, which corresponds to a ratio $l/h \approx 2.1$ to maintain stability. For the low-order IB-DG we choose $\alpha = 200$, and $\alpha = 450$ for the higher-order IB-DG. In addition we use element extensions with $\delta_1 = 0.6$, $\delta_2 = 1$. No radius r_E needs to be specified because no singularities are present. In this example we impose fixed Dirichlet boundary conditions along the boundary of the computational domain, as we calculate from the analytical solution.

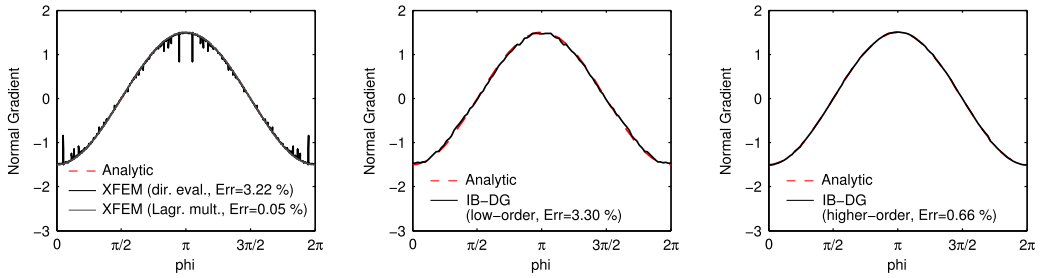
A typical contour plot for the two different $\epsilon_{\mathcal{R}}$ is shown in Figure 18, which nicely shows the constant e-field inside the cylinder. As can be expected, $\epsilon_{\mathcal{R}} = 10^6$ enforces a quasi-vanishing electrical field inside the cylinder, whereas $\epsilon_{\mathcal{R}} = 3$ allows a penetrating field. Because for both

DOI: 10.1002/nme

$\epsilon_{\mathcal{R}}$ we made similar numerical observations, we will focus on $\epsilon_{\mathcal{R}} = 3$ in the following. We draw attention to the detailed point-wise error maps in Figure 19(a-c) around the boundary: for X-FEM we have the large errors occurring near the boundary location, whereas for the IB-DG methods the error concentrates near the element edges. This is expected, because for X-FEM one enforces continuity in a weak sense along Γ, whereas for the IB-DG based methods continuity is enforced in a weak sense along element boundaries of the intersected elements. When looking at the normal gradients in Figure 19(d-e), we observe a smooth approximation via the Lagrange multiplier and higher-order IB-DG, whereas the direct evaluation via X-FEM and low-order IB-DG show some error spikes and more jittery behavior. This is reflected in the convergence plots in Figure 20, where the error



(a) Potential error: XFEM (h=1/25). (b) Potential error: IB-DG (low-order, h=1/25). (c) Potential error: IB-DG (higher-order, h=1/25).



(d) Normal gradient: XFEM (h=1/200). (e) Normal gradient: IB-DG (low-order, h=1/200). (f) Normal gradient: IB-DG (higher-order, h=1/200).

Figure 19. Dielectric-interface example: (top) detailed potential error $|\Phi^h - \Phi|$, Γ-boundary and background mesh with control-points; (bottom) normal gradient along Γ-boundary.

