



# An Immersed Finite Element Method with Integral Equation Correction

Thomas Rüberg and Fehmi Cirak\*

*Department of Engineering, University of Cambridge, UK*

## SUMMARY

We propose a robust immersed finite element method in which an integral equation formulation is used to enforce essential boundary conditions. The solution of a boundary value problem is expressed as the superposition of a finite element solution and an integral equation solution. For computing the finite element solution, the physical domain is embedded into a slightly larger Cartesian (box-shaped) domain and is discretized using a block-structured mesh. The defect in the essential boundary conditions, which occurs along the physical domain boundaries, is subsequently corrected with an integral equation method. In order to facilitate the mapping between the finite element and integral equation solutions, the physical domain boundary is represented with a signed distance function on the block-structured mesh. As a result, only a boundary mesh of the physical domain is necessary and no domain mesh needs to be generated, except for the non-boundary-conforming block-structured mesh. The overall approach is first presented for the Poisson equation and then generalised to incompressible viscous flow equations. As an example for fluid-structure coupling, the settling of a heavy rigid particle in a closed tank is considered.

KEY WORDS: immersed finite element method, finite elements, integral equations, implicit geometry representation

---

\*Correspondence to: f.cirak@eng.cam.ac.uk

## 1. INTRODUCTION

Finite element discretization of domains with complex, time-dependent geometric features pose challenges in terms of mesh generation and implementation of scalable, adaptive solution algorithms. In particular, if the computational domain boundaries are subject to large deformations, such as in fluid-structure interaction, the mesh may become highly distorted which degrades the conditioning of the discrete problem and the quality of the finite element approximation. The problems related to mesh distortion can partly be remedied by using continuous adaptive remeshing, which requires, especially in the three-dimensional setting, complex and time-consuming algorithms. In order to circumvent mesh generation and continuous remeshing, a number of discretization methods have been proposed, in particular in computational fluid dynamics, which do not rely on boundary conforming meshes. These include, to name a few, the fictitious domain method of Glowinski et. al [1], the immersed boundary method of Peskin [2] and the ghost fluid method of Fedkiw et al. [3]. In these methods, the physical domain is first embedded into a larger Cartesian (box-shaped) domain on which a non-body-fitted, block-structured mesh is used. Subsequently, auxiliary procedures are used for incorporating the physical domain boundaries into the block-structured mesh solution. For a comprehensive overview of interface capturing methods in computational fluid dynamics see the review paper by Mittal and Iaccarino [4].

Whereas most classical interface capturing techniques were developed for finite difference or finite volume methods, the robustness and efficiency of such methods in computing large-scale problems motivated lately research on similar methods in the context of finite element methods. Amongst others, the extended finite element method has been used for developing immersed finite elements for solids discretized with non-boundary-conforming, block-structured meshes [5, 6]. Alternatively, the enforcement of interface or boundary conditions on a non-boundary-conforming mesh can be formulated as a constrained variational problem. Based on this idea, a number of approaches using penalty, Lagrange multiplier or Nitsche methods have been considered for solving the constrained variational problem in the context of immersed finite elements [1, 7]. Although such methods are highly versatile, one of their drawbacks is that their robustness depends on the shape and size of the finite elements which are cut by the domain boundary or interface. If the size of the so-called cut-elements tends to zero, the conditioning of the numerical problem deteriorates. In order to remedy that, the discrete problem is usually regularised by relocating the nodes of the cut-elements. The necessary amount of regularisation depends on the physical problem and discretization parameters, such as the characteristic element size or the polynomial order of shape functions. Although regularisation appears to be a viable approach for practical computations, its dependence on too many parameters makes it hard to generalise.

In order to side-step the problems associated with the generation and updating of domain meshes, it is appealing to resort to an integral equation formulation. If the fundamental solution of a boundary value problem is known, the domain problem can be reformulated as an integral equation involving the domain surface. The related discretization methods are known as integral equation or boundary element methods. Integral equation methods were extensively studied in the past and are still actively pursued albeit to a lesser extent than finite elements, see e.g. [8, 9, 10]. The application areas of integral equation methods are somewhat restricted since they rely on the availability of the fundamental solution of the considered problem. The fundamental solution is not available (at least not in closed form) for most non-linear and spatially inhomogeneous problems. One of the key advantages of the integral equation methods is that they require only a discretization of the domain boundary instead of the domain itself. However, domain integrals still appear due to body forces, in the form of so-called Newton potentials, which in general cannot be reduced to surface integrals so that an integration over the domain is necessary. In a naive implementation of the integral equation method, the evaluation of the volume integrals can become a runtime bottleneck and a number of approaches have been proposed to address that (see, e.g., [11]).

Integral equation methods and domain discretization methods, such as finite elements or finite differences, can also be used in a complementary way for developing fast solvers for problems with complex geometric features. As demonstrated by Mayo for the Poisson equation in [12], it is possible to develop solvers with optimal algorithmic complexity by combining integral equation methods with finite differences. In line with immersed boundary methods, in Mayo's approach the physical domain is embedded into a larger non-body-fitted Cartesian domain and then a Poisson problem is solved using finite differences. The thereby obtained solution does not conform to the boundary

conditions of the original Poisson problem. In a second step, in order to eliminate the errors in the finite difference solution occurring at the domain boundaries, a Laplace equation is solved using the integral equation method. The solution of the original problem is the superposition of the integral equation and finite difference solutions. Although not strictly necessary, in [12] one more Poisson problem is solved for efficiently superimposing both solutions inside the physical domain. In [13], Biros et al. extended Mayo's approach to the Stokes equation and combined integral equations with stabilised  $Q_1$ - $Q_1$  finite elements.

The immersed finite element method developed in this paper enables to robustly solve incompressible Navier-Stokes problems with moving boundaries. It is in spirit similar to Mayo's [12] work on the Poisson equation and Biros et al. [13] work on the stationary Stokes equation. The Navier-Stokes equations are integrated in time using an implicit Euler scheme and the resulting semi-discrete non-linear equations are solved with a fix-point iteration scheme. In each fix-point iteration step, the linearised stationary solution is computed as the superposition of a finite element solution on the Cartesian domain and an integral equation solution on the physical domain boundary. The integral equation corresponds to a Brinkman boundary value problem for which a fundamental solution exists. The Brinkman equation is a modified Stokes equation and is usually used in porous flow problems. On the Cartesian domain, the unknowns in the finite element solution are the velocity and pressure fields, which are interpolated with Taylor-Hood ( $Q_2$ - $Q_1$ ) elements. The discretization of the Cartesian domain is particularly straightforward since a structured grid is used in which all elements have the same geometry. On the physical domain boundary, the integral equation is solved with a single layer formulation using a collocation method, whereby the source distribution is interpolated by piecewise constants within each boundary element. Although the presented method is applicable to flow problems with stationary boundaries and large Reynolds numbers, the target applications are fluid-structure interaction problems with moving interfaces at small Reynolds numbers. As a prototype for fluid-structure interaction, the introduced immersed Navier-Stokes solver is loosely coupled with an embedded rigid body solver. Throughout the computations the structured mesh on the Cartesian domain is kept fixed and the coupling between the fluid and rigid body is achieved through exchanging solution variables on the rigid body surface. Considering the pervasiveness of integral equations in solving fluid-structure interaction problems with Stokes flow (see, e.g., [14, 15]), the developed method enables to extend their applicability to flow problems with small Reynolds numbers.

The outline of this paper is as follows. Section 2 introduces the proposed immersed finite element method with integral equation correction using a scalar boundary value problem. It is demonstrated how the solution of a Poisson problem can be computed as the superposition of a non-boundary-conforming finite element solution and an integral equation solution. In section 3, the solution of the non-stationary Navier-Stokes equation is considered. First the equations are discretized in time and subsequently a fix-point iteration is used in order to cope with nonlinearities. The resulting linear equations are solved with the proposed immersed finite element method with integral equation correction. In section 4, the application of the developed method to problems of fluid-rigid body interaction is introduced. Each of the following sections of the paper include numerical examples which demonstrate the excellent robustness of the method.

## 2. BASIC APPROACH FOR SCALAR BOUNDARY VALUE PROBLEMS

In this section, we introduce the proposed method for linear elliptic boundary value problems with Dirichlet boundary conditions. Although the method is applicable to any partial differential operator for which the fundamental solution is known, we focus in the following on the Laplace operator.

### 2.1. Governing equations and their discretization

We consider the Poisson equation with given Dirichlet boundary conditions in the open  $d$ -dimensional domain  $\Omega \subset \mathbb{R}^d$  with the boundary  $\Gamma = \partial\Omega$

$$-(\nabla^2 u)(\mathbf{x}) = f(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega \tag{1a}$$

$$u(\mathbf{x}) = g(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Gamma, \tag{1b}$$

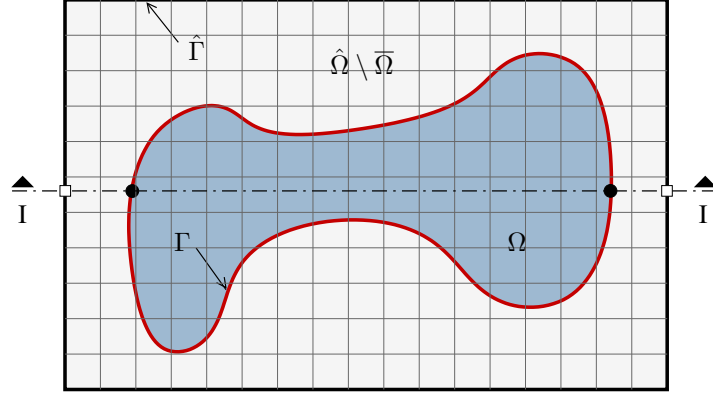


Figure 1. Embedding of the physical domain  $\Omega$  into a larger Cartesian domain  $\hat{\Omega}$  with a block-structured finite element mesh (cut I-I is used in figure 2)

where  $f$  and  $g$  are the prescribed domain loading and boundary conditions, respectively. The corresponding weak form of equation (1) is given by

$$a(u, v) = \ell(v) \quad (2)$$

with the bilinear form  $a(\cdot, \cdot)$  and the linear form  $\ell(\cdot)$

$$a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v \, d\mathbf{x}, \quad \ell(v) = \int_{\Omega} f v \, d\mathbf{x} \quad (3)$$

and  $u \in H_g^1(\Omega)$  and  $v \in H_0^1(\Omega)$ . The relevant function spaces are

$$H_g^1(\Omega) = \{u \in H^1(\Omega) \mid u = g \text{ on } \Gamma\}, \quad (4)$$

which implies  $H_0^1$  with  $g = 0$ , and  $H^1$  is the Sobolev space of square-integrable functions with square-integrable first order derivatives.

If the weak form (2) is discretized using a boundary-conforming mesh, the standard finite element formalism provides appropriate approximation spaces which are subsets of the relevant continuous function spaces,  $H_g^1(\Omega)$  and  $H_0^1(\Omega)$ , respectively. Hence, the computed finite element solution inherently satisfies the Dirichlet boundary conditions (at least at the boundary nodes).

In immersed finite element methods, the problem domain  $\Omega$  is embedded into a larger Cartesian (i.e., box-shaped) domain  $\hat{\Omega}$  with the boundary  $\hat{\Gamma}$  on which a block-structured mesh is used, see Figure 1. It is important that the physical domain  $\Omega$  is fully contained in  $\hat{\Omega}$  so that

$$\Omega \subset \hat{\Omega} \quad \text{and} \quad \text{distance}(\Gamma, \hat{\Gamma}) > \delta, \quad (5)$$

where  $\delta$  is a positive real number which is larger than the grid distance of the block-structured mesh.

