Computers and Fluids 153 (2017) 118–129

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

An incompressible immersed boundary solver for moving body flows using a cut cell discontinuous Galerkin method

Dennis Krause[a,b,*], Florian Kummer[a]

[a] Chair of Fluid Dynamics, TU Darmstadt, Otto-Berndt-Str. 2, Darmstadt D-64287, Germany
[b] Graduate School Computational Engineering, Dolivostr. 15, Darmstadt D-64293, Germany

ARTICLE INFO

Article history:
Received 21 November 2016
Revised 3 May 2017
Accepted 12 May 2017
Available online 15 May 2017

Keywords:
Discontinuous Galerkin
Immersed boundary method
Moving body flow
Extended/unfitted DG
Incompressible Navier-Stokes equations
Fluid-structure interaction

ABSTRACT

We present a higher-order immersed boundary method (IBM) for moving body flows, based on a discontinuous Galerkin (DG) discretization. In our method bodies are represented by a sharp interface approach using cut cells. The position of the bodies and their velocity is prescribed by Newtons equation of motion. Furthermore, we introduce a splitting type approach of coupling fluid and rigid body solver and point out the explicit calculation of hydrodynamic forces using hierarchical moment fitting. To verify our method we compare our new approach to a range of numerical benchmarks taken from the literature. At first we compare our method with calculations using a body fitted grid. Thereafter we will focus on representing boundaries with fixed motion to point out the accuracy and flexibility of our IBM. Afterwards we show results of full coupled test cases in which the explicit coupling approach and the calculation of forces mentioned above come into play. It can be shown, that with our method we can get similar results by decreasing the number of total unknowns significantly, which is the main advantage of the proposed scheme.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

Particulate flows with lots of moving bodies are of high scientific and technological interest with numerous applications. The understanding of such flows is important in chemical engineering (separation), life science (blood flow) and basic understanding of nature (sedimentation in ocean or river beds). On both, the macro- and microscopic scale, the phenomena of particulate flows are not fully understood and are still under ongoing research in research facilities and industry. Furthermore, there is still a lack of accurate and efficient solvers to tackle these type of flows numerically. Even with the computer power nowadays available on big research and industry clusters, the efficiency of such solvers is still a bottle-neck especially for industrial applications. This is due to the complexity of such particulate flows, meaning both, particle position and velocity are unknown.

Methods for particulate flows can be separated in two general groups: The first one is the so called Lagrangian approach which uses a mesh fitted to the particle surfaces. As the mesh can move arbitrary in the fluid, those methods are called Arbitrary Lagrangian Eulerian (ALE). The ALE method was used for particulate flows by Hu et al. [1,2] and Maury [3,4]. The second group are immersed boundary methods. The first immersed boundary method was proposed by Peskin [5] in the field of fluid-structure interaction for simulating flow patterns around heart valves. The new feature of this method was, that all calculations were done on a fixed Cartesian grid. It was not needed to remesh and project the solution onto the new grid in every timestep in order to be conform with the geometry. The success of his method was to impose the influence of the immersed boundary on the flow without remeshing. In the following years, various modifications on this method have been proposed and an overview can be extracted from Mittal and Iccarino [6]. Immersed boundary methods are commonly used for fluid-structure interaction (FSI) problems including particulate flows. A broad review on immersed boundary methods (IBM) for FSI can be found in the work of [7].

In context of particulate flows, IBM can be further differentiated by the coupling between fluid and particle interactions. The two variants are implicit and explicit coupling schemes. In the implicit coupling approach the forcing is incorporated into the flow equations before discretization. Important work in this field was done by Glowinski (e.g [8,9].) using body-force distributed lagrange multipliers and Patankar [10] using a stress distributed Lagrange multiplier ansatz to model the coupling forces. In the second approach the forcing is introduced after discretization.

[*] Corresponding author at: Chair of Fluid Dynamics, TU Darmstadt, Otto-Berndt-Str. 2, Darmstadt D-64287, Germany.
E-mail address: krause@fdy.tu-darmstadt.de (D. Krause).

http://dx.doi.org/10.1016/j.compfluid.2017.05.008
0045-7930/© 2017 Elsevier Ltd. All rights reserved.

First, the IBM was extended to Stokes flow around suspended particles [11] and Navier-Stokes flow around fixed cylinders [12]. The number of particles was increased by the scheme proposed by Höfer and Schwarz [13]. Further work has been done by Wang and Turek [14] who proposed a multigrid fictitious domain method. To track the particles a volume based integral function is used, which was first proposed by Duchanoy and Jongen [15]. In this context, an alternative scheme was proposed by Uhlmann [16], who combines Peskins regularized delta function approach [17] with direct forcing in a finite-difference context. The immersed boundary method was also used in a lattice-Boltzmann context, e.g. by Feng and Michaelides [18].

An important issue which should be mentioned in terms of immersed boundary methods with moving domains is the occurrence of spurious pressure oscillations. These oscillations are present in many different immersed boundaries methods [16,19–22]. Seo and Mittal [23] found out that the reason for those oscillations lie in a violation of the conservation law due to the appearance and disappearance of cells at the interface. They suggest to apply a cut cell approach with virtual cell merging to enforce mass conservation. Further attempts to eliminate pressure oscillations have been made by [19–21].

For decades, finite-volume (FV) methods have dominated the computational fluid dynamics community not only for single phase problems but for multi-phase problems like water-air interaction and particulate flows. In contrast, Discontinuous Galerkin (DG) methods, first proposed by Reed and Hill [24], became popular because of their ability to use higher-order ansatz spaces, like finite element (FE) methods, but still preserve conservation properties by definition, like FV. DG methods also have several additional advantages, e.g. it is easy to handle hanging nodes and local refinements because of their discontinuous ansatz spaces at cell boundaries.

The work in this paper is based on the extended discontinuous Galerkin (XDG) approach by Kummer [25]. Here, a sharp-interface representation is used. In order to treat the problem of high condition numbers for arbitrary small cut cells a cell-agglomeration procedure is employed. Using such a sharp-interface representation shifts the problem of accuracy and efficiency to the quadrature on those cut cells. For this, we use the hierarchical moment fitting strategy (HMF) first proposed by Müller et al. [26] and later extended in the work of Kummer [25]. To the best of the authors knowledge, there is no work using cut cell/extended DG methods with hierarchical moment fitting in connection with immersed boundaries to tackle particulate flow problems. However, extensive work in case of extended discretization methods has been done in context of extended FE methods (XFEM), first introduced by Möes et al. [27] and later used for fluid dynamics by Gross and Reusken [28]. Beside other authors working in the field of XFEM, the first actual cut cell DG method was presented by Bastian and Engwer [29].