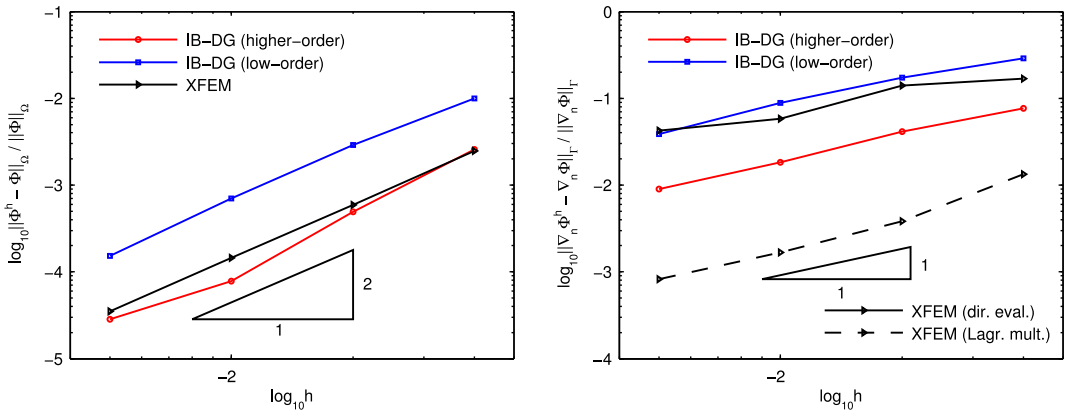


Figure 20. Dielectric-interface example, maximum L2-error convergence: (left) bulk field; (right) normal gradient.

DOI: 10.1002/nme

constants for X-FEM with Lagrange multipliers and high-order IB-DG are smaller. Nevertheless all methods show the same order of convergence for this example.

6. CONCLUSION

We have presented a new implementation of a high-order DG based immersed boundary method. The method is based on a high-order boundary representation, as well as a high-order field approximation in a small band of elements cut by the boundary. The boundary shape is approximated locally by possibly nonlinear geometric primitives. In this paper we develop elements for straight-, circular-, and corner-boundaries in two dimensions. The field approximation is spanned by shape functions that are motivated by the analytical solution of the underlying PDE in the proximity of the corresponding boundary features. Employing this higher-order solution space has several advantages compared to low-order approximations. As a main argument to use, it appears that (i) Dirichlet boundary conditions along Γ can be strongly enforced, (ii) the gradient interpolation is more accurate than low-order embedded boundary methods, (iii) no oscillations occur in the gradient and no post-processing is required to obtain smooth results, and (iv) singularities in the PDE can be incorporated in a natural way. All enhanced elements are coupled together and to the rest of the standard FE domain via DG. The DG-based immersed boundary method is very robust, and we performed several benchmark tests to demonstrate the performance and convergence. The method is also efficient, in the sense that a higher-order interpolation is employed only around the boundary where a high accuracy for the gradient field is needed, whereas the rest of the domain may utilize low-order approximations. We do require heuristic parameters to deal with intersection adaption and singularities. However, the method has been found to be relatively insensitive to these. Moreover, our DG method does feature a stabilization term α, as in most immersed boundary methods there is a stabilization needed at some point. In this work we choose α following a convergence study to obtain optimal results. Whereas this gave us very accurate results in the discussed examples, the selection of α is carried out by the user's choice, and future works may consider alternative DG methodologies that provide for accuracy and stability with an automated stability parameter. The focus of this work is on the enhancement of the gradient accuracy along higher-order immersed boundary shapes, which has been successfully adopted. We note that the basic idea is very simple in two dimensions and can be extended to more complicated boundary shapes or other types of PDEs. In order to deal with three space dimensions, the basic strategy will remain the same, but an extended library of geometric primitives and boundary intersection scenarios will become necessary. While the development of such libraries requires clearly more effort in the three-dimensional setup as compared to the present two-dimensional work, there is a finite set of cases in the end. Once these libraries have been created, we believe that the proposed methodology has good potential to improve upon robustness and accuracy in comparison to current immersed boundary technologies, especially if higher-order boundary features are present or singularities in the field have to be resolved.