In a first step, we ignore the Dirichlet boundary conditions  $u = g$  on the physical boundary  $\Gamma$  and consider on the Cartesian domain the extended weak form

$$\hat{a}(\hat{u}_1, \hat{v}) = \hat{\ell}(\hat{v}) \quad (6)$$

with  $\hat{u}_1 \in H_0^1(\hat{\Omega})$  and  $\hat{v} \in H_0^1(\hat{\Omega})$ . Here, the hat indicates variables which are defined on the Cartesian domain  $\hat{\Omega}$ . In equation (6) the integrations are carried out over  $\hat{\Omega}$  and, moreover, as the source term  $\hat{f}$  a suitable extension of  $f$  is used. Since the regularity requirements on  $f$  are usually low, such an extension is easily constructed, e.g., by zero extension into the fictitious domain  $\hat{\Omega} \setminus \Omega$ . Note that the Dirichlet boundary conditions on the boundary of the Cartesian domain  $\hat{\Gamma}$  can be chosen arbitrarily; here,  $\hat{u}_1 = 0$  is chosen for simplicity. Obviously, the solution  $\hat{u}_1$  of the Cartesian domain problem (6) does not satisfy the Dirichlet boundary condition of the original problem (1b), which leads to a boundary defect

$$h(\mathbf{x}) = g(\mathbf{x}) - \hat{u}_1(\mathbf{x}) \neq 0 \quad \text{for } \mathbf{x} \in \Gamma. \quad (7)$$

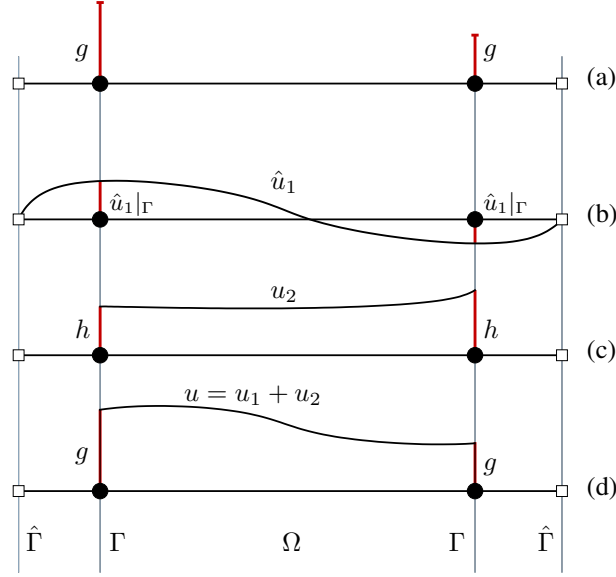


Figure 2. Components of the solution along the cut I-I shown in figure 1: (a) boundary condition  $u = g$  on the physical boundary  $\Gamma$ , (b) solution  $\hat{u}_1$  of the Cartesian domain problem, (c) corrector  $u_2$  with defect  $h = g - \hat{u}_1|_{\Gamma}$  and (d) final solution  $u$

To enforce the Dirichlet boundary conditions of the original problem (1) we consider an auxiliary homogeneous boundary value problem on the physical domain  $\Omega$  with

$$\begin{aligned} -(\nabla^2 u_2)(\mathbf{x}) &= 0 & \text{for } \mathbf{x} \in \Omega \\ u_2(\mathbf{x}) &= h(\mathbf{x}) & \text{for } \mathbf{x} \in \Gamma. \end{aligned} \quad (8)$$

The solution  $u$  of the original problem can now be recovered as the sum of the restriction of the Cartesian domain solution  $\hat{u}_1$  to the original (physical) domain  $\Omega$  and the corrector  $u_2$

$$u = (\hat{u}_1)|_{\Omega} + u_2 = u_1 + u_2. \quad (9)$$

In fact, the components  $u_1$  and  $u_2$  can be viewed as a combination of a particular and the corresponding homogeneous solution of the original problem.

It remains to specify the discretization method for the auxiliary boundary value problem (8). This problem is homogeneous but inherits the geometric complexity of the physical domain  $\Omega$ . Staying with a classical finite element approach, the difficulty of constructing suitable approximation spaces for the original problem would be deferred to this auxiliary problem. Therefore, we use an integral equation method for computing  $u_2$ . The advantage of this choice is that only a surface triangulation is needed for the discretization of the corresponding integral equation.

In figure 2, the section along the cut I-I shown in figure 1 is depicted. With this illustration at hand, the presented solution method consists of the following steps:

1. compute the solution  $\hat{u}_1$  of the Cartesian domain problem (6) using a block-structured grid and the finite element method,
2. evaluate the boundary defect  $h$  along the physical boundary according to equation (7),
3. compute the corrector  $u_2$  as the solution of the auxiliary problem (8) using an integral equation formulation,
4. compute the final solution  $u$  as the superposition of the restriction of  $\hat{u}_1$  to the physical domain  $\Omega$  and the corrector  $u_2$ .

## 2.2. Finite element solution $\hat{u}_1$

Due to the extension  $\Omega \rightarrow \hat{\Omega}$ , it is straightforward to construct suitable approximations of the solution space  $H_0^1(\hat{\Omega})$ . In particular, the choice of a Cartesian domain for  $\hat{\Omega}$  enables to use a simple block-structured mesh. The corresponding

trial functions  $\varphi_I$  are the piecewise polynomials on the elements associated with the grid nodes in the interior of  $\hat{\Omega}$ , which span the subspace

$$V^h = \text{span}\{\varphi_I\} \subset H_0^1(\hat{\Omega}). \quad (10)$$

Then, the finite element approximation is of the form

$$\hat{u}_1(\mathbf{x}) \approx \hat{u}_1^h(\mathbf{x}) = \sum_{I=1}^{N_1} \varphi_I(\mathbf{x}) u_{1,I}, \quad (11)$$

where  $N_1 = \dim(V^h)$  is the dimension of the finite element space  $V^h$ . The subsequent steps in computing the finite element solution  $\hat{u}_1$  are the same as in standard finite elements and can be found in any finite element textbook.

### 2.3. Integral equation correction $u_2$

The solution  $u_2$  of the auxiliary homogeneous boundary value problem (8) can be represented by means of a single layer potential

$$u_2(\mathbf{x}) = \int_{\Gamma} U^*(\mathbf{x} - \mathbf{y}) q(\mathbf{y}) \, ds_{\mathbf{y}}, \quad \mathbf{x} \in \Omega, \quad (12)$$

where  $U^*(\mathbf{x} - \mathbf{y})$  is the fundamental solution of the Laplace operator

$$U^*(\mathbf{x} - \mathbf{y}) = -\frac{1}{2\pi} \log |\mathbf{x} - \mathbf{y}|. \quad (13)$$

The function  $q: \Gamma \rightarrow \mathbb{R}$  is the, yet unknown, surface density which is determined from

$$h(\mathbf{x}) = \int_{\Gamma} U^*(\mathbf{x} - \mathbf{y}) q(\mathbf{y}) \, ds_{\mathbf{y}}, \quad \mathbf{x} \in \Gamma. \quad (14)$$

This integral equation is obtained by taking the limit  $\Omega \ni \tilde{\mathbf{x}} \rightarrow \mathbf{x} \in \Gamma$  in equation (12) and using the Dirichlet boundary condition of problem (8). Note that the integration in equation (14) has to be understood in an improper sense, because the integrand has a weak singularity as  $\mathbf{y}$  approaches  $\mathbf{x}$  (see, e.g., [16]). To obtain an approximation to  $u_2$  using equation (12), first a suitable discretization of the integral equation (14) has to be introduced. To this end the surface density  $q$  and the boundary value  $h$  are approximated with

$$q(\mathbf{x}) \approx \sum_J \psi_J(\mathbf{x}) q_J \quad \text{and} \quad h(\mathbf{x}) \approx \sum_K \chi_K(\mathbf{x}) h_K, \quad (15)$$

where  $\psi_J$  and  $\chi_K$  are the shape functions. Owing to the mapping properties of the boundary integral operator in equation (14), it is possible to use piecewise constant polynomials for the trial functions  $\psi_J$  and piecewise linear continuous polynomials for the interpolation functions  $\chi_K$ , see [8, 10, 16]. For higher order approximations it is important that the trial functions  $\psi_J$  allow for discontinuous source distributions at domain corners.

To solve integral equation (14) a collocation method is used. Hence, at each collocation point  $\mathbf{x}_L^* \in \Gamma$  on the physical boundary the condition to be satisfied is

$$\sum_J \int_{\Gamma} U^*(\mathbf{x}_L^* - \mathbf{y}) \psi_J(\mathbf{y}) \, ds_{\mathbf{y}} q_J = \sum_K h_K \chi_K(\mathbf{x}_L^*). \quad (16)$$

Choosing sufficiently many and well-placed collocation points  $\mathbf{x}_L^*$ , equation (16) yields a system of linear equations for determining the unknown source densities  $q_J$ . Note that the related matrix coefficients are in general non-zero for all combinations of  $\psi_J$  and  $\mathbf{x}_L^*$  so that the resulting system matrix is fully populated. Once, the source densities  $q_J$  have been computed, the approximate correction is computed with the single layer potential at any point  $\mathbf{x}$  inside the physical domain  $\Omega$ :

$$u_2^h(\mathbf{x}) = \sum_J \int_{\Gamma} U^*(\mathbf{x} - \mathbf{y}) \psi_J(\mathbf{y}) \, ds_{\mathbf{y}} q_J. \quad (17)$$

To avoid the rather involved analytical computation of the integrals in equations (16) and (17), numerical quadrature is used. The application of quadrature rules to these integrals is not straightforward due to the divergent behaviour of

the integral kernels as the integration variable  $\mathbf{y}$  approaches the collocation point  $\mathbf{x}_L^*$ . In the integration in equation (16) two cases can be distinguished: (i) singular integration if  $\mathbf{x}_L^*$  is in the support of  $\psi_J(\mathbf{y})$  or (ii) regular integration if this is not the case. In the singular case, first a coordinate transformation is applied which removes the singularity. In the two-dimensional implementation presented in this work the semi-sigmoidal transform suggested by Johnston [17] is used. Although there is no distinct singularity in the regular case, it is possible that the point  $\mathbf{x}_L^*$  is very close to the support of  $\psi_J(\mathbf{y})$ . For this so-called quasi-singular integration, an adaptive quadrature rule is used which recursively subdivides the integration interval according to some heuristic rule depending on the size of this interval and its minimal distance to the collocation point. For further details on this approach see [18]. Note that the adaptive integration is also well suited for evaluating the integrals appearing in equation (17).

#### 2.4. Implicit boundary representation

In the proposed method, it is necessary to determine the position of the finite element nodes of the structured mesh with respect to the physical domain boundary  $\Gamma$ . It is, for instance, necessary to know if a node is located inside or outside the physical domain  $\Omega$ . To this end, a level set representation of the domain surface  $\Gamma$  in form of a signed distance field  $\phi(\mathbf{x}, \Gamma)$  is used, see [19],

$$\phi(\mathbf{x}, \Gamma) = \begin{cases} \text{distance}(\mathbf{x}, \Gamma) & \text{if } \mathbf{x} \in \Omega \\ 0 & \text{if } \mathbf{x} \in \Gamma \\ -\text{distance}(\mathbf{x}, \Gamma) & \text{otherwise.} \end{cases} \quad (18)$$

This function yields the distance between the point  $\mathbf{x}$  and the domain surface  $\Gamma$  and is positive if  $\mathbf{x}$  is inside  $\Omega$  and negative else. The domain surface  $\Gamma$  coincides with the zero contour of the level set function  $\phi(\mathbf{x}, \Gamma)$ . In this work, the signed distance function is computed using the highly efficient closest point algorithm given in [20].

#### 2.5. Illustrative example

As an illustrative example we consider the Poisson equation

$$-\nabla^2 u = 1 \quad \text{for } \mathbf{x} \in \Omega, \quad u = 0 \quad \text{for } \mathbf{x} \in \Gamma \quad (19)$$

on a circular domain of radius 0.5, i.e.,  $\Omega = \{\mathbf{x} \in \mathbb{R}^2 \mid |\mathbf{x}| < 0.5\}$ . The analytical solution of this equation is

$$u(\mathbf{x}) = \frac{1}{16} (1 - 4|\mathbf{x}|^2) . \quad (20)$$

The Cartesian domain is chosen to be  $\hat{\Omega} = (-0.6, 0.6) \times (-0.6, 0.6)$  which is discretized with four-node rectangular finite elements. The surface is discretized with piecewise linear functions for the boundary defect  $h$  and piecewise constant functions for the surface density  $q$ . This example is solved using different mesh combinations which are denoted by the numbers  $N_{FE}/N_{IE}$ .  $N_{FE}$  represents the number of finite element nodes in each direction (resulting in  $N_{FE}^2$  total nodes).  $N_{IE}$  refers to the number of boundary segments. Figure 3 shows a representative numerical solution and its finite element and integral equation components with a mesh combination 41/40.