This paper is divided as follows: In Section 2 the physical models for fluid and rigid bodies are presented as well as the calculation of the hydrodynamic forces. Section 3 is about the numerical discretization of the problem and the quadrature on cut cells. In Section 4 numerical experiments are presented to proof the accuracy of the solver. At first two testcases testing the immersed boundary method itself are introduced. This is followed by two test calculations where the full coupling is taken into account. We sum up in Section 5 with a conclusion and an outlook to ongoing work and possible extensions of our solver.

2. Governing equations

2.1. Incompressible Navier–Stokes equations

For introducing the immersed boundary method we define the following disjoint partitioning of the computational domain $\Omega \subset \mathbb{R}^2$:

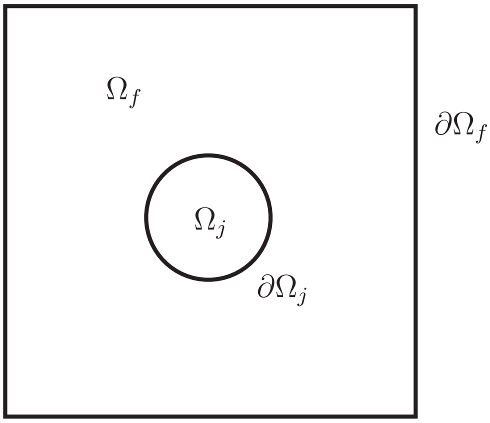


Fig. 1. Computational domain.

$$\Omega = \Omega_f(t) \cup \Omega_j(t) \tag{1}$$

with Dirichlet- and Neumann-type boundaries

$$\Gamma_D \cup \Gamma_N = \partial\Omega_f(t) \backslash \partial\Omega_j(t) \tag{2}$$

and body boundary

$$\Gamma_j(t) = \partial\Omega_j(t). \tag{3}$$

We restrict our setting to be two-dimensional. A schematic figure of the computational domain can be seen in Fig. 1. Therefore physical parameters like densities will be split to ρ_f in Ω_f and ρ_j in Ω_j.

The immersed boundary solver will be used to calculate incompressible flows in interaction with circular shaped bodies. The flow is described by the unsteady Navier–Stokes equations in the fluid region

$$\rho_f\left(\frac{\partial\vec{u}}{\partial t} + \vec{u}\cdot\nabla\vec{u}\right) + \nabla p - \mu_f\Delta\vec{u} = \vec{f} \tag{4a}$$

and the continuity equation

$$\nabla\cdot\vec{u} = 0 \quad \forall\, t \in (0,T) \quad \text{in } \Omega_f(t) \tag{4b}$$

with given initial and boundary conditions

$$\vec{u}(\vec{x},0) = \vec{u}_0(\vec{x}) \quad \forall\, \vec{x} \in \Omega_f(0) \quad \text{with} \quad \nabla\cdot\vec{u}_0 = 0 \tag{4c}$$

$$\vec{u} = \vec{u}_D \text{ on } \Gamma_D,\ (\boldsymbol{I}p - \mu_f\nabla\vec{u})\vec{n}_{\Gamma_N} = 0 \text{ on } \Gamma_N \text{ and } \vec{u} = \vec{u}_j \text{ on } \Gamma_j. \tag{4d}$$

In the equations above $\vec{u}$ is the velocity vector, p the pressure and $\vec{u}_j$ the body velocity. The fluid density is denoted by ρ_f, while $\mu_f = \rho_f \cdot \nu_f$ is the dynamic viscosity of the fluid. Furthermore, volume forces acting on the fluid are described by the force vector $\vec{f}$. Boundary conditions like Dirichlet- and Neumann-types have to be imposed on the outer boundaries $\partial\Omega_f(t) = \Gamma_D \cup \Gamma_N$ of the fluid domain and the presence of the body will be represented in the fluid domain by Dirichlet type boundary conditions for velocity on Γ_j.

2.2. Body motion

The rigid bodies are allowed to translate and rotate freely in the fluid domain. In the following we use capital letters for Lagrangian and small ones for Eulerian quantities. The body movement is described by the Newton-Euler Equations:

$$M_j\frac{d\vec{U}_j}{dt} = M_j\vec{f} + \vec{F}_j \tag{5a}$$

and

$$\boldsymbol{I}_j \frac{d\vec{\omega}_j}{dt} + \vec{\omega}_j \times \boldsymbol{I}_j \vec{\omega}_j = T_j, \tag{5b}$$

where M_j denotes the mass of the body, $\vec{U}_j$ its translational velocity, $\vec{f}$ volume forces like gravity and $\vec{F}_j$ the hydrodynamic forces acting on the body. For the rotational equation $\boldsymbol{I}_j$ is the moment of inertia, $\vec{\omega}_j$ the angular velocity and T_j the hydrodynamic torque.

The position $\vec{X}_j$ of the body and its angle $\vec{\theta}_j$ is obtained by time integration of the kinematic equations

$$\frac{d\vec{X}_j}{dt} = \vec{U}_j \tag{6a}$$

and

$$\frac{d\vec{\theta}_j}{dt} = \vec{\omega}_j. \tag{6b}$$

The hydrodynamic forces and torque are calculated by

$$\vec{F}_j = \int_{\Gamma_j} \boldsymbol{\sigma} \cdot \vec{n} \, \mathrm{d}s, \tag{7a}$$

$$T_j = \int_{\Gamma_j} (\vec{X} - \vec{X}_j) \times (\boldsymbol{\sigma} \cdot \vec{n}) \, \mathrm{d}s. \tag{7b}$$

Eq. (7) contains the surface integration of the total stress tensor $\boldsymbol{\sigma}$ along the body surface, which is defined as follows:

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \mu_f [\nabla\vec{u} + (\nabla\vec{u})^T], \tag{8}$$

where $\boldsymbol{I}$ is the identity tensor. The term $(\vec{X} - \vec{X}_j)$ denotes the distance between a point on the body surface and its center. In particular, for our circular shaped bodies this is just the radius r.

At the interface between fluid and body Γ_j, no-slip boundary conditions are enforced. Hence the body velocity described by

$$\vec{u}_j(\vec{X}) = \vec{U}_j + \vec{\omega}_j \times (\vec{X} - \vec{X}_j) \quad \text{on } \Gamma_j \tag{9}$$

is imposed at the interface, as it can be seen in (4d). This technique is well known for FSI problems, see, e.g., [6,7,17].