APPENDIX A: FAR FIELD BOUNDARY CONDITION VIA BOUNDARY ELEMENT METHOD

We briefly outline our approach to the far field boundary condition we employ at the boundary of the computational domain of interest. To this end we use the boundary element method, which relies upon the boundary integral equation, for each $\boldsymbol{x}_i \in \mathcal{W}$,

$$c\,\Phi_{\mathcal{W}}(\boldsymbol{x}_i) - \int_{\partial\mathcal{W}} \Phi_{\mathcal{W}} q^*_{\mathcal{W}}\,\mathrm{da} = -\int_{\partial\mathcal{W}} q_{\mathcal{W}} \Phi^*_{\mathcal{W}}\,\mathrm{da}, \tag{A.1}$$

where

$$\Phi^*_{\mathcal{W}} = \begin{cases} \frac{1}{2\pi\epsilon_{\mathcal{W}}} \ln\frac{1}{r}, & \text{in 2D,} \\ \frac{1}{4\pi\epsilon_{\mathcal{W}} r}, & \text{in 3D,} \end{cases} \tag{A.2}$$

DOI: 10.1002/nme

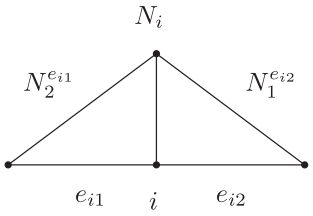


Figure A.1. Coupling finite element and boundary element method.

with $r = \|\mathbf{x} - \mathbf{x}_i\|, q^*_{\mathcal{W}} = -\epsilon_{\mathcal{W}} \nabla \Phi^*_{\mathcal{W}} \cdot \mathbf{n}_{\mathcal{W}}$ and c is a constant depending on the location of the collocation point $\mathbf{x}_i$. If $\mathbf{x}_i$ is inside $\mathcal{W}$, then $c = 1$. If $\mathbf{x}_i$ is on $\partial\mathcal{W}$, then c depends on the smoothness of the boundary (see e.g. [44, p.107])– for a smooth boundary one has $c = 0.5$.

We consider the discretization of the boundary integral Equation (A.1). In this context we will use a point collocation method where we assume that (A.1) holds strongly for $\mathbf{x}_i$ at all nodal points of the mesh $\partial\mathcal{W}^h$. This results in

$$[\Delta\mathbf{Q}]\,\tilde{\mathbf{\Phi}}_{\mathcal{W}} = [\Delta\mathbf{\Phi}]\,\tilde{\mathbf{q}}_{\mathcal{W}}, \tag{A.3}$$

where the i-th row corresponds to collocation point $\mathbf{x}_i$:

$$[\Delta\mathbf{Q}]_{i,:}\,\tilde{\mathbf{\Phi}}_{\mathcal{W}} = c\,\tilde{\Phi}_{\mathcal{W}i} - \sum_e \left[\int_{\partial\mathcal{W}_e} q^*_{\mathcal{W}}(\mathbf{x}_i, \xi)\tilde{\mathbf{N}}_e(\xi)\mathrm{da}(\xi)\right]\tilde{\mathbf{\Phi}}_{\mathcal{W}e} \tag{A.4}$$

$$[\Delta\mathbf{\Phi}]_{i,:}\,\tilde{\mathbf{q}}_{\mathcal{W}} = -\sum_e \left[\int_{\partial\mathcal{W}_e} \Phi^*_{\mathcal{W}}(\mathbf{x}_i, \xi)\tilde{\mathbf{N}}_e(\xi)\mathrm{da}(\xi)\right]\tilde{\mathbf{q}}_{\mathcal{W}e}, \tag{A.5}$$

and $\tilde{\mathbf{N}}_e(\xi)$ are standard shape function matrices. Note the notation (e.g. 2D)

$$\Phi^*_{\mathcal{W}}(\mathbf{x}_i, \xi) = \frac{1}{2\pi\epsilon_{\mathcal{W}}} \ln \frac{1}{\|\mathbf{x}(\xi) - \mathbf{x}_i\|}, \tag{A.6}$$

where ξ is the integration parameter such that $\mathbf{x}(\xi)$ maps to the element integration domain. The matrices $[\Delta\mathbf{Q}]$, $[\Delta\mathbf{\Phi}]$ are fully populated. We highlight three facts: first, the integrals involve singular functions and special care must be taken in order to evaluate them correctly via numerical Gauss or modified Gauss quadrature rules ([44, p.139]). Second, note that the diagonal terms of $[\Delta\mathbf{Q}]$ can be easily obtained by summing up all other coefficients in the corresponding row and changing the sign ([44, p.135], rigid body motion argument); this saves us a strongly singular integration plus the computation of the factor c. And lastly, note that for an infinite domain one must consider the so called *azimuthal integral*, that is the integration over the (semi-)sphere with infinite radius ([44, p.136]). This will only give a contribution to the strongly singular integral; that is, the diagonal terms of $[\Delta\mathbf{Q}]$. In the case of an infinite domain this requires one to add $+1$ to the diagonal, whereas in the semi-infinite case one adds $+0.5$.