The error in the solution  $e(\mathbf{x}) = |u(\mathbf{x}) - u^h(\mathbf{x})|$  is computed with the known analytical solution (20). The error distribution along the diagonal  $x_1 = x_2$  from the centre of the circle to its boundary is plotted in figure 4 for different mesh sizes. The coarsest mesh combination is 41/40 and three finer meshes are obtained by quadrisecting the bilinear finite elements and bisecting the boundary elements. Clearly, the error decreases in overall magnitude for decreasing mesh sizes. Moreover, one can detect from the oscillatory pattern of the error curves (which is most pronounced for the finest mesh), the higher accuracy when boundary segments pass through superconvergent points of the finite element mesh. On the other hand, the results appear to be less accurate for points closer to the surface  $\Gamma$ . This is explained by the loss of accuracy in evaluating the single layer potential (12) when the evaluation point gets closer to the integration range. Hence, this is an artifact of post-processing and, notably, it disappears with decreasing mesh size.

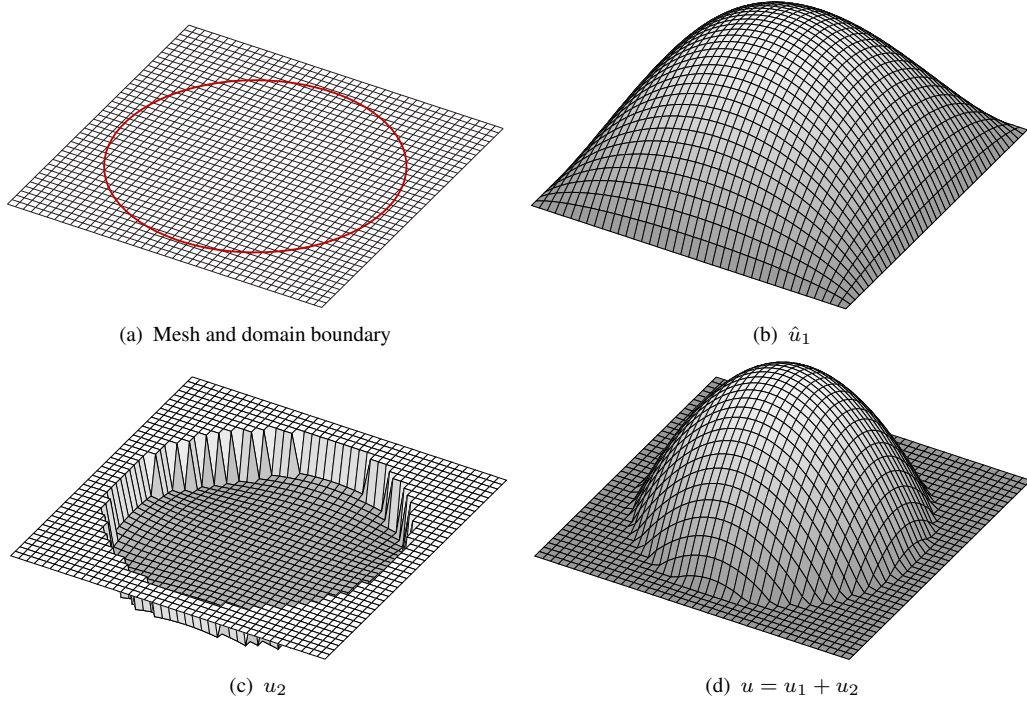


Figure 3. Solution of the Poisson equation on a circular domain: (a) Cartesian domain with the structured mesh and the circle describing the domain boundary, (b) finite element solution  $\hat{u}_1$  on the Cartesian domain, (c) integral equation solution  $u_2$  and (d) the solution  $u = (\hat{u}_1)|_{\Omega} + u_2$  of the original problem

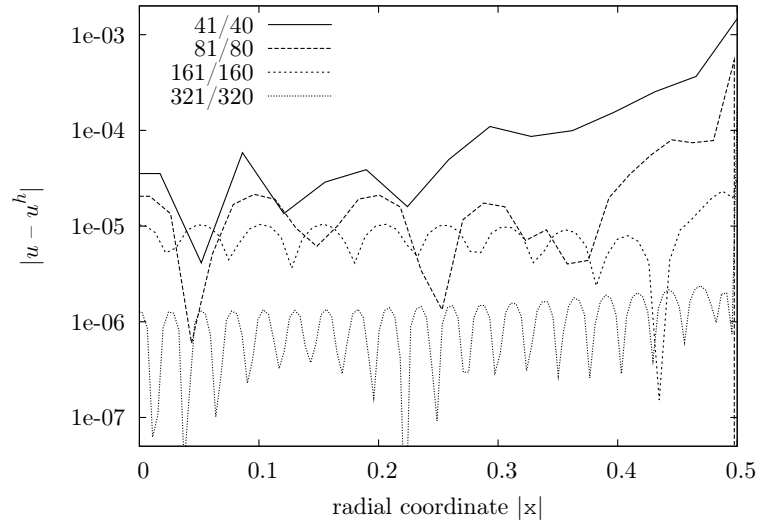


Figure 4. Error in the solution along the diagonal ( $x_1 = x_2$ ) of the circle for four different meshes



### 3. VISCOUS INCOMPRESSIBLE FLOW

#### 3.1. Navier-Stokes equations

Viscous incompressible fluid flow is described by the Navier-Stokes equations which can be expressed in the stress-divergence form as

$$\rho \frac{D\mathbf{u}}{Dt} - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, p) = \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0, \quad (21)$$

where  $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$  denotes the velocity vector and  $\rho$  the mass density. The Newtonian fluid stress tensor  $\boldsymbol{\sigma}$  is defined as

$$\boldsymbol{\sigma}(\mathbf{u}, p) = -p\mathbf{I} + \mu[\nabla\mathbf{u} + (\nabla\mathbf{u})^\top]. \quad (22)$$

where  $p = p(\mathbf{x}, t)$  is the hydrostatic pressure,  $\mathbf{I}$  the identity tensor, and  $\mu$  is the dynamic viscosity. The first equation in (21) is the momentum balance with the total time derivative

$$\frac{D\mathbf{u}}{Dt} = \frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}. \quad (23)$$

The second equation in (21) is the so-called incompressibility condition  $\nabla \cdot \mathbf{u} = 0$ , which represents the mass balance for an incompressible fluid with constant mass density. Inserting the expression for the stress tensor (22) and the total time derivative (23) into the momentum balance equation (21) and making use of the incompressibility condition, yields the initial boundary value problem for viscous incompressible flow

$$\begin{aligned} \frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \nabla^2 \mathbf{u} + \frac{1}{\rho} \nabla p &= \frac{1}{\rho} \mathbf{f}, & \nabla \cdot \mathbf{u} &= 0 & \text{for } \mathbf{x} \in \Omega \\ \mathbf{u} &= \mathbf{g} & \text{for } \mathbf{x} \in \Gamma \\ \mathbf{u}(\cdot, t=0) &= 0 & \text{for } \mathbf{x} \in \Omega \end{aligned} \quad (24)$$

with the prescribed boundary velocities  $\mathbf{g}$  and the kinematic viscosity  $\nu = \mu/\rho$ .

To derive the semi-discrete Navier-Stokes equations, the time variable is subdivided into subintervals  $t_0 = 0 < t_1 < \dots < t_n < t_{n+1}$ . Furthermore, the shorthand notation  $\mathbf{u}_n = \mathbf{u}^h(\cdot, t_n)$ ,  $p_n = p^h(\cdot, t_n)$ ,  $\mathbf{f}_n = \mathbf{f}(\cdot, t_n)$  and  $\Delta t = t_{n+1} - t_n$  is introduced. The time discretization of the Navier-Stokes equations (24) with the backward Euler method gives the nonlinear recurrence relation

$$\frac{1}{\Delta t} \mathbf{u}_{n+1} + (\mathbf{u}_{n+1} \cdot \nabla) \mathbf{u}_{n+1} - \nu \nabla^2 \mathbf{u}_{n+1} + \frac{1}{\rho} \nabla p_{n+1} = \frac{1}{\Delta t} \mathbf{u}_n + \frac{1}{\rho} \mathbf{f}_{n+1} \quad (25)$$

subject to  $\nabla \cdot \mathbf{u}_{n+1} = 0$ . If the solutions  $\mathbf{u}_n$  and  $p_n$  at time  $t_n$  are known, the solution at  $t_{n+1}$  can be computed with a fixed point iteration (see, e.g., [21])

$$\begin{aligned} \frac{1}{\Delta t} \mathbf{u}^{(k+1)} + (\mathbf{u}^{(k)} \cdot \nabla) \mathbf{u}^{(k+1)} - \nu \nabla^2 \mathbf{u}^{(k+1)} + \frac{1}{\rho} \nabla p^{(k+1)} &= \frac{1}{\Delta t} \mathbf{u}_n + \frac{1}{\rho} \mathbf{f}_{n+1} \\ \nabla \cdot \mathbf{u}^{(k+1)} &= 0, \end{aligned} \quad (26)$$

where some of the time indices have been dropped and the superscripts  $(\cdot)^{(k)}$  refer to the  $k$ -th iterate. This iteration scheme is initialised with the predictor  $\mathbf{u}^{(k)} = \mathbf{u}_n$  and converges to the new solution  $\mathbf{u}_{n+1} = \mathbf{u}^{(k+1)}$  at  $t_{n+1}$ . Importantly, equation (26) is linear in the unknowns  $\mathbf{u}^{(k+1)}$  and  $p^{(k+1)}$ . Therefore, these unknowns can again be written as the superposition of a finite element solution on a Cartesian domain  $\mathbf{u}_1$  and integral equation correction  $\mathbf{u}_2$

$$\mathbf{u}^{(k+1)} = \mathbf{u}_1 + \mathbf{u}_2. \quad (27)$$

There is a slight abuse of notation here since the indices in  $\mathbf{u}_1$  and  $\mathbf{u}_2$  do not refer to time. After introducing these decompositions into the fixed point iteration equation (26) two linear system of equations are obtained for computing  $\mathbf{u}_1$  and  $\mathbf{u}_2$ .

### 3.2. Finite element solution $(\hat{\mathbf{u}}_1, \hat{p}_1)$

The solution pair  $(\hat{\mathbf{u}}_1, \hat{p}_1)$  is obtained by solving the following linear equation system using a finite element method

$$\begin{aligned} \frac{1}{\Delta t} \hat{\mathbf{u}}_1 + (\hat{\mathbf{u}}^{(k)} \cdot \nabla) \hat{\mathbf{u}}_1 - \nu \nabla^2 \hat{\mathbf{u}}_1 + \frac{1}{\rho} \nabla \hat{p}_1 &= \frac{1}{\Delta t} \hat{\mathbf{u}}_n + \frac{1}{\rho} \hat{\mathbf{f}}_{n+1} - (\hat{\mathbf{u}}^{(k)} \cdot \nabla) \hat{\mathbf{u}}_2 & \text{for } \mathbf{x} \in \hat{\Omega} \\ \nabla \cdot \hat{\mathbf{u}}_1 &= 0 & \text{for } \mathbf{x} \in \hat{\Omega} \\ \hat{\mathbf{u}}_1 &= 0 & \text{for } \mathbf{x} \in \hat{\Gamma} \end{aligned} \quad (28)$$

where  $\hat{\Omega}$  is the Cartesian domain and  $\hat{\Gamma}$  is its boundary. Further, recall that  $\hat{\mathbf{u}}_1$  and  $\hat{p}_1$  denote the extensions of  $\mathbf{u}_1$  and  $p_1$  from the physical domain to the Cartesian domain, respectively. The specification of these extensions is deferred to section 3.4. In equation (28), note the appearance of the unknown integral equation correction  $\hat{\mathbf{u}}_2$  on the right hand side. Since the advection term has to be treated in an explicit manner for the boundary integral correction part (the operator would otherwise be inhomogeneous), it has to be handled as a body force in the finite element solution step in order to maintain consistency.