2.3. Solver scheme

To give a better overview, before the method is being described in detail in the next section, Fig. 2 shows the solver process on a short view:

First, the Navier–Stokes Equations are solved with given boundary conditions and the body surface. Afterwards the hydrodynamic forces are calculated by integrating the stress tensor over the body surface. Then, in the body solver part, the Newton-Euler equations are solved with the given forces and the position and angle is yield by integrating the kinematic equations in time. Finally, to set up the solver for the next timestep the body is moved to its new position and the new velocities are imposed as Dirichlet boundary conditions at moving body.

3. Cut cell immersed boundary method

In this section, the numerical background of the cut cell immersed boundary method within a discontinuous Galerkin context will be presented. It should be mentioned that the present solver is based on the work Kummer [25] has presented in terms of two phase flows and is implemented into the software package *BoSSS* [30]. First, some basic definitions and notations will be given. Afterwards, the discretization in time and space will be introduced. At last, a few words about the hierachical-moment fitting method, first published by Müller et al. [26], will be given.

Foremost, basic notations and definitions of the discontinuous Galerkin method have to be introduced:

- The computational domain $\Omega \subset \mathbb{R}^2$ has to be polygonal and simply connected.
- The numerical grid $K_h = \{K_1, \ldots, K_n\}$ with h being the maximum diameter of all cells.
- Let $\Gamma(t)$ be the union of all faces in the fluid region: $\Gamma(t) = \bigcup_j \partial(K_j \cap \Omega_f(t))$.
- Let $\Gamma_I(t)$ be the union of all interior faces $\Gamma_I(t) = \Gamma(t) \backslash \Gamma_D \backslash \Gamma_N \backslash \Gamma_j(t)$.
- Jump and average operators of a function ϕ are denoted by: $[\![\phi]\!] := \phi|_{K^+} - \phi|_{K^-}$ and $\{\phi\} := \frac{1}{2}\phi|_{K^+} + \frac{1}{2}\phi|_{K^-}$ on $\Gamma_I(t)$.
- a normal field $\vec{n}_\Gamma$ on $\Gamma(t)$.
- The broken gradient ∇_h: for $u \in C^1(\Omega_f)$, $\nabla_h u$ denotes the gradient on the domain Ω_f and $\nabla_h \cdot \vec{u}$ the broken divergence.

In order to formulate the discretization we also need to define the cut cell DG space (see Kummer [25]) in the following way:

$$\begin{aligned} \mathbb{P}_k(t, K_h) := \{ f \in L^2(\Omega_f(t)); \forall K \in K_h : f|_{K \cap \Omega_f(t)} \\ \text{is polynomial, } \deg(f|_{K \cap \Omega_f(t)}) \le k \}. \end{aligned} \tag{10}$$

Furthermore we need to define a mapping to extrapolate the known velocities $\vec{u}^n$ and $\vec{u}^{n-1}$ onto the new polynomial space:

$$\mathbb{E}_n^{K_h} : \mathbb{P}_k(t^\alpha, K_h) \to \mathbb{P}_k(t^\beta, K_h) \tag{11a}$$

with

$$\vec{u}^\alpha = \vec{u}^{\alpha *} \text{ for } \Omega_f(t^\alpha) \cap \Omega_f(t^\beta). \tag{11b}$$

It is important to note that, if the body surface exceeds a cell boundary, values for velocity and pressure will be polynomial extrapolated out of neighboring cells.

3.1. Temporal discretization

For the temporal discretization of (4) we use a second-order backward differencing formula (BDF2) and a Crank–Nicolson scheme for the rigid body solver. Further, let the total time T be divided into uniform timesteps Δt and proceed with the following scheme:

1. Assume initial conditions or results from the previous timestep t^n for body position $\vec{X}_j^n$, body angle Θ_j^n, body velocity $\vec{U}_j^n$ and body angular velocity ω_j^n. Set the new computational domain for the fluid phase $\Omega_f(t^{n+1})$.
2. Apply velocity boundary conditions at the body surface Γ_j following (9).
3. For proceeding a timestep we first need the previous defined extrapolated velocities:
$$\vec{u}^{n^*} = \vec{u}^n \text{ in } \Omega_f(t^{n+1}) \tag{12a}$$
and
$$\vec{u}^{n-1^*} = \vec{u}^{n-1} \text{ in } \Omega_f(t^{n+1}). \tag{12b}$$
4. Solve the fluid equations for $\vec{u}^{n+1}$ and p^{n+1} using a fixpoint iteration and the direct sparse solver *PARDISO* [31–33] for the linearized equation system. Since cut cells can become arbitrary small we use a procedure called cell agglomeration to restrict the condition number of the system [25,34]. For all calculations in this paper a cell agglomeration factor of $\gamma = 0.2$ is used. Thus, after extrapolation the discretized Navier–Stokes equations (4) in time is described by
$$\begin{aligned} \rho_f \left(\frac{3\vec{u}^{n+1}}{2\Delta t} - \frac{2\vec{u}^{n^*}}{\Delta t} + \frac{\vec{u}^{n-1^*}}{\Delta t} + \vec{u}^{n+1} \cdot \nabla \vec{u}^{n+1} \right) \\ + \nabla p^{n+1} - \mu_f \Delta \vec{u}^{n+1} = \vec{f}^{n+1} \end{aligned} \tag{13a}$$
and
$$\nabla \cdot \vec{u}^{n+1} = 0 \quad \text{in } \Omega_f(t^{n+1}). \tag{13b}$$
Resulting out of the BDF2-scheme the discretization of the Navier–Stokes equation handles to be second order in time.