In our experience the use of a piecewise constant boundary element discretization performs satisfactorily and is particularly easy to implement. We assume that the BE-nodes are in the middle of each surface patch, and that the constant potential equals the average of the attached domain mesh interpolation. Using constant elements allows us an analytical integration of the singular integrals. In particular note that $[\Delta\mathbf{Q}]_{i,i} = 1.5$ as arising from the azimuthal integral, whereas the strongly singular integral vanishes in this case. Moreover

$$[\Delta\mathbf{\Phi}]_{i,i} = -\int_{\Gamma_{BEi}} \Phi^*_{\mathcal{W}}(\mathbf{x}_i, \xi)\mathrm{da}(\xi) = -2\int_0^{l/2} \frac{1}{2\pi\epsilon_{\mathcal{W}}} \ln\frac{1}{\xi}\mathrm{d}\xi = -\frac{l}{2\pi\epsilon_{\mathcal{W}}}\left[\ln\frac{2}{l} + 1\right], \tag{A.7}$$

where l is the length of the boundary element. For the off-diagonal terms, a standard 4-point Gauss quadrature is used.

DOI: 10.1002/nme

We couple the boundary elements to the standard finite elements as in [44, Ch.16]. One can write (A.3) as

$$\tilde{\mathbf{q}}_{\mathcal{W}} = \tilde{\boldsymbol{K}}_{BE}\tilde{\boldsymbol{\Phi}}_{\mathcal{W}}, \tag{A.8}$$

featuring the 'pseudo'-stiffness matrix

$$\tilde{\boldsymbol{K}}_{BE} = \left[\tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2, \ldots, \tilde{\mathbf{q}}_{N_b}\right], \tag{A.9}$$

where $\tilde{\mathbf{q}}_i$ is the solution to

$$[\Delta\boldsymbol{\Phi}]\,\tilde{\mathbf{q}}_i = [\Delta\mathbf{Q}]_{:,i}\,, \tag{A.10}$$

with $[\Delta\mathbf{Q}]_{:,i}$ the i-th column of the matrix $[\Delta\mathbf{Q}]$. In order to obtain an expression for the finite element flux vector $\mathbf{f}_{BE}$, we note that $q_{\mathcal{V}} = -q_{\mathcal{W}}$ along Γ_{BE}. Thus one can write the equivalent nodal flux at Node i as

$$f_{BEi} = -\sum_{\{e_i\}}\sum_{n=1}^{nbe}\left[\int_{\partial\mathcal{V}_{e_i}\cap\Gamma_{BE}} N_j^{e_i}\,\tilde{N}_n^{be(e_i)}\mathrm{da}\right]\tilde{q}_{\mathcal{W}n}^{be(e_i)}, \tag{A.11}$$

where $\{e_i\}$ ranges over the adjacent elements of node i, $be(e_i)$ is the boundary element number corresponding to the adjacent finite element e_i, and one takes the local finite element shape function $N_j^{e_i}$ associated with the j-th node in the element numbering, which corresponds to the i-th global node (Figure A.1). We write (A.11) as

$$\mathbf{f}_{BE} = -\mathbf{N}\tilde{\mathbf{q}}_{\mathcal{W}}. \tag{A.12}$$

Finally $\boldsymbol{\Phi}_{\mathcal{W}} = \boldsymbol{\Phi}_{\mathcal{V}}$, and we use the projection $\mathbb{P}$, such that