The finite element discretization of the equation (28) requires a weak form which reads

$$\begin{aligned} \frac{1}{\Delta t} (\hat{\mathbf{u}}_1, \hat{\mathbf{v}})_{\hat{\Omega}} + \hat{c}(\hat{\mathbf{u}}^{(k)}, \hat{\mathbf{u}}_1, \hat{\mathbf{v}}) + \hat{a}(\hat{\mathbf{u}}_1, \hat{\mathbf{v}}) - \hat{b}(\hat{p}_1, \hat{\mathbf{v}}) &= \tilde{\ell}(\mathbf{v}) \\ \hat{b}(\hat{\gamma}, \hat{\mathbf{u}}_1) &= 0 \end{aligned} \quad (29)$$

with the trial functions  $(\hat{\mathbf{u}}_1, \hat{p}_1) \in [H_0^1(\hat{\Omega})]^d \times L_2^0(\hat{\Omega})$  and the test functions  $(\hat{\mathbf{v}}, \hat{\gamma}) \in [H_0^1(\hat{\Omega})]^d \times L_2^0(\hat{\Omega})$ . The abbreviations in equation (29) are as follows

$$\begin{aligned} \hat{a}(\hat{\mathbf{u}}_1, \hat{\mathbf{v}}) &= \int_{\hat{\Omega}} \nu \nabla \hat{\mathbf{u}}_1 : \nabla \hat{\mathbf{v}} \, d\mathbf{x}, \\ \hat{c}(\mathbf{w}, \hat{\mathbf{u}}_1, \hat{\mathbf{v}}) &= \int_{\hat{\Omega}} [(\mathbf{w} \cdot \nabla) \hat{\mathbf{u}}_1] \cdot \hat{\mathbf{v}} \, d\mathbf{x}, \\ \hat{b}(\hat{\gamma}, \hat{\mathbf{v}}) &= \int_{\hat{\Omega}} \frac{1}{\rho} \hat{\gamma} (\nabla \cdot \hat{\mathbf{v}}) \, d\mathbf{x}, \\ \tilde{\ell}(\mathbf{v}) &= \frac{1}{\rho} \int_{\hat{\Omega}} \hat{\mathbf{f}}_{n+1} \cdot \hat{\mathbf{v}} \, d\mathbf{x} + \frac{1}{\Delta t} (\hat{\mathbf{u}}_n, \hat{\mathbf{v}})_{\hat{\Omega}} - \hat{c}(\hat{\mathbf{u}}^{(k)}, \hat{\mathbf{u}}_2^{(k)}, \hat{\mathbf{v}}), \end{aligned} \quad (30)$$

where  $(\mathbf{a}, \mathbf{b})_{\hat{\Omega}} = \int_{\hat{\Omega}} \mathbf{a} \cdot \mathbf{b} \, d\mathbf{x}$  is the  $L_2$ -product, and  $L_2^0(\hat{\Omega}) = \{\hat{\gamma} \in L_2 \mid \int_{\hat{\Omega}} \hat{\gamma} \, d\mathbf{x} = 0\}$  is the space of all square-integrable functions whose mean vanishes. This special space is introduced because the pressure is only known up to an arbitrary constant for the Navier-Stokes equations with only Dirichlet boundary conditions. In order to ensure this vanishing mean and thus guarantee solvability, the second equation in (29) is actually augmented by the term  $d(\hat{p}_1, \hat{\gamma}) = (\hat{p}, 1)_{\hat{\Omega}} (\hat{\gamma}, 1)_{\hat{\Omega}} = 0$ . As well known, the finite element discretization of the variational statement (29) has to be performed with care in order to avoid stability problems (see, e.g., [22]). Here, the Taylor-Hood element is chosen which is inherently LBB-stable [22].

### 3.3. Integral equation correction $(\mathbf{u}_2, p_2)$

The correction component  $\mathbf{u}_2$  of the solution is obtained from the linear homogeneous boundary value problem

$$\begin{aligned} \frac{\rho}{\Delta t} \mathbf{u}_2 - \mu \nabla^2 \mathbf{u}_2 + \nabla p_2 &= \mathbf{0}, & \nabla \cdot \mathbf{u}_2 &= 0 & \text{for } \mathbf{x} \in \Omega \\ \mathbf{u}_2 &= \mathbf{h} & \text{for } \mathbf{x} \in \Gamma \end{aligned} \quad (31)$$

with the boundary defect  $\mathbf{h} = \mathbf{g} - (\hat{\mathbf{u}}_1)|_{\Gamma}$ . Recall that  $\mathbf{g}$  is the prescribed velocity boundary condition of the original problem and  $\hat{\mathbf{u}}_1$  is the finite element solution on the Cartesian domain. Equation (31) is known as the Brinkman or modified Stokes problem and the corresponding fundamental solution is given in the appendix in equation (50). Using the fundamental solution for the velocities,  $\mathbf{u}_2$  can be represented with the single layer potential

$$\mathbf{u}_2(\mathbf{x}) = \int_{\Gamma} \mathbf{U}^*(\mathbf{x} - \mathbf{y}) \mathbf{q}(\mathbf{y}) \, d\mathbf{s}_{\mathbf{y}}, \quad \mathbf{x} \in \Omega, \quad (32)$$

where  $\mathbf{q}$  is the unknown vectorial surface density, which is determined from the integral equation

$$\mathbf{h}(\mathbf{x}) = \int_{\Gamma} \mathbf{U}^*(\mathbf{x} - \mathbf{y}) \mathbf{q}(\mathbf{y}) \, ds_{\mathbf{y}}, \quad \mathbf{x} \in \Gamma. \quad (33)$$

The discretization and solution of this equation is carried out along the lines of the scalar problem introduced in section 2.3.

The pressure distribution is computed with the pressure single layer potential

$$p_2(\mathbf{x}) = \int_{\Gamma} \mathbf{P}^*(\mathbf{x} - \mathbf{y}) \mathbf{q}(\mathbf{y}) \, ds_{\mathbf{y}}, \quad \mathbf{x} \in \Omega \quad (34)$$

using the surface density  $\mathbf{q}$  determined by equation (33). For Navier-Stokes equations with pure Dirichlet boundary conditions, the pressure distribution  $p$  can only be determined up to a constant  $c$ . The same is true for the surface traction  $\mathbf{t} = \boldsymbol{\sigma}(\mathbf{u}, p) \cdot \mathbf{n}$ , where  $\mathbf{n}$  is the unit surface normal. According to the stress definition (22), replacing  $p$  by some shifted value  $p + c$  alters the traction by  $-c\mathbf{n}$ . This deficiency carries over to the single layer potential (32), see [16] for mathematical details in case of the Stokes system. As a result, the vector field  $\mathbf{n}$  is an eigensolution of the single layer integral equation with corresponding zero eigenvalue, or formally,

$$\int_{\Gamma} \mathbf{U}^*(\mathbf{x} - \mathbf{y}) \mathbf{n} \, ds_{\mathbf{y}} = \mathbf{0}. \quad (35)$$

Therefore, the discretized integral equations are ill-conditioned. In fact, the corresponding system matrix has a zero eigenvalue and the associated eigenvector is a discrete representation of the surface normal field  $\mathbf{n}$ . In order to resolve this problem, the stabilisation presented in [23] for the Galerkin boundary element method is adapted to the used collocation method. This stabilisation is carried out by adding to each  $d \times d$ -matrix block of the system matrix (here,  $d = 2$  for two-dimensional problems), the matrix block of the form

$$\mathbf{N}_{LJ} = \frac{1}{\mu} \mathbf{n}(\mathbf{x}_L^*) \otimes \int_{\Gamma} \mathbf{n}(\mathbf{y}) \psi_J(\mathbf{y}) \, ds_{\mathbf{y}}, \quad (36)$$

which corresponds to the collocation point  $\mathbf{x}_L^*$  and the shape function  $\psi_J$ . Note that this procedure is equivalent to a deflation of the corresponding eigenvalues in matrix analysis [24]. The factor of  $1/\mu$  in equation (36) is introduced to ensure that the extra matrix entries are of similar magnitude as the original matrix entries stemming from equation (16). In case of a multiply connected boundary surface, this stabilisation technique has to be generalised accordingly, see [23] for details.

All the expressions given so far in this section refer to the unsteady flow case. Nevertheless, the steady state limit is recovered by neglecting the expressions with the partial time derivatives from the governing and subsequent equations. The only noteworthy change involves the integral equation solution which reduces to a Stokes equation with the corresponding stokeslet as the fundamental solution (see appendix, equation (48)).

### 3.4. Extrapolation of the physical domain variables

In order to ensure the convergence and robustness of the proposed algorithm, special care needs to be taken in extending physical domain variables into the fictitious domain. In particular, in the iteration (28) for computing the finite element solution  $\hat{\mathbf{u}}_1$ , the integral equation solution  $\hat{\mathbf{u}}_2$  over the entire Cartesian domain is required. However, the integral equation solution outside of the physical domain is not defined and is not of physical relevance. Thus, it can be chosen in  $\hat{\Omega} \setminus \bar{\Omega}$  based on ease of numerical implementation.

To motivate our particular choice of the extension operator used, consider the situation depicted in figure 5. The finite element nodes are represented by squares and the nodes of the surface mesh by circles. Moreover, the nodes  $\mathbf{x}_I$  and  $\mathbf{x}_{I+1}$  are located in the physical domain  $\Omega$  and the nodes  $\hat{\mathbf{x}}_M$  and  $\hat{\mathbf{x}}_{M+1}$  are in the fictitious domain  $\hat{\Omega} \setminus \bar{\Omega}$ . For the nodes inside the physical domain, the finite element solution  $\mathbf{u}_1$  as well as the integral equation solution  $\mathbf{u}_2$  are available and their superposition yields the solution  $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2$ . In contrast, for the nodes in the fictitious domain only the finite element solution is available, but not the integral equation solution. In our implementation the velocities  $\hat{\mathbf{u}}_2$  for the nodes outside physical domain  $\Omega$  are determined by constant extrapolation from the closest point on the domain boundary (indicated by the dotted lines in figure 5). The function value at this point is given by evaluating the boundary element approximation (15). In order to obtain the closest point, the closest point algorithm of Mauch [20] is used, which delivers the necessary quantities with optimal computational complexity.

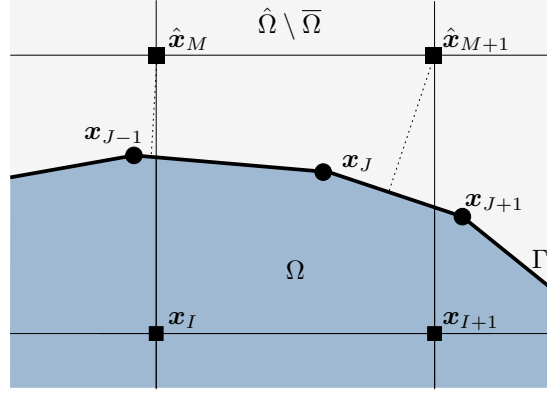


Figure 5. Quadrilateral finite element mesh traversed by a surface mesh

### 3.5. Evaluation of the boundary defect

Another remark concerns the computation of the boundary defect  $\mathbf{h}$ . For flow problems, the finite element velocities  $\hat{\mathbf{u}}_1$  are interpolated with piecewise bi-quadratic polynomials and the defect  $\mathbf{h}$  in the velocities is interpolated with piecewise linear functions. For such a combination, the interpolation of the nodal defect values using the finite element shape functions with

$$\mathbf{h}_K = \mathbf{g}(\mathbf{x}_K) - \hat{\mathbf{u}}_1^h(\mathbf{x}_K) = \mathbf{g}(\mathbf{x}_K) - \sum_I \varphi_I(\mathbf{x}_K) \mathbf{u}_{1,I} \quad (37)$$

does not lead to satisfactory results. This equation contains the difference between a linear polynomial (approximation of  $\mathbf{g}$ ) and a polynomial with up to fourth order terms. In order to mitigate this mismatch, the interpolation shape functions  $\varphi_I$  in equation (37) are replaced with piecewise linear shape functions.

### 3.6. Fluid traction

In some applications, in addition to the velocity-pressure pair  $(\mathbf{u}, p)$  also the traction field  $\mathbf{t} = \boldsymbol{\sigma}(\mathbf{u}, p) \cdot \mathbf{n}$  on the surface  $\Gamma$  needs to be computed. Since the integral equation solution pair  $(\mathbf{u}_2, p_2)$  is not available outside the physical domain  $\Omega$ , the contribution of the integral equation solution is computed using

$$\mathbf{t}_2(\mathbf{x}) = (\mathbf{C}\mathbf{q})(\mathbf{x}) + \int_{\Gamma} \mathbf{T}^*(\mathbf{x} - \mathbf{y}) \mathbf{q}(\mathbf{y}) \, ds_{\mathbf{y}}, \quad (38)$$

which is commonly referred to as the adjoint double layer potential. This expression is obtained by inserting the single layer potentials for the velocity (12) and pressure (34) into the definition of the fluid stress tensor (22), moving the evaluation point to the surface and multiplying the result with the surface normal vector  $\mathbf{n}(\mathbf{x})$ . This yields the new integral kernel  $\mathbf{T}^*$  and the jump term  $\mathbf{C}\mathbf{q}$  due to the limiting process [10, 16]. If the evaluation point  $\mathbf{x}$  is positioned on a smooth part of the boundary  $\Gamma$ , the jump term is simply  $(\mathbf{C}\mathbf{q})(\mathbf{x}) = \frac{1}{2}\mathbf{q}(\mathbf{x})$ . To ensure the sufficient smoothness of the adjoint double layer potential (38), it is evaluated inside the boundary elements and not at their vertices. Note that for the applications presented in this work, the integral kernel  $\mathbf{T}^*$  remains weakly singular, whereas, for instance, in the case of linear elasticity the integration in (38) has to be carried out in the sense of a Cauchy principal value (see, e.g. [16]). An explicit expression of  $\mathbf{T}^*$  can be found in [25].