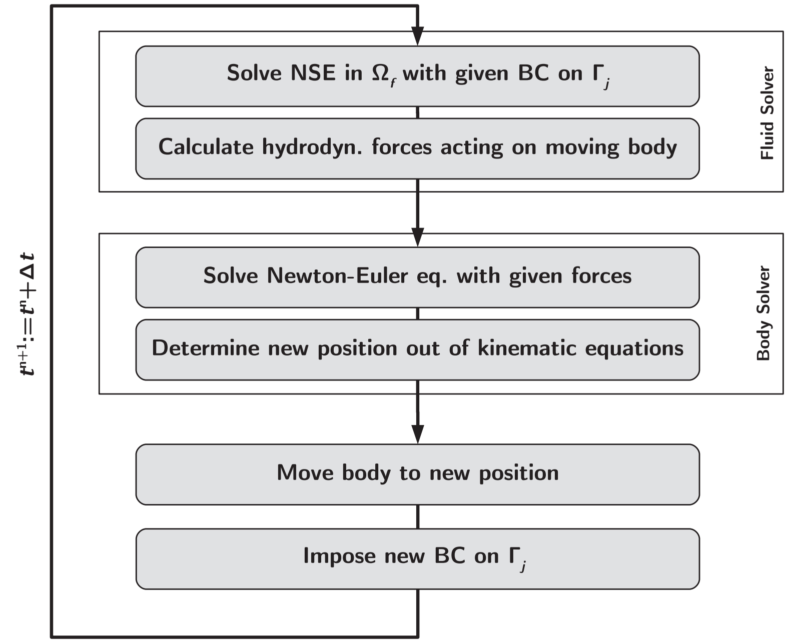

Solve NSE in Ω_f with given BC on Γ_j
Calculate hydrodyn. forces acting on moving body
Fluid Solver
Solve Newton-Euler eq. with given forces
Determine new position out of kinematic equations
Body Solver
Move body to new position
Impose new BC on Γ_j
$t^{n+1} := t^n + \Delta t$

Fig. 2. Solver process scheme.

5. Calculate the hydrodynamic force $\vec{F}_j^{n+1}$ and torque T_j^{n+1} acting on the body using the hierachical-moment-fitting on body surface Γ_j. For further details on the HMF quadrature, see Section 3.3.
6. Solve Newton–Euler equations (5) for new body velocities $\vec{U}_j^{n+1}$ and ω_j^{n+1} by using a second-order Crank–Nicolson method in time:

$$\vec{U}_j^{n+1} = \vec{U}_j^n + \left(\frac{\Delta M_j \vec{f}}{M_j} + \frac{\vec{F}_j^n + \vec{F}_j^{n+1}}{2M_j}\right)\Delta t, \tag{14}$$

$$\omega_j^{n+1} = \omega_j^n + \left(\frac{T_j^n + T_j^{n+1}}{2I_{jx}}\right)\Delta t, \tag{15}$$

where $I_{jx} = \frac{mr^2}{2}$ is the moment of inertia for rotation of a thin disk with radius r around its centerpoint and ΔM_j is the mass difference between body and fluid occupying the same volume.
7. Determine new body position $\vec{X}_j^{n+1}$ and angles θ_j^{n+1}:

$$\vec{X}_j^{n+1} = \vec{X}_j^n + \vec{U}_j^n \Delta t + \frac{1}{2}\left(\frac{\Delta M_j \vec{f}}{M_j} + \frac{\vec{F}_j^n + \vec{F}_j^{n+1}}{2M_j}\right)\Delta t^2, \tag{16}$$

$$\theta_j^{n+1} = \theta_j^n + \omega_j^n \Delta t + \frac{1}{2}\left(\frac{T_j^n + T_j^{n+1}}{2I_{jx}}\right)\Delta t^2. \tag{17}$$

8. Set $\vec{X}_j^n = \vec{X}_j^{n+1}$, $\vec{U}_j^n = \vec{U}_j^{n+1}$, $\omega^n = \omega^{n+1}$, $\theta^n = \theta^{n+1}$, $\vec{F}_j^n = \vec{F}_j^{n+1}$ and $T_j^n = T_j^{n+1}$ respectively.
9. Proceed to the next timestep by setting $t^n := t^{n+1}$ and restart with step 1.

In this context it should be noted, that using an explicit splitting renders the total scheme still to be first order in time only.

3.2. Spatial discretization

First, to simplify the notation of the spatial discretization let all domains and faces be part of the time level t^{n+1}, meaning $\Omega_f = \Omega_f(t^{n+1})$ and $\Gamma = \Gamma(t^{n+1})$. In order to satisfy the Ladyzĕnskaja–Babusk̆a–Brezzi condition of coercivity (e.g [35,36].) we discritize velocity and pressure in spaces of order k and $k' := k-1$, respectively. Admittedly to our knowledge there is no inf-sup stability proven the velocity/pressure pair applied. There is only an inf-sup condition published for $1 \leq k \leq 3$ in terms of a triangular grid with hanging nodes by Girault et al. [37]. Therefore the variational problem reads as follows:

Find

$$(\vec{u}^{n+1}, p^{n+1}) \in \mathbb{P}_k(t^{n+1}, \Omega_h)^2 \times \mathbb{P}_{k-1}(t^{n+1}, \Omega_h) =: V_k \tag{18a}$$

such that

$$\forall\ (\vec{v}, \tau) \in V_k$$

$$\begin{aligned}\left(\frac{3\vec{u}^{n+1}}{2\Delta t}, \vec{v}\right)_{\Omega_f} &+ t(\vec{w}^{n+1}, \vec{u}^{n+1}, \vec{v}) + b(p^{n+1}, \vec{v}) - a(\vec{u}^{n+1}, \vec{v}) \\ &- b(\vec{u}^{n+1}, \tau) = g(\vec{v}, \tau, \vec{u}^{n^*}, \vec{u}^{n+1^*}).\end{aligned} \tag{18b}$$

Here the first term denotes the temporal derivative term, $t(\vec{w}, \vec{u}, \vec{v})$ the convective term, $b(p, \vec{v})$ the pressure gradient, the $a(\vec{u}, \vec{v})$ the viscous terms and $b(\tau, \vec{u})$ the continuity term. For discretization we use a standard Symmetric Interior Penalty (SIP) method combined with a Lax–Friedrichs flux for the convective terms. The trilinear/bilinear forms can be seen in the following:

$$\left(\frac{3\vec{u}}{2\Delta t}, \vec{v}\right)_{\Omega_f} = \int_{\Omega_f} \frac{3}{2\Delta t}\vec{u} \cdot \vec{v}\ \mathrm{dV}, \tag{19}$$

$$\begin{aligned}t(\vec{w}, \vec{u}, \vec{v}) = &- \int_{\Omega_f} \rho_f (\vec{u} \otimes \vec{w}) : \nabla_h \vec{v}\ \mathrm{dV} \\ &+ \oint_{\Gamma} \rho_f (\{\vec{u} \otimes \vec{w}\}\vec{n}_\Gamma + (\lambda/2)[\![\vec{u}]\!]) \cdot [\![\vec{v}]\!]\ \mathrm{dS},\end{aligned} \tag{20}$$