$$\tilde{\boldsymbol{\Phi}}_{\mathcal{W}} = \mathbb{P}\boldsymbol{\Phi}_{\mathcal{W}}, \tag{A.13}$$

relates the BE interpolation to the FE nodal values. We summarize

$$\mathbf{f}_{BE} = -\mathbf{N}\tilde{\boldsymbol{K}}_{BE}\mathbb{P}\boldsymbol{\Phi}_{\mathcal{V}} = -\boldsymbol{K}_{BE}\boldsymbol{\Phi}_{\mathcal{V}}, \tag{A.14}$$

where

$$\boldsymbol{K}_{BE} = \mathbf{N}\tilde{\boldsymbol{K}}_{BE}\mathbb{P} \tag{A.15}$$

is the boundary element stiffness (non-symmetric).

REFERENCES

1. Lew AJ, Buscaglia GC. A discontinuous-Galerkin-based immersed boundary method. *International Journal for Numerical Methods in Engineering* 2008; **76**:427–454.
2. Peskin CS. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics* 1972; **10**: 252–271.
3. Peskin CS. The immersed boundary method. *Acta Numerica* 2002; **11**:479–517.
4. LeVeque RJ, Li Z. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *Journal on Numerical Analysis* 1994; **31**:1019–1044.
5. Glowinski R, Pan TW, Périaux J. A fictitious domain method for Dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering* 1994; **111**:283–303.
6. Glowinski R, Pan TW, Hesla TI, Joseph DD, Périaux J. A distributed Lagrange multiplier/fictitious domain method for flows around moving rigid bodies: Application to particulate flow. *International Journal for Numerical Methods in Fluids* 1999; **30**:1043–1066.
7. Moës N, Dolbow JE, Belytschko T. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering* 1999; **46**:131–150.
8. Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering* 1999; **45**:601–620.
9. Andreykiv A, Rixen DJ. Numerical modelling of electromechanical coupling using fictitious domain and level set methods. *International Journal for Numerical Methods in Engineering* 2009; **80**:478–506.