### 3.7. Examples

**3.7.1. Circular Couette flow.** As a first example for the Navier-Stokes equations, the circular Couette flow is considered (see, e.g., [26]). This flow occurs between two coaxial cylindrical surfaces  $\Gamma_1$  and  $\Gamma_2$  with radii  $r_1$  and  $r_2$  which rotate with constant angular velocities  $\omega_1$  and  $\omega_2$ , respectively. The stationary flow pattern is axisymmetric and, hence, is best described in polar coordinates  $(r, \phi)$ . The analytical solution for the velocity and pressure is

$$u_r = 0, \quad u_{\phi}(r) = Ar + \frac{B}{r}, \quad p(r) = \frac{A^2}{2}r^2 + 2AB \log r - \frac{B^2}{2r^2} + C \quad (39)$$

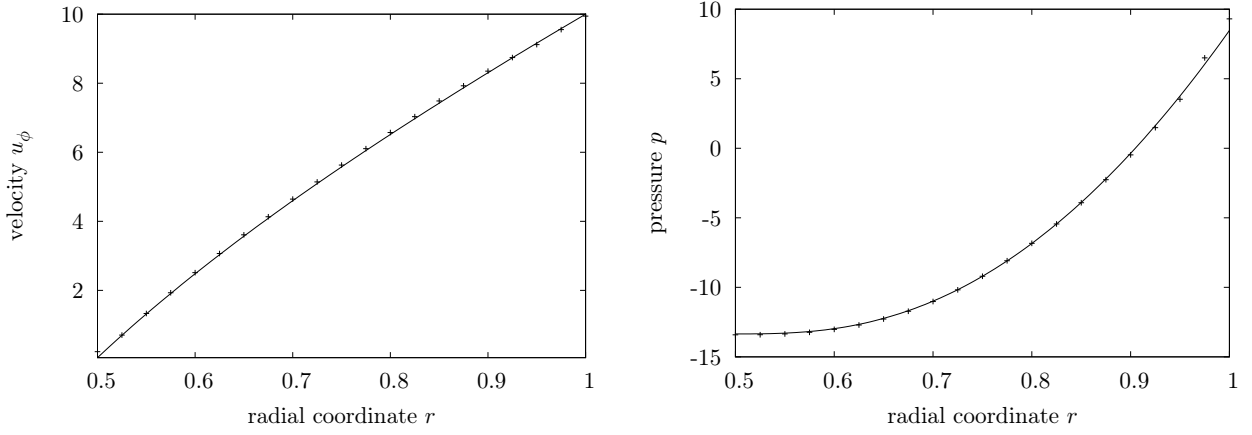


Figure 6. Tangential velocity  $u_\phi$  (left) and the pressure  $p$  (right) across the radius; the solid lines represent the analytical solution according to equation (39) and the crosses the computed values

with

$$A = \frac{\omega_1 r_1^2 - \omega_2 r_2^2}{r_1^2 - r_2^2}, \quad B = \frac{(\omega_1 - \omega_2) r_1^2 r_2^2}{r_2^2 - r_1^2}. \quad (40)$$

The pressure is defined up to an arbitrary constant  $C$ . The torque on the cylinders is  $T = \pm 4\pi\mu B$ , where the plus sign refers to the outer and the minus sign to the inner cylinder. For the numerical computations, a physical domain  $\Omega$  with an internal radius  $r_1 = 0.5$  and an external radius  $r_2 = 1$  is chosen. The internal cylinder rotates with the angular velocity  $\omega_1 = 0.1$  and the external with  $\omega_2 = 10$ , which give, according to equation (40),  $A = 13.3$  and  $B = -3.3$ . The Cartesian domain is a square with  $\hat{\Omega} = (-1.2, 1.2) \times (-1.2, 1.2)$  and is discretized with 7569  $Q_2$ - $Q_1$  rectangles. The surfaces of the cylinders are discretized with 320 (outer cylinder) and 160 (inner cylinder) line elements with a piecewise constant approximation for the surface density  $\mathbf{q}$  and continuous piecewise linear interpolation of the Dirichlet datum  $\mathbf{h}$ . In the numerical computations the viscosity of the fluid is chosen as  $\mu = 0.1$  and its density as  $\rho = 1$ . Note that the analytical solution of the circular Couette flow is independent of the viscosity and density.

In Figure 6, the analytical and numerical solutions of the tangential velocity  $u_\phi$  and the pressure  $p$  across the radius are compared. The very good agreement between the analytical and numerical solutions is evident. A closer look reveals only a slight deviation of the pressure solution towards the boundaries. The constant  $C$  in equation (39) is adjusted such that the numerical and analytical solutions agree at the midpoint, i.e.,  $p^h(r_m) = p(r_m)$  for  $r_m = (r_1 + r_2)/2$ .

In addition, the torque  $T$  acting on the two cylinders is computed with

$$T_i = \int_{\Gamma_i} (\mathbf{x} - \mathbf{x}_0) \times [\boldsymbol{\sigma}(\mathbf{u}, p) \cdot \mathbf{n}] \, ds_x \quad (41)$$

where the origin  $\mathbf{x}_0$  is chosen as the centre of the rotation and the index  $i = 1, 2$  refers to the outer or inner surface, respectively. The vector product ' $\times$ ' in equation (41) is to be understood in the usual vectorial sense with the resulting torque vector pointing in the third coordinate direction. The temporal change of the torque magnitudes is given in figure 7, where one can clearly observe how they converge in the course of time. The exact stationary solution is given as the solid line in the same figure. The relative numerical error in the computed torques is given in table I. Here, the numerical solution of the steady problem is provided in addition to the unsteady solution. In both cases, the results for the outer cylinder are less accurate than the results for the inner cylinder, which can be attributed to the relative coarseness of the finite element mesh for the outer cylinder in comparison to the inner cylinder.

**3.7.2. Driven cavity.** The next verification example is the driven cavity which has been extensively used as a test case in literature. The physical fluid domain is a unit square  $\Omega = (-0.5, 0.5) \times (-0.5, 0.5)$  with a prescribed unit velocity along its top boundary and no-slip boundary along the remaining boundaries. The fluid has a density of  $\rho = 1$  so that the Reynolds number is controlled by  $Re = 1/\mu$ . The Cartesian domain is a square  $\hat{\Omega} = (-0.6, 0.6) \times (-0.6, 0.6)$ , which is discretized with  $(N_{FE} - 1)^2/4$  finite elements with  $Q_2$ - $Q_1$  interpolation. The surface discretization

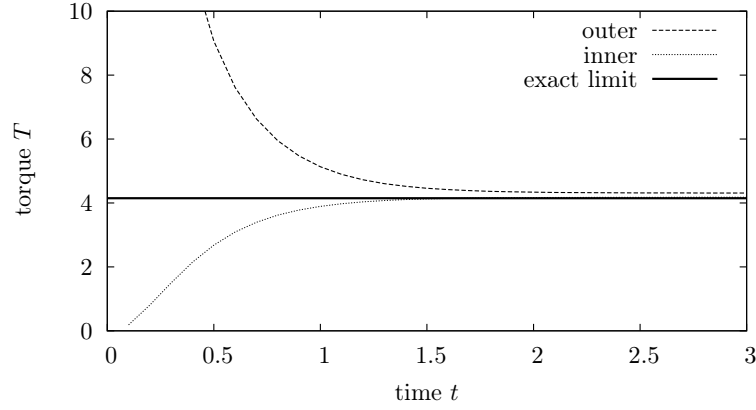


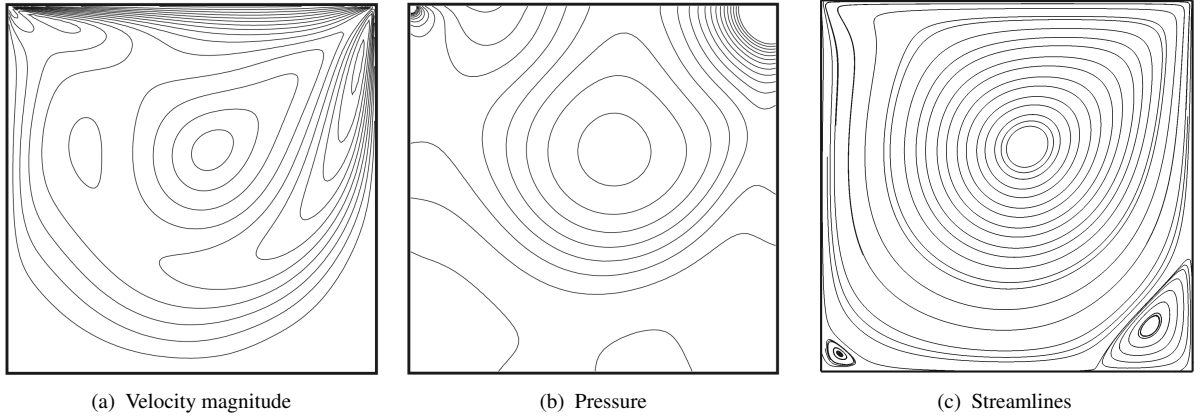
Figure 7. Magnitude of the torque against time for the outer and inner cylinder and the exact stationary solution

	steady	unsteady	exact
inner	4.158 (0.27%)	4.179 (0.74%)	4.147
outer	4.264 (2.8%)	4.311 (4.0%)	4.147

Table I. Magnitudes and relative errors of the torques on the cylinders

consists of  $4N_{IE}$  line elements with constant shape functions. In all computations the convergence criterion is  $|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}| < 10^{-5}$ .

In figure 8, the contour lines for the velocity magnitude and the pressure are plotted for  $Re = 400$ . Additionally, the streamlines computed from the velocity field are also given in figure 8. This flow field is obtained by a steady state analysis with  $N_{FE} = 161$  and  $N_{IE} = 160$ , i.e., 6400 quadrilateral finite elements and 640 line elements. The key features of the driven cavity flow at  $Re = 400$  are a core vortex and two smaller vortices at the bottom corners, which are all well reproduced by the computation.

Figure 8. Contour lines of velocity magnitude  $|u|$ , pressure  $p$  and streamlines for  $Re = 400$ 

This example is subjected to a more systematic mesh convergence study using the reference solutions of Ghia et al. [27] for Reynolds numbers 100, 400 and 1000. In a first set of computations, the mesh parameters  $N_{FE}$  and  $N_{IE}$  are simultaneously varied. Figure 9 shows the  $u_1$  component of the computed velocities along the  $x_2$ -axis and their  $u_2$  component along the  $x_1$ -axis for the considered three Reynolds numbers and three different mesh sizes together with the reference solutions. The plotted data were obtained with a steady state analysis. In each case, there is a good

agreement between the reference and numerical solutions and in most cases already relatively coarse meshes give very good results. Note that for higher Reynolds numbers, the used element sizes are always a step smaller in line with higher resolution requirements.

In a second set of computations, the mesh parameters  $N_{FE}$  and  $N_{IE}$  are independently varied for the problem with Reynolds number 400. In figure 10, the finite element mesh stays the same with  $N_{FE} = 41$  and for the surface mesh  $N_{IE}$  is 20, 40 or 80. Figure 11 shows the results for the opposite case, where the surface mesh stays the same with  $N_{IE} = 40$  and for the finite element mesh  $N_{FE}$  is 21, 41 and 81. In the first case, where the surface mesh is refined and the finite element mesh size is fixed, the numerical solution converges to the reference result with successive mesh refinement. In the second case, where the finite mesh is refined and the surface mesh size is fixed, the numerical solution slightly deteriorates with successive mesh refinement. From these numerical studies can be concluded that the characteristic element size of the finite element mesh and the surface mesh have to be comparable in order to achieve maximum accuracy.