$$b(p, \vec{v}) = -\int_{\Omega_f} p\ \nabla_h \vec{v}\ \mathrm{dV} - \oint_{\Gamma} [\![\vec{v}]\!] \cdot \vec{n}_\Gamma \{p\}\ \mathrm{dS}, \tag{21}$$

$$\begin{aligned}a(\vec{u}, \vec{v}) = &- \int_{\Omega_f} \mu_f (\nabla_h \vec{u} : \nabla_h \vec{v})\ \mathrm{dV} \\ &+ \oint_{\Gamma} \mu_f \{\nabla_h \vec{u}\}\vec{n}_\Gamma \cdot [\![\vec{v}]\!]\mathrm{dS}\end{aligned}$$

$$+ \oint_{\Gamma} \mu_f \{\nabla_h \vec{v}\} \vec{n}_\Gamma \cdot [\![\vec{u}]\!] \, \mathrm{dS} - \oint_{\Gamma} \eta [\![\vec{u}]\!] \cdot [\![\vec{v}]\!] \, \mathrm{dS} \tag{22}$$

and

$$b(\vec{u}, \tau) = - \int_{\Omega_f} \vec{u} \, \nabla_h \tau \, \mathrm{dV} - \oint_{\Gamma} [\![\tau]\!] \cdot \vec{n}_\Gamma \{\vec{u}\} \, \mathrm{dS}. \tag{23}$$

The penalty parameter η is determined by the following equation:

$$\eta = \mu \, k^2 \, G_{K_h}, \tag{24}$$

where μ is the so called penalty safety factor, k is the polynomial degree and G_{K_h} is a geometrical factor depending on the particular cut cell. If not stated otherwise, the safety factor is chosen to be 4 in all calculations. More details on the choice of the penalty parameter η and the Lax–Friedrichs parameter λ in (19) can be found in the work of Kummer [25].

Finally we can specify the source and boundary condition terms on the right-hand side of the variational problem in (18), which can be split into

$$g(\vec{v}, \tau, \vec{u}^{n^*}, \vec{u}^{n+1^*}) = q(\vec{v}) + s(\vec{v}) + p(\vec{v}) + r(\tau), \tag{25}$$

where the functional $q(\vec{v})$ denotes the Dirichlet boundary conditions of the convective operator, $s(\vec{v})$ the volume force and Dirichlet boundary conditions of the linear Stokes problem, $p(\vec{v})$ the right-hand-side of the temporal discretization using BDF2 and $r(\tau)$ the Dirichlet boundary conditions of the continuity equation. Details of the functionals are given in the following:

$$q(\vec{v}) = \oint_{\Gamma_D} ((\vec{u}_D \otimes \vec{u}_D) \vec{n}_{\Gamma_D} \cdot \rho_f \; + (\lambda/2) \; \vec{u}_D) \cdot [\![\vec{v}]\!] \, \mathrm{dS} + \oint_{\Gamma_j} ((\vec{u}_j \otimes \vec{u}_j) \vec{n}_{\Gamma_j} \cdot \rho_f + (\lambda/2) \; \vec{u}_j) \cdot [\![\vec{v}]\!] \, \mathrm{dS}, \tag{26}$$

$$s(\vec{v}) = - \int_{\Omega_f} \vec{f} \cdot \vec{v} \, \mathrm{dV} - \oint_{\Gamma_D} \vec{u}_D \cdot (\nabla_h \vec{v} \; \vec{n}_{\Gamma_D} - \eta \vec{v}) \, \mathrm{dS} - \oint_{\Gamma_j} \vec{u}_j \cdot (\nabla_h \vec{v} \; \vec{n}_{\Gamma_j} - \eta \vec{v}) \, \mathrm{dS}, \tag{27}$$

$$p(\vec{v}, \vec{u}^{n^*}, \vec{u}^{n+1^*}) = - \int_{\Omega_f} \left(\frac{2\vec{u}^{n^*}}{\Delta t} - \frac{\vec{u}^{n-1^*}}{\Delta t} \right) \cdot \vec{v} \, \mathrm{dV} \tag{28}$$

and

$$r(\tau) = \oint_{\Gamma_D} \tau \vec{u}_D \cdot \vec{n}_{\Gamma_D} \, \mathrm{dS} + \oint_{\Gamma_j} \tau \vec{u}_j \cdot \vec{n}_{\Gamma_j} \, \mathrm{dS}, \tag{29}$$

where Γ_D denotes all Dirichlet boundaries out of $\partial\Omega_f \backslash \partial\Omega_j$ and Γ_j denotes the Dirichlet boundaries on $\partial\Omega_j$. Therefore, velocity boundary conditions at the outer boundaries and the body interface are treat equally. Note that at this point the actual coupling between fluid and body interface takes place.

3.3. Integration on cut cells

The crucial part about a cut cell immersed boundary method is the accurate and efficient integration. In the section above it can be seen that we have to compute the integrals

$$\oint_{\Gamma_I} f \, \mathrm{dS} \quad \text{(inner edges between cut cells)},$$

$$\oint_{\Gamma_j} f \, \mathrm{dS} \quad \text{(body surface)},$$

and

$$\oint_{K \cap \Omega_f} f \, \mathrm{dV} \quad \text{(cut cell volumes)}$$

numerically on a reference cell $K \subset \mathbb{R}^2$, where f has to be sufficiently smooth. For our solver we use the method of hierachical-moment-fitting integration (HMF) proposed by Müller et al. [26]. The key point is the construction of HMF rules by using the Gauss theorem

$$\int_{K \cap \Omega_f} \nabla \cdot \vec{f} \, \mathrm{dV} - \oint_{K \cap \Gamma_j} \vec{f} \cdot \vec{n}_{\Gamma_j} \, \mathrm{dS} = \oint_{K \cap \Gamma_I} \vec{f} \cdot \vec{n}_{\Gamma_I} \, \mathrm{dS} \tag{30}$$

for a given vector field $\vec{f}$. Kummer [25] introduced a variant of the original HMF method which does not only use the Gauss theorem but also enforces the Stokes theorem which reads

$$\oint_{K \cap \Gamma_j} \frac{1}{r} \; \vec{n}_{\Gamma_j} \cdot \vec{f} - (I - \vec{n}_{\Gamma_j} \otimes \vec{n}_{\Gamma_j}) : \nabla \vec{f} \, \mathrm{dS} = - \int_{\Gamma_I \cap \Gamma_j} \vec{s} \cdot \vec{f} \, \mathrm{dL}, \tag{31}$$

where the integral on the RHS denotes a zero-dimensional point measure over all points in $\Gamma_I \cap \Gamma_j$, r is the radius of the circular shaped body, $\frac{1}{r}$ denotes its curvature and $\vec{s}$ is the outward tangent to Γ_j.