DOI: 10.1002/nme

10. Rochus V, Van Miegroet L, Rixen DJ, Duysinx P. Electrostatic simulation using XFEM for conductor and dielectric interfaces. *International Journal for Numerical Methods in Engineering* 2011; **85**:1207–1226.
11. Ji H, Dolbow JE. On strategies for enforcing interfacial constraints and evaluating jump conditions with the extended finite element method. *International Journal for Numerical Methods in Engineering* 2004; **61**:2508–2535.
12. Rangarajan R, Lew AJ, Buscaglia GC. A discontinuous-Galerkin-based immersed boundary method with non-homogeneous boundary conditions and its application to elasticity. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**:1513–1534.
13. Babuska I. Error-bounds for finite element method. *Numerische Mathematik* 1971; **16**:322–333.
14. Moës N, Béchet E, Tourbier M. Imposing Dirichlet boundary conditions in the extended finite element method. *International Journal for Numerical Methods in Engineering* 2006; **67**:1641–1669.
15. Béchet E, Moës N, Wohlmuth B. A stable Lagrange multiplier space for stiff interface conditions within the extended finite element method. *International Journal for Numerical Methods in Engineering* 2009; **78**:931–954.
16. Hautefeuille M, Annavarapu C, Dolbow JE. Robust imposition of Dirichlet boundary conditions on embedded surfaces. *International Journal for Numerical Methods in Engineering* 2012; **90**:40–64.
17. 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* 1971; **36**:9–15.
18. Hansbo A, Hansbo P. An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:5537–5552.
19. Heintz P, Hansbo P. Stabilized Lagrange multiplier methods for bilateral elastic contact with friction. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:4323–4333.
20. Mourad HM, Dolbow JE, Harari I. A bubble-stabilized finite element method for Dirichlet constraints on embedded interfaces. *International Journal for Numerical Methods in Engineering* 2007; **69**:772–793.
21. Dolbow JE, Franca LP. Residual-free bubbles for embedded Dirichlet problems. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**:3751–3759.
22. Dolbow JE, Harari I. An efficient finite element method for embedded interface problems. *International Journal for Numerical Methods in Engineering* 2009; **78**:229–252.
23. Oliver J, Hartmann S, Cante JC, Weyler R, Hernández JA. A contact domain method for large deformation frictional contact problems. Part 1: Theoretical basis. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**:2591–2606.
24. Hartmann S, Oliver J, Weyler R, Cante JC, Hernández JA. A contact domain method for large deformation frictional contact problems. Part 2: Numerical aspects. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**:2607–2631.
25. Burman E, Hansbo P. Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:2680–2686.
26. Cheng KW, Fries TP. Higher-order XFEM for curved strong and weak discontinuities. *International Journal for Numerical Methods in Engineering* 2010; **82**:564–590.
27. Gracie R, Wang H, Belytschko T. Blending in the extended finite element method by discontinuous Galerkin and assumed strain methods. *International Journal for Numerical Methods in Engineering* 2008; **74**:1645–1669.
28. Hildebrand FB. *Advanced Calculus for Applications*. Prentice-Hall, Inc.: Englewood Cliffs, N.J., 1962.
29. Douglas, Jr., J, Dupont T. Interior penalty procedures for elliptic and parabolic Galerkin methods. *Computing Methods in Applied Sciences (Second International Symposium, Versailles, 1975), Lecture Notes in Physics* 1975; **58**: 207–216.
30. Wheeler MF. An elliptic collocation-finite element method with interior penalties. *SIAM Journal on Numerical Analysis* 1978; **15**:152–161.
31. Arnold DN. An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis* 1982; **19**:742–760.
32. Bassi F, Rebay S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics* 1997; **131**:267–279.
33. Cockburn B, Shu CW. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis* 1998; **35**:2440–2463.
34. Peraire J, Persson PO. The compact discontinuous Galerkin (CDG) method for elliptic problems. *SIAM Journal on Scientific Computing* 2008; **30**:1806–1824.
35. Embar A, Dolbow JE, Harari I. Imposing Dirichlet boundary conditions with Nitsche's method and spline-based finite elements. *International Journal for Numerical Methods in Engineering* 2010; **83**:877–898.
36. Annavarapu C, Hautefeuille M, Dolbow JE. A robust Nitsche's formulation for interface problems. *Computer Methods in Applied Mechanics and Engineering* 2012; **225-228**:44–54.
37. Annavarapu C, Hautefeuille M, Dolbow JE. Stable imposition of stiff constraints in explicit dynamics for embedded finite element methods. *International Journal for Numerical Methods in Engineering* 2012; **92**:206–228.
38. Mousavi SE, Sukumar N. Generalized Duffy transformation for integrating vertex singularities. *Computational Mechanics* 2010; **45**:127–140.
39. Laborde P, Pommier J, Renard Y, Salaün M. High-order extended finite element method for cracked domains. *International Journal for Numerical Methods in Engineering* 2005; **64**:354–381.
40. Béchet E, Minnebo H, Moës N, Burgardt B. Improved implementation and robustness study of the X-FEM for stress analysis around cracks. *International Journal for Numerical Methods in Engineering* 2005; **64**:1033–1056.

DOI: 10.1002/nme

41. Johansson A, Larson MG. A high order discontinuous Galerkin Nitsche method for elliptic problems with fictitious boundary. *Numerische Mathematik* 2013; **123**:607–628.
42. Landau LD, Lifshitz EM. *Electrodynamics of Continuous Media*. Butterworth-Heinemann: Burlington, MA, USA, 2008.
43. Hannot SDA. Modeling strategies for electro-mechanical microsystems with uncertainty quantification. *Ph.D. Thesis*, TU Delft, 2010.
44. Beer G, Smith I, Duenser C. *The Boundary Element Method with Programming*. Springer Verlag: Wien, 2008.

DOI: 10.1002/nme