Next, in order to investigate how the relative placement of the surface mesh with respect to the structured finite element mesh influences the results, the physical domain is rotated by an angle of 30 and 45 degrees. The mesh parameter of the surface mesh is  $N_{IE} = 80$ . In order to accommodate for the rotated surface, the Cartesian domain is enlarged to  $\hat{\Omega} = (-0.8, 0.8) \times (-0.8, 0.8)$ . Its mesh parameter is  $N_{FE} = 107$ , which is comparable to  $N_{FE} = 81$  for the smaller domain used previously. In figure 12, the velocity profiles for  $Re = 400$  are given, which clearly show that the numerical results are only slightly influenced by the domain rotation.

#### 4. FLUID-RIGID BODY INTERACTION

##### 4.1. Problem formulation

A key advantage of the proposed discretization method is that the finite element mesh on the Cartesian domain does not depend on the location of the immersed domain boundaries. This is particularly appealing for applications with moving boundaries. As an example for such an application, the settling of a particle in a closed tank filled with a viscous fluid is considered. Finite element simulations of particle settling with adaptive remeshing have been presented, amongst others, in [28] and [29].

The considered setup is depicted in figure 13, where a circular particle  $P$  is located at the position  $(x_1, x_2) = (X_1, X_2)$  with an orientation angle  $\vartheta$  with respect to the horizontal axis. The tank  $\Omega_T$  with boundary  $\Gamma_T$  is rectangular and has the dimensions  $W \times H$ . The velocity of the particle is  $\mathbf{U} = (U_1, U_2)$  and it rotates around its centre  $\mathbf{X} = (X_1, X_2)$  with the angular velocity  $\omega = \dot{\vartheta}$ . The gravity field  $\mathbf{g} = (0, -g)$  acts vertically downwards. If the mass density of the particle is larger than the fluid density, its weight will accelerate it downwards. At the same time, the fluid in the tank exerts a drag force on the particle which depends on the fluid flow in the tank.

This coupled problem is described by Newton's law for the particle and the Navier-Stokes equations for incompressible viscous fluid flow (21). The boundary conditions of the fluid problem are no-slip conditions at the tank walls and at the particle surface

$$\mathbf{u} = \mathbf{0} \quad \mathbf{x} \in \Gamma_T \quad (42)$$

$$\mathbf{u} = \mathbf{U} + \omega \times (\mathbf{x} - \mathbf{X}) \quad \mathbf{x} \in \partial P(t), \quad (43)$$

wherein the vector product  $\times$  has to be understood in a three-dimensional sense. Equation (43) emphasises that the location of the boundary  $\partial P$ , i.e., the surface of the particle, is a function of time. The fluid problem thus consists of the Navier-Stokes equations (21) with gravity as a body force and equations (42) and (43) as velocity boundary conditions. The fluid is assumed to be initially at rest so that the initial condition is  $\mathbf{u}_0 = \mathbf{0}$ .

The motion of the particle itself is governed by Newton's law, i.e.,

$$m \frac{d\mathbf{U}}{dt} = \mathbf{F}, \quad \frac{d\mathbf{X}}{dt} = \mathbf{U}, \quad (44a)$$

$$J \frac{d\omega}{dt} = T, \quad \frac{d\vartheta}{dt} = \omega. \quad (44b)$$

The first equation (44a) describes the translational motion and  $m$  denotes the particle mass. The rotational motion is described by equation (44b), where  $J$  refers to the polar moment of inertia.

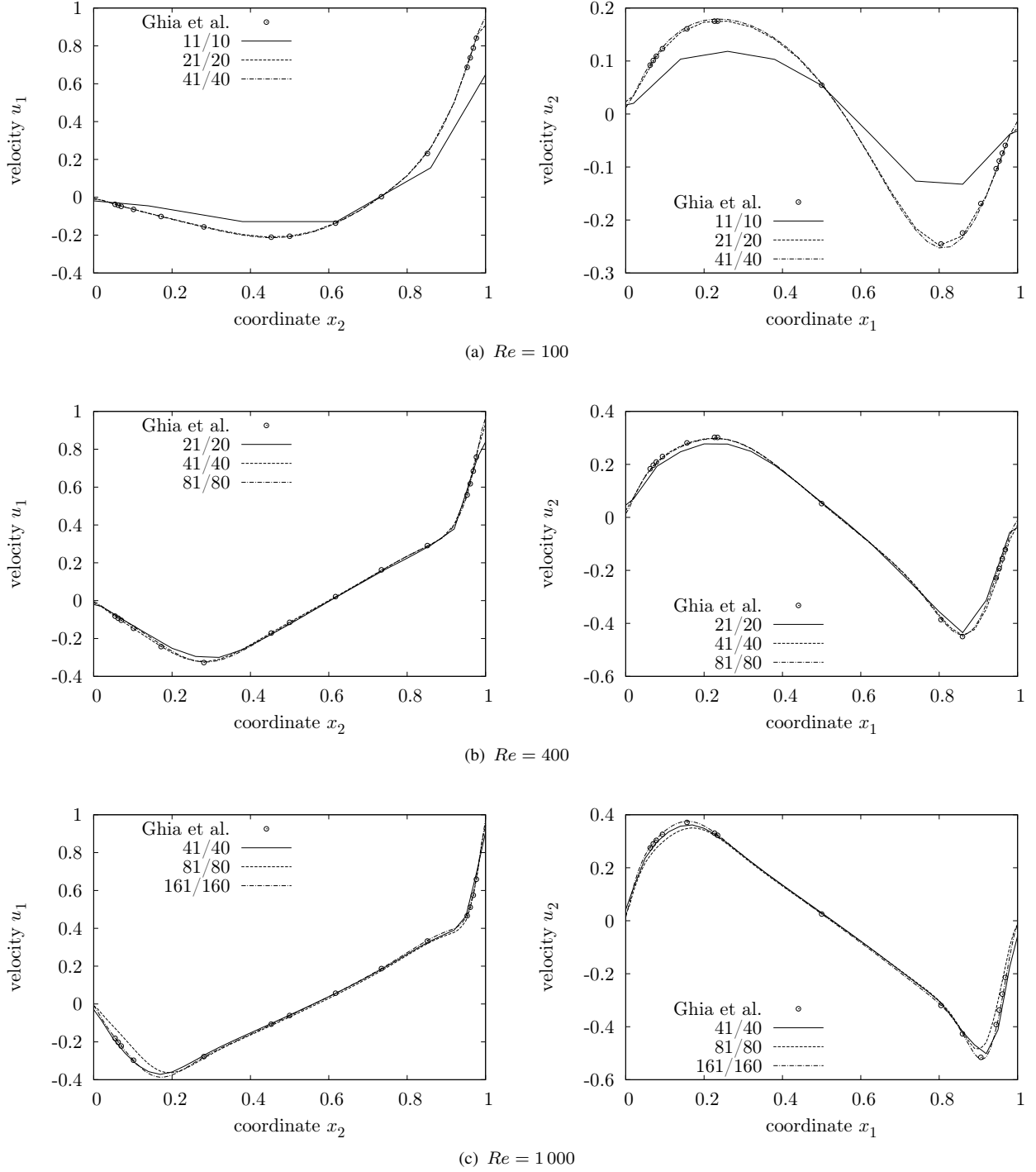


Figure 9. Velocity profiles of the driven cavity for Reynolds numbers 100, 400 and 1000; the number pairs  $N_{FE}/N_{IE}$  in the legends refer to the mesh parameters (see text)



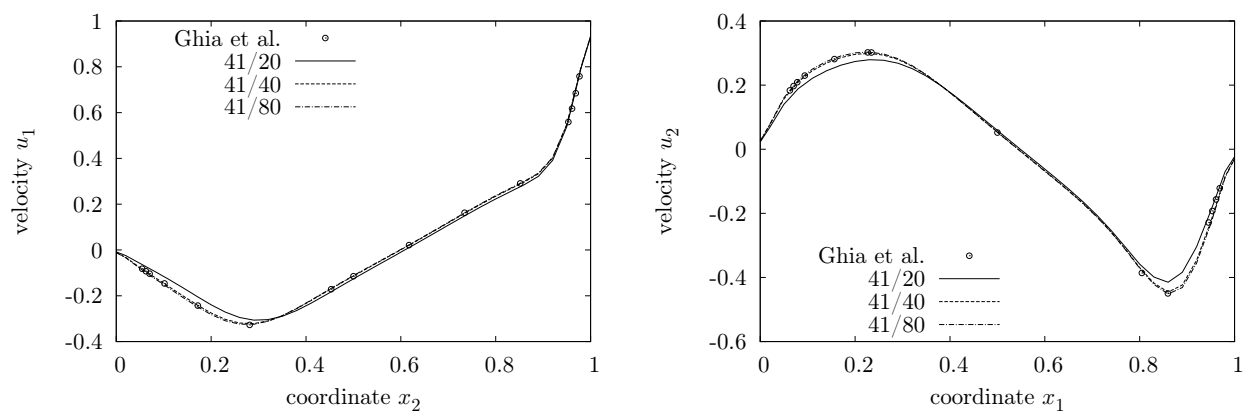


Figure 10. Velocity profiles of the driven cavity for  $Re = 400$  for fixed finite element mesh and successively refined surface meshes

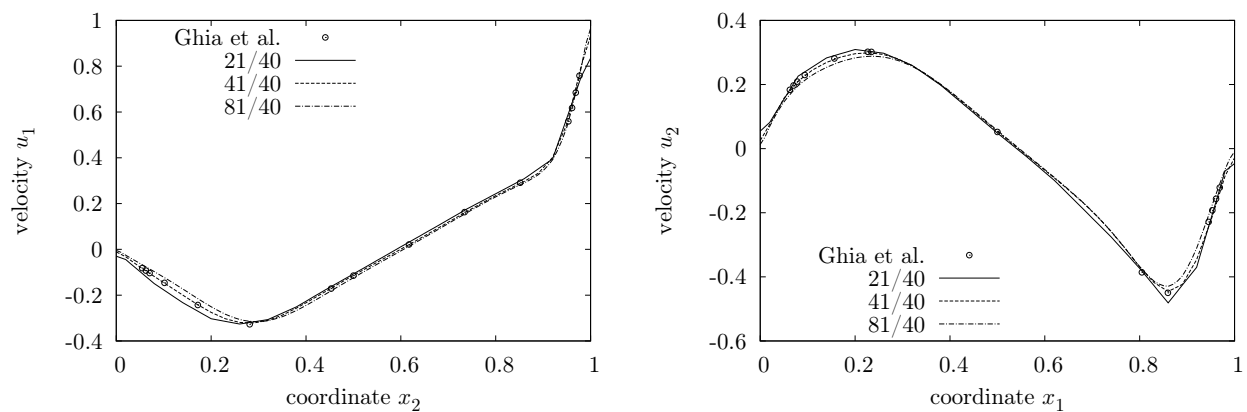


Figure 11. Velocity profiles of the driven cavity for  $Re = 400$  for fixed surface mesh and successively refined finite element meshes

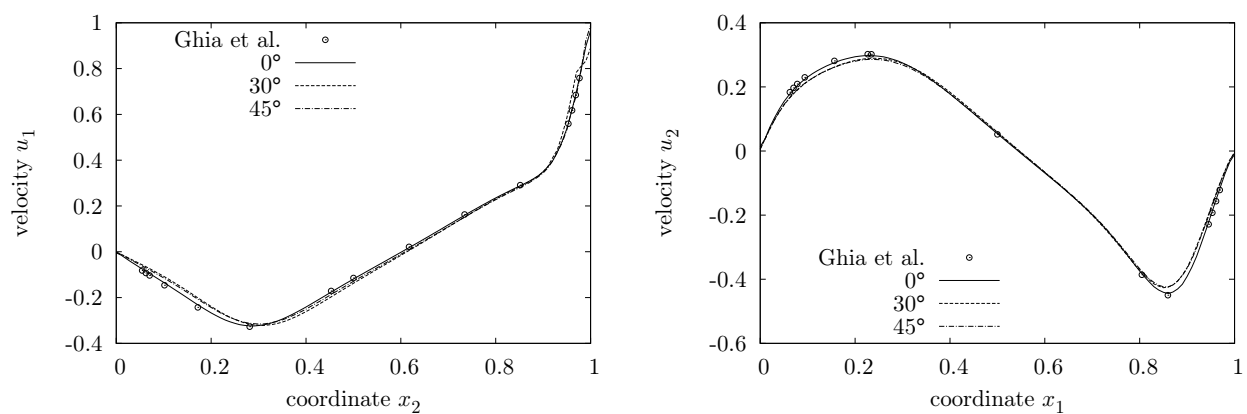


Figure 12. Velocity profiles of the driven cavity for  $Re = 400$  for different rotations of the surface

It remains to specify the force  $\mathbf{F}$  and the torque  $T$  which act on the particle. The force and the torque on the particle with the surface normal  $\mathbf{n}$  are given by

$$\mathbf{F} = m\mathbf{g} - \int_{\partial P(t)} \boldsymbol{\sigma}(\mathbf{u}, p) \cdot \mathbf{n} \, ds \quad \text{and} \quad T = - \int_{\partial P(t)} (\mathbf{x} - \mathbf{X}) \times [\boldsymbol{\sigma}(\mathbf{u}, p) \cdot \mathbf{n}] \, ds. \quad (45)$$

Note that the minus signs appear due to the orientation of the normal vector  $\mathbf{n}$  which points by definition to the inside of the particle (or, outside of the fluid).