For numerical integration we define the notation

$$\int_{(\underline{\vec{x}}, \underline{w})}^{num} f := \sum_{i=1}^{L} w_i f(\vec{x}_i) \tag{32}$$

with weights w_i and nodes $\vec{x}_i$. In consequence of this we want to compute the least-square solution of the following system in order to get the weights $\underline{w}^{\Gamma_j}$ and $\underline{w}^{\Omega_f}$ for given nodes $\underline{\vec{x}}$ over body surfaces Γ_j and cut cell volumes $\Omega_f \cap K$:

$$\begin{aligned} &\int_{(\underline{\vec{x}}, \underline{w}^{\Omega_f})}^{num} \nabla \cdot \vec{f} - \oint_{(\underline{\vec{x}}, \underline{w}^{\Gamma_j})}^{num} \vec{f} \cdot \vec{n}_{\Gamma_j} \\ &\quad = \oint_{\Gamma_j} \vec{f} \cdot \vec{n}_{\Gamma_I} \, \mathrm{dS} \oint_{(\underline{\vec{x}}, \underline{w}^{\Gamma_j})}^{num} \frac{1}{r} \; \vec{n}_{\Gamma_j} \cdot \vec{f} - (I - \vec{n}_{\Gamma_j} \otimes \vec{n}_{\Gamma_j}) : \nabla \vec{f} \, \mathrm{dS} \\ &\quad = - \int_{\Gamma_I \cap \Gamma_j} \vec{s} \cdot \vec{f} \, \mathrm{dL} \forall \vec{f} \in \mathbb{P}_{k'}(K)^2 \end{aligned} \tag{33}$$

with surface weights $\underline{w}_{\Gamma_j}$ and volume weights $\underline{w}_{\Omega_f}$. Note that we use $k' = 3k$ for surface and volume integrals in cells and $k' = 3k + 2$ for the body surfaces. We additionally want to make a short comment in terms of the least-square solution. In the HMF we always have under-determined systems and therefore the solution with the smallest norm-value is being chosen.

Further details and variants of the HMF can be extracted from the references cited above. For the method presented in this paper the procedure prescribed above is being used. This is due to the fact that the immersed boundary should be more flexible in terms of non circular shaped surfaces in future progressions.

4. Numerical experiments

In the following, several testcases are presented which show the accuracy of our solver. At first the immersed boundary method is tested for fixed boundaries. Next, a given motion is added to test the accuracy of the solver to handle fixed motion. Afterwards the immersed boundary method will be coupled with the body solver for an easy non-moving interface test case. At the end a fully coupled testcase for a falling disk in incompressible fluid is presented.

4.1. Flow around a fixed cylinder

At first a steady test case for a laminar flow around a fixed cylinder at a Reynolds number of 20 was analyzed. The test case has benchmark character and was published by Schäfer et al. [38]. The geometry is a channel flow with channel height $H = 0.41$ and length $L = 2.2$. The cylinder is placed at (0.2, 0.2) and has a diameter of $D = 0.1$. Fig. 3 showing the computational domain which was taken from the reference cited above.

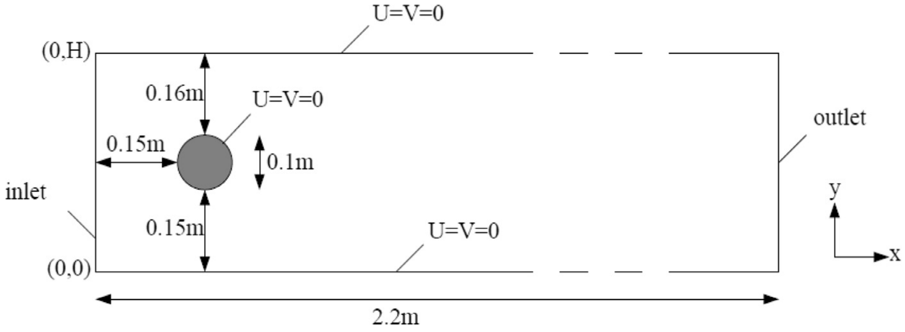


Fig. 3. Computational domain for cylinder flow from [38].

Table 1
Results for steady flow around a fixed cylinder at Re=20.

μ	k	Unknowns	C_D (± Tol.)		C_L (± Tol.)	
1	1	46 000	5.2644	(+5.7%)	0.0097	(+9.4%)
	2	45 000	5.5934	(-0.2%)	0.0108	(-0.9%)
	3	46 000	5.5866	(-0.1%)	0.0106	(+0.9%)
2	1	46 000	5.3057	(+4.9%)	0.0093	(+13.1%)
	2	45 000	5.6161	(-0.7%)	0.0108	(-0.9%)
	3	46 000	5.5905	(-0.2%)	0.0106	(+0.9%)
4	1	46 000	5.2607	(+5.7%)	0.0095	(+11.2%)
	2	45 000	5.6760	(-1.7%)	0.0108	(-0.9%)
	3	46 000	5.6049	(-0.5%)	0.0107	(± 0.0%)
		120 000	5.5851	(-0.1%)	0.0108	(-0.9%)
Schäfer et al. [38].			5.5800	(± 0.01)	0.0107	(± 0.0003)

The boundary conditions are as follows: At the left wall a velocity inlet with a parabolic velocity profile of $U(y) = 6y(H-y)/H^2)$ and $V = 0$ is applied. At the cylinder surface as well as at the upper and lower walls of the channel no-slip boundary conditions are imposed. As outflow boundary condition a common pressure outlet is used.

The results of calculations with three different penalty safety factors, see (24), in combination with polynomial degrees $k = 1, 2, 3$ may be taken from Table 1. The second column describes the polynomial degree of the test functions k, the third one are the total spatial unknowns, C_D for drag- and C_L for lift coefficient which are denoted through the following equations:

$$C_D = \frac{2F_D}{\rho_f \bar{U}^2 D} \tag{34a}$$

and

$$C_L = \frac{2F_L}{\rho_f \bar{U}^2 D} \tag{34b}$$

with F_D being the drag force, ρ_f being the density of the fluid phase, D being the diameter of the cylinder and $\bar{U}$ being the averaged velocity at the inlet. The deviation of the calculated values from the mean values of Schäfer [38] is also given in brackets.

It can be seen, that for the choice of all penalty safety factors better agreement with the literature is yield if the polynomial degree is increased and background mesh is coarsened simultaneously. This can also be confirmed by Fig. 4. Here the mean value is represented as dashed line and the tolerance of the mean value is shown by the shaded area.

It should be further mentioned that the penalty has an impact on drag and lift values especially if the general number of unknowns is low. In Table 1 also one calculation with an increased number of unknowns is presented and literature values are perfectly reached with a penalty safety factor of $\mu = 4$. Those results can also be seen in Fig. 4 indicated by black bars.

Next, the testcase is extended to an unsteady solution by increasing the Reynolds number to 100. The height of the channel is chosen again to $H = 0.41$ in order to trigger the vortex shedding, because the flow field at the cylinder position results to be slightly asymmetric.

As it can be seen in Table 2, different choices of the polynomial degree were tested. In Table 2 $(C_D)_{max}$ is the maximum drag, $(C_L)_{max}$ the maximum lift coefficient and

$$St = \frac{f_n \cdot D}{u} \tag{35}$$

being the Strouhal number. In (35) f_n is the natural shedding frequency and u the flow velocity.

It can be confirmed again that, with a fixed penalty safety factor, increasing the polynomial degree and keeping the total number of unknowns fixed, a much better result compared to the benchmark data of [38], see also Fig. 5. It has further to be stated that the results in the literature cited were obtained by different academic and commercial codes which all use a conventional body-fitted grid. Nevertheless, again, the penalty factor has a particular influence on drag and lift values. In contrast to the calculation at Re=20 the grid refinement leads only to drag coefficient which is closer to the literature whereas the lift coefficient is slightly deteriorated.

In consequence, those first testcases show the ability of our solver to handle a body immersed in the grid and an accurate integration of the total stresses for obtaining the hydrodynamical forces. However, here the hydrodynamical forces are only computed for validation purposes and the choice of the penalty safety factor has also an influence on the results. In the following, all calculations are performed with a penalty safety factor of $\mu = 4$ due to stability issues and the recommendation made by Kummer [25].

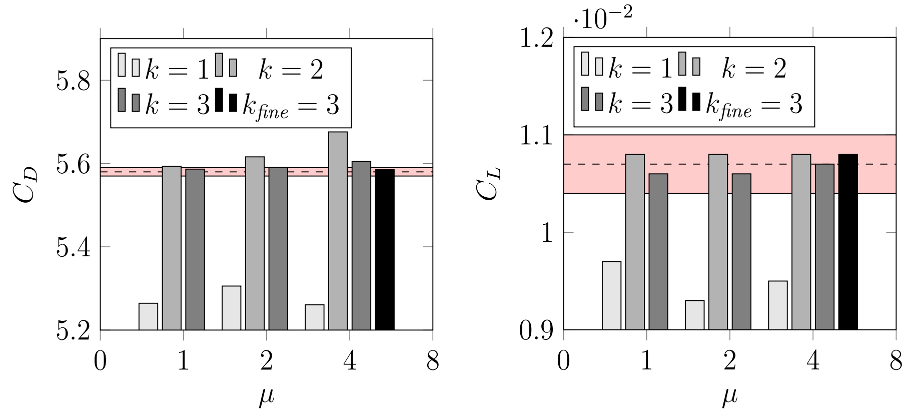
4.2. Flow around a transversally oscillating cylinder

In this test case the flow around a cylinder oscillating in a free stream is investigated. The motion is prescribed by a given function $\vec{X}_c$. The total motion can be considered as a free flow with an superimposed oscillating flow part. The oscillating flow field is induced by (36)

$$\rho_f \frac{d\vec{u}_c}{dt} = -\nabla p \quad \text{in } \Omega_f \tag{36}$$

with $\vec{u}_c$ being the time dependent oscillation velocity.

As a result of this summation, the force $\vec{F}_j$ acting on the cylinder has contributions from the force $\vec{F}_{FL}$ resulting from the free flow and the Froude–Krylov force $\vec{F}_{FK}$ resulting from the unsteady pressure field generated by the oscillations, see (37). In order to compare our lift coefficient with literature values, where the coordinate system is fixed to the cylinder, the total hydrodynamic force, which is calculated by solving the (7a), has to be subtracted

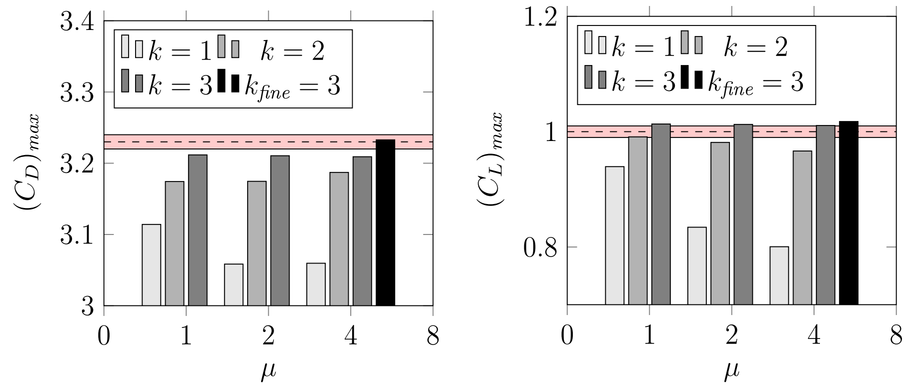


(a) Impact of penalty factor on Drag coef. (b) Impact of penalty factor on Lift coef.

Fig. 4. Results of different penalty factors combined with different polynomial degrees for Re=20.

Table 2
Results for unsteady flow around a fixed cylinder at Re=100.