#### 4.2. Coupling algorithm

The solution strategy used in this work is the semi-implicit predictor-corrector algorithm introduced by Hu et al. [28]. It is assumed that the solution of the fluid-particle system is known at the current time step  $t = t_n$ . Hence, the quantities  $\mathbf{X}_n, \vartheta_n, \mathbf{U}_n, \omega_n$  (i.e., the state of the particle) and  $\mathbf{u}_n, p_n$  (i.e., the state of the fluid) are known. Then the implicit iteration loop within each time step begins with the following predictor values for the particle force and torque

$$\mathbf{F}^{(0)} = \mathbf{F}(\mathbf{u}_n, p_n) \quad \text{and} \quad T^{(0)} = T(\mathbf{u}_n, p_n) \quad (46)$$

The subsequent nonlinear iterations consist of the following substeps:

1. Compute particle velocities  $\mathbf{U}^{(k+1)} = \mathbf{U}_n + \alpha(\Delta t/m)\mathbf{F}^{(k)}$  and  $\omega^{(k+1)} = \omega_n + \alpha(\Delta t/J)T^{(k)}$ .
2. Solve the fluid problem for velocity  $\mathbf{u}^{(k+1)}$  and pressure  $p^{(k+1)}$  with boundary conditions according to (43) using the immersed finite element method described in section 3.
3. Compute the force  $\mathbf{F}^{(k+1)}$  and torque  $T^{(k+1)}$  with equation (45) based on the current fluid state  $(\mathbf{u}^{(k+1)}, p^{(k+1)})$ .
4. Iterate  $k \leftarrow k + 1$ .

Once this nonlinear iteration is converged, the new particle velocities are assigned, i.e.  $\mathbf{U}_{n+1} = \mathbf{U}^{(k+1)}$  and  $\omega_{n+1} = \omega^{(k+1)}$ , and the new particle position and orientation is computed with a forward Euler step

$$\mathbf{X}_{n+1} = \mathbf{X}_n + \Delta t \mathbf{U}_{n+1} \quad \text{and} \quad \vartheta_{n+1} = \vartheta + \Delta t \omega_{n+1}. \quad (47)$$

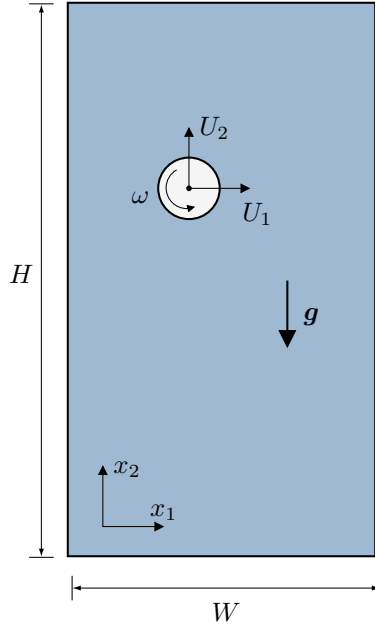


Figure 13. Circular particle in a closed tank

As realised by Hu et al. [28], an under-relaxation improves the convergence of the coupling algorithm. Therefore, the relaxation parameter  $\alpha$  is introduced in step 1, which is chosen with  $\alpha = 0.4$  as suggested in [28]. Note that step 2 of the coupling algorithm still involves the solution of a fully nonlinear problem which requires sub-iterations. In the computations usually only two to three such sub-iterations are sufficient.

#### 4.3. Example

Two sets of computations using the above described coupling algorithm are carried out. In the first set of computations, the particle is released from the symmetry axis of the tank. In the second set, the particle is released from an off-centre position. All other parameters are identical for both computations. The closed tank occupies the region  $\Omega_T = (-0.05, 0.05) \times (0, 0.5)$  and the particle is of circular shape with radius  $r_p = 0.01$ . The fluid has unit density  $\rho = 1$  and the particle is twice as heavy with  $\rho_p = 2$ . The gravitation constant is chosen simply as  $g = 10$ . The viscosity of the fluid is chosen as  $\mu = 0.001$ . Based on the terminal settling velocity of about  $|U_2| = 0.155$  (see figure 15), the particle Reynolds number is  $Re = 2r_p U_2 / \mu \approx 3.1$ , which is well below the values for which vortex shedding can be expected.

The Cartesian domain is chosen as  $\hat{\Omega} = (-0.06, 0.06) \times (-0.01, 0.51)$  and is discretized with  $45 \times 197 = 8865$   $Q_2$ - $Q_1$  finite elements. The boundary surfaces of the particle  $\partial P$  and the tank  $\Gamma_T$  are discretized with 80 and 240 line elements, respectively, using constant shape functions for the surface density. The temporal discretization begins with a time step size of  $\Delta t = 0.001$  which is increased for the first ten time steps by ten percent in each step yielding a final step size of approximately  $\Delta t = 0.0026$ . Since in the computation the particle is suddenly equipped with its excess weight, numerical instabilities are likely to occur at the beginning of the computation. Therefore, it is reasonable to begin the computation with a smaller time step size. Numerical instabilities also occur when the particle gets close to the bottom wall. This effect is a lot more pronounced for the particle which was released from an eccentric position. For this reason, the computation was stopped after 800 time steps at approximately  $t = 2.0$  or  $X_2 = 0.18$ .

In the first computation the particle is released at position  $\mathbf{X} = (0, 0.475)$  and in the second computation at  $\mathbf{X} = (-0.03, 0.475)$ . In figure 14 the particle paths for both computations are shown with the tank being rotated by 90 degrees. Whereas the centred particle simply falls straight down due to its own weight, the initially eccentric particle also moves horizontally. A closer look at its motion at the beginning reveals that it moves slightly towards the left wall before reversing the direction and heading for the centre line  $X_1 = 0$ . It overshoots a little and then returns to the centre.

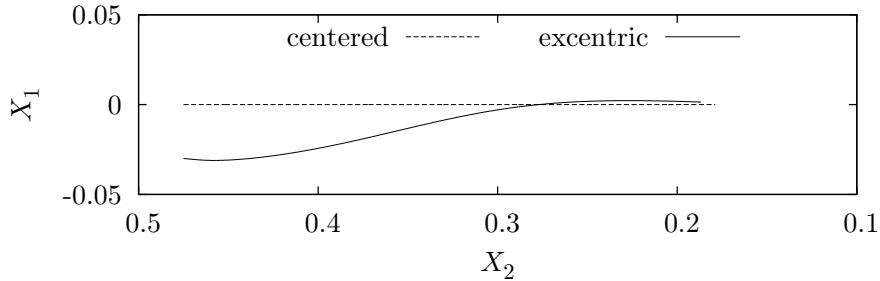


Figure 14. Particle paths (rotated by 90 degrees)

Figure 15 shows the vertical particle position, its settling velocity and the force acting on it against time. Clearly, in both cases the particle reaches the same terminal settling velocity of approximately  $U_2 \approx -0.155$ . This steady state situation corresponds to an equilibrium of the particle excess weight of  $W_p = \pi r_p^2 (\rho_p - \rho) g \approx 3.1416 \cdot 10^{-3}$  and the drag force  $F_D$  on the particle. The particle excess weight is indicated with a dotted horizontal line in figure 15(c), which serves as an asymptote for the drag force in both cases. The pictures in figure 15 also indicate that the vertical motion of the particle with the initially eccentric position deviates noticeably from the particle which is initially centred in the acceleration phase  $t < 0.8$ . Its downward acceleration is clearly inhibited by its vicinity to the wall and it lags behind in terms of its vertical position.

The sequence of snapshots in figure 16 show the streamlines and pressure contours in the fluid at time instants 0.25, 0.5, 1.0, 1.5 and 2.0 for the eccentric particle. Similarly, figure 17 shows the vorticity isolines and the velocity

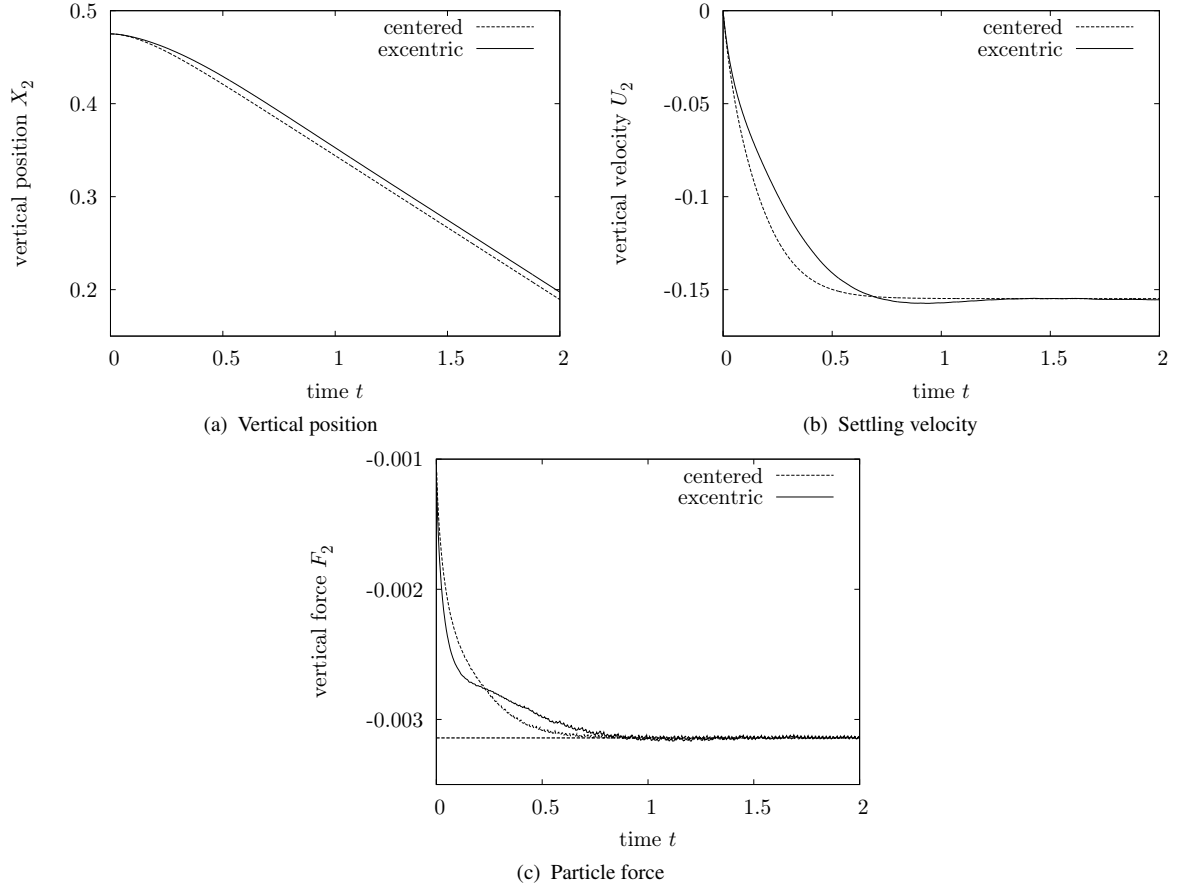


Figure 15. Vertical position and velocity of the particle and fluid force acting on the particle

magnitude contours at the same time instants.

## 5. CONCLUSION

We have presented a new immersed finite element method for computing problems with complex, time-dependent domains. In common with other immersed discretization methods only a surface mesh of the domain is required. The discretization of the embedding Cartesian domain relies on a non-body-conforming structured mesh, which can be easily generated without resorting to specialised mesh generation algorithms. A unique feature of the proposed method is that the solution of the original problem is computed as the superposition of a finite element and an integral equation solution. As a result, the conditioning of the discretized equations does not depend on the shape and number of cut-elements (i.e., finite elements traversed by the boundary mesh).

In the finite element solution on the Cartesian domain, the consideration of the physical domain leads only to a modified forcing term. This is relevant for using any pre-existing fast solver packages for finite elements, since only suitably chosen boundary forces need to be prescribed. Furthermore, the structured mesh on the Cartesian domain facilitates the use of solvers based on Fourier transforms or multigrid methods (see [13]). There are also well-established fast integral equation solvers that can be used for solving the integral equations, which appear in the proposed method (see, e.g., [30]). In this regard, it is important that the integral equations in the proposed method correspond to homogeneous boundary value problems so that no costly Newton potentials need to be evaluated.

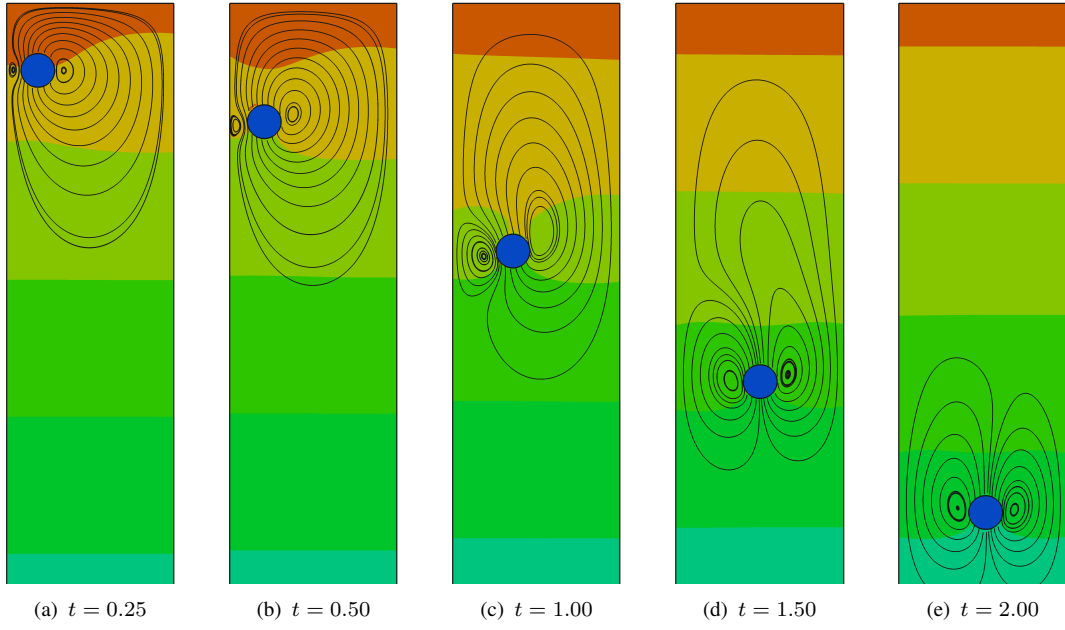


Figure 16. Streamlines and pressure contours at different time instants (excentric settling, bottom of tank is truncated)

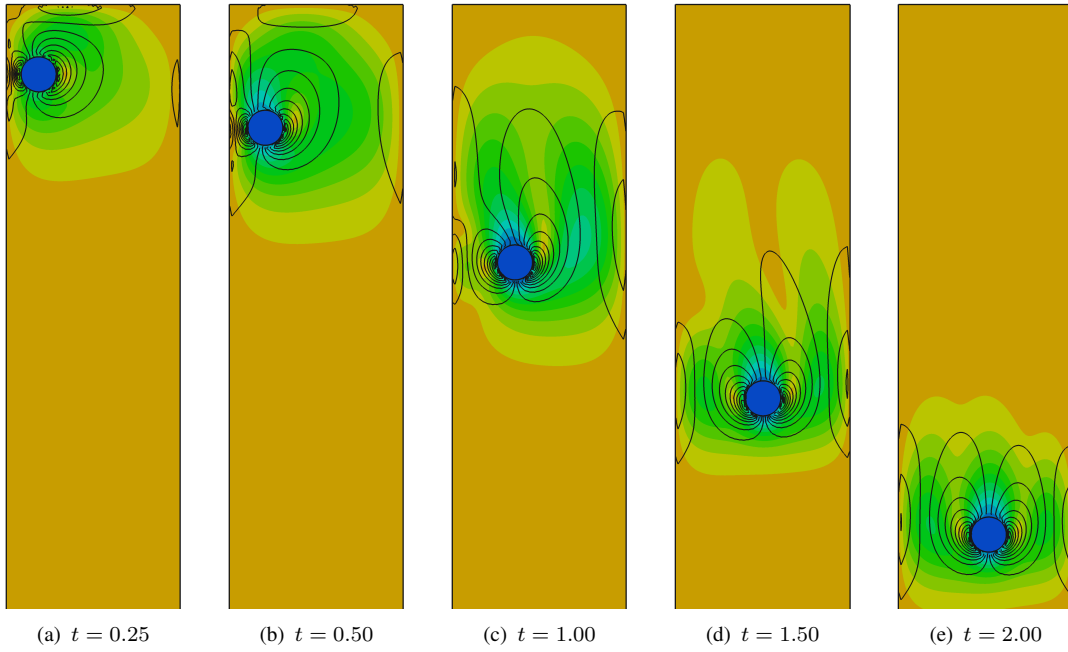


Figure 17. Vorticity isolines and velocity magnitude contours at different time instants (excentric settling, bottom of tank is truncated)

## ACKNOWLEDGEMENT

The support of the EPSRC through grant # EP/G008531/1 is gratefully acknowledged.

## APPENDIX

## FUNDAMENTAL SOLUTIONS

In the following, the two-dimensional fundamental solutions used in this work are given for completeness. In these equations the integration point is denoted with  $\mathbf{y}$  and the collocation or evaluation point with  $\mathbf{x}$  and  $\mathbf{I}$  denotes the  $2 \times 2$ -identity matrix. References for the given fundamental solutions are, amongst others, [25, 10, 16].

*Stokes system.* The fundamental solutions of the Stokes system  $-\mu \nabla^2 \mathbf{u} + \nabla p = \mathbf{f}$ ,  $\nabla \cdot \mathbf{u} = 0$  are of the form

$$\mathbf{U}^*(\mathbf{x} - \mathbf{y}) = \frac{1}{4\pi\mu} \left( -\log |\mathbf{x} - \mathbf{y}| \mathbf{I} + \frac{(\mathbf{x} - \mathbf{y}) \otimes (\mathbf{x} - \mathbf{y})}{|\mathbf{x} - \mathbf{y}|^2} \right) \quad (48)$$

$$\mathbf{P}^*(\mathbf{x} - \mathbf{y}) = -\frac{1}{2\pi} \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|^2}. \quad (49)$$

*Brinkman system.* The Brinkman or modified Stokes system reads  $(\rho/\Delta t) - \mu \nabla^2 \mathbf{u} + \nabla p = \mathbf{f}$ ,  $\nabla \cdot \mathbf{u} = 0$ . With the abbreviation  $z = |\mathbf{x} - \mathbf{y}| \sqrt{\rho/(\mu \Delta t)}$ , the fundamental solutions are

$$\begin{aligned} \mathbf{U}^*(\mathbf{x} - \mathbf{y}) = \frac{1}{2\pi\mu z^2} & \left[ (z K_1(z) + z^2 K_0(z) - 1) \mathbf{I} \right. \\ & \left. + (2 - 2z K_1(z) - z^2 K_0(z)) \frac{(\mathbf{x} - \mathbf{y}) \otimes (\mathbf{x} - \mathbf{y})}{|\mathbf{x} - \mathbf{y}|^2} \right] \end{aligned} \quad (50)$$

$$\mathbf{P}^*(\mathbf{x} - \mathbf{y}) = -\frac{1}{2\pi} \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|^2}. \quad (51)$$

where  $K_0(z)$  and  $K_1(z)$  are the modified Bessel functions of the second kind. In the numerical computations  $K_0(z)$  and  $K_1(z)$  are replaced by their asymptotic expansions for very small ( $z < 10^{-6}$ ) or very large ( $z > 500$ ) arguments.

## REFERENCES

1. Glowinski R, Pan TW, Periaux J. A fictitious domain method for Dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering* 1994; **111**:283–303.
2. Peskin C. The immersed boundary method. *Acta Numerica* 2002; **11**:1–39.
3. Fedkiw R, T A, Merriman B, Osher S. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics* 1999; **152**:457–492.
4. Mittal R, Iaccarino G. Immersed boundary methods. *Annual Review of Fluid Mechanics* 2005; **37**:239–261.
5. Sukumar N, Chopp D, Moes N, Belytschko T. Modeling holes and inclusions by level sets in the extended finite element method. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:6183–6200.
6. Belytschko T, Parimi C, Moes N, Sukumar N, Usui S. Structured extended finite element methods for solids defined by implicit surfaces. *International Journal for Numerical Methods in Engineering* 2003; **56**:609–635.
7. 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.
8. Banerjee P, Butterfield R. *Boundary element methods in engineering science*. McGraw-Hill, 1981.
9. Aliabadi M. *The boundary element method; applications in solids and structures*, vol. 2. John Wiley & Sons Ltd., 2002.
10. Steinbach O. *Numerical Approximation Methods for Elliptic Boundary Value Problems*. Springer, 2008.
11. Jung M, Steinbach O. A finite element-boundary element algorithm for inhomogeneous boundary value problems. *Computing* 2002; **68**:1–17.
12. Mayo A. The fast solution of Poisson’s and the biharmonic equations on irregular regions. *SIAM Journal on Numerical Analysis* 1984; **21**:285–299.
13. Biros G, Ying L, Zorin D. A fast solver for the Stokes equations with distributed forces in complex geometries. *Journal of Computational Physics* 2004; **193**:317–348.
14. Tornberg AK, Shelley MJ. Simulating the dynamics and interactions of flexible fibers in Stokes flows. *Journal of Computational Physics* 2004; **196**:8–40.
15. Veerapaneni S, Gueyffier D, Zorin D, Biros G. A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2D. *Journal of Computational Physics* 2009; **228**:2334–2353.
16. Hsiao G, Wendland W. *Boundary Integral Equations*. Springer, 2008.

17. Johnston P. Semi-sigmoidal transformations for evaluating weakly singular boundary element integrals. *International Journal for Numerical Methods in Engineering* 2000; **47**:1709–1730.
18. Rüberg T, Schanz M. An alternative collocation boundary element method for static and dynamic problems. *Computational Mechanics* 2009; **44**:247–261.
19. Cirak F, Deiterding R, Mauch S. Large-scale fluid-structure interaction simulation of viscoplastic and fracturing thin-shells subjected to shocks and detonations. *Computers & Structures* 2007; **85**:1049–1065.
20. Mauch S. A fast algorithm for computing the closest point and distance transform. *Technical Report*, California Institute of Technology 2000.
21. Gresho P, Sani R. *Incompressible Flow and the Finite Element Method*. 1998, John Wiley & Sons.
22. Ern A, Guermond JL. *Theory and Practice of Finite Elements*. Springer, 2003.
23. Reidinger B, Steinbach O. A symmetric boundary element method for the Stokes problem in multiple connected domains. *Mathematical Methods in the Applied Sciences* 2002; **26**:77–93.
24. Golub G, van Loan C. *Matrix Computations*. Johns Hopkins University Press, 1996.
25. Pozrikidis C. *A practical guide to Boundary Element Methods with the Software Library BEMLIB*. Chapman & Hall/CRC, 2002.
26. Batchelor G. *An Introduction to Fluid Dynamics*. Cambridge University Press, 1967.
27. Ghia U, Ghia K, Shin C. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**:387–411.
28. Hu H, Joseph D, Crochet M. Direct simulation of fluid particle motions. *Theoretical and Computational Fluid Dynamics* 1992; **3**:285–306.
29. Saksono P, Dettmer W, Perić D. An adaptive remeshing strategy for flows with moving boundaries and fluid-structure interaction. *International Journal for Numerical Methods in Engineering* 2007; **71**:1009–1050.
30. Bebendorf M. *Hierarchical Matrices — A Means to Efficiently Solve Elliptic Boundary Value Problems*. Springer, 2008.