Δt	μ	k	Unknowns	$(C_D)_{max}$	(± Tol.)	$(C_L)_{max}$	(± Tol.)	St	(± Tol.)
0.05	1	1	46 000	3.1141	(+3.6%)	0.9394	(+6.1%)	0.2941	(+2.0%)
		2	45 000	3.1743	(+1.7%)	0.9910	(+0.9%)	0.2985	(+0.5%)
		3	46 000	3.2117	(+0.6%)	1.0133	(-1.3%)	0.3030	(-0.1%)
0.05	2	1	46 000	3.0584	(+5.3%)	0.8343	(+16.6%)	0.2985	(+0.5%)
		2	45 000	3.1746	(+1.7%)	0.9814	(+1.9%)	0.2985	(+0.5%)
		3	46 000	3.2105	(+0.6%)	1.0126	(-1.3%)	0.3030	(-0.1%)
0.05	4	1	46 000	3.0596	(+5.3%)	0.8005	(+20.1%)	0.2985	(+0.5%)
		2	45 000	3.1871	(+1.3%)	0.9665	(+3.4%)	0.2985	(+0.5%)
		3	46 000	3.2091	(+0.7%)	1.0107	(-1.1%)	0.3030	(-0.1%)
			120 000	3.2328	(+0.1 %)	1.0176	(-1.8%)	0.3030	(-0.1%)
	Schäfer et al. [38].			3.2300	(± 0.01)	1.0000	(± 0.01)	0.3000	(± 0.0050)



(a) Impact of penalty factor on Drag coef. (b) Impact of penalty factor on Lift coef.

Fig. 5. Results of different penalty factors combined with different polynomial degrees for Re=100.

by $\vec{F}_{FK}$. This technique is well-known and was used by Meneghini and Bearman [39].

$$\vec{F}_j = \vec{F}_{FL} + \vec{F}_{FK}, \qquad (37)$$

where

$$\vec{F}_{FK} = -\int_{\partial\Omega_j} p\cdot\vec{n}\ \mathrm{ds} \overset{\text{Gauss}}{=} -\int_{\Omega_j}\nabla p\ \mathrm{dV} \overset{(36)}{=} \rho_f\int_{\Omega_j}\frac{d\vec{u}_c}{dt}\ \mathrm{dV} = \rho_f V_j\frac{d^2\vec{X}_c}{dt^2}. \qquad (38)$$

The geometrical data of this test case was first described by Lai and Peskin [12] and later used by Uhlmann [16]. A cylinder with a diameter of $d = 0.3$ is placed at the origin of the computational domain $\Omega = [-1.85, 6.15] \times [-4, 4]$. At $x = -1.85$ a velocity inlet condition with a uniform free stream velocity of $\vec{u} = (1, 0)$ is imposed. A pressure outlet condition for all three other outer boundaries is used. The cylinder surface is treated by using velocity boundary conditions of the oscillating motion and the Reynolds number is set to $Re = 185$. Furthermore the cylinder is forced to oscillate only in y-direction by describing its position using the following function:

$$y_c(t) = 0.2\ d\ \cos(2\pi f_f t), \qquad (39)$$

with d being the diameter of the cylinder and f_f being 0.8 times the natural shedding frequency f_n at $Re = 185$ in case of a non-oscillating cylinder. The domain is discretized using a mesh with hanging nodes in order to obtain a fine resolution for the cylinder surface without increasing the total number of unknowns mas-

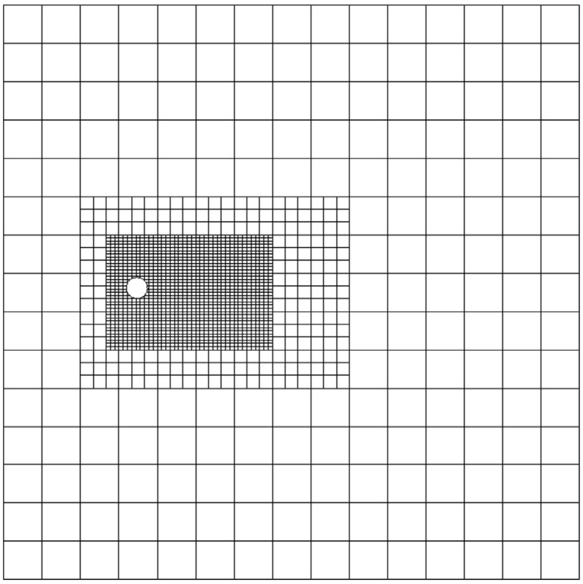


Fig. 6. Background mesh at $t = t_0$ for calculation with polynomial degree of $k = 3$.

Table 3
Results for unsteady flow around an oscillating cylinder at Re 185 with $\Delta t = 0.1$.

Method	k	Unknowns	$\overline{C_D}$	$C_D^{'}$	$(C_L)_{rms}$
Cut Cell DG	1	61 000	1.24	± 0.067	0.247
	2	55 000	1.27	± 0.057	0.235
	3	47 000	1.26	± 0.078	0.225
Uhlmann [16]			1.380	± 0.063	0.176
[16] with Kajishima and Takiguchi forcing [41]			1.282	± 0.088	0.223
Lu and Dalton [40]			1.25		0.18

sively. The coarsest background mesh for the calculation using a polynomial degree of 3 for velocity and 2 for pressure can be seen in Fig. 6.

As a result of keeping the total number of unknowns fixed for different polynomial degrees three meshes have to be created. Results of those calculations can be seen in Table 3. Here the mean drag coefficient $\overline{C_D}$, the amplitude of the drag coefficient $C_D^{'}$ and the root-mean-square value of the lift coefficient $(C_L)_{rms}$ are compared with calculations out of [16] (immersed boundary method) and [40] (body-fitted grid). Additionally, Uhlmann [16] implemented in his work the forcing scheme of Kajishima and Takiguchi [41] and published his calculation results.

It can be seen, that the results of our method are in good agreement with the literature cited. The best agreement is yield if we compare the results with the forcing scheme of Kajishima et al. [41,42]. In their work they use a volume integration over the so-called interacting force which is a linear interpolation between the fluid velocity and the particle velocity, also including cells which are cut by the interface using a volume fraction approach.

In contrast to the observation made by [16] using the method of [42] we cannot observe strong oscillations in the drag hysteresis like Uhlmann. The drag hysteresis of the calculation using a polynomial degree of $k = 3$ and a timestep of $\Delta t = 0.1$ can be seen in Fig. 7. This is probably because we use a coarser background mesh discretization, naming $d/h = 5$ for $k = 3$ in comparison with Uhlmann $d/h = 38.4$, where h is the mesh width. Also the use of an implicit time stepping scheme and a larger timestep, i.e., 64 timesteps per cylinder oscillation, certainly has an influence.

Nevertheless the rms-value of the lift coefficient is predicted much higher than in the reference calculation of Lu and Dalton [40]. This is a result which is not observed by the immersed boundary method of Uhlmann. To sum up, very good agreement with literature values can be obtained using again a smaller amount of unknowns at a higher-order DG approximation. Therefore our solver is able to handle fixed motion within the grid accurately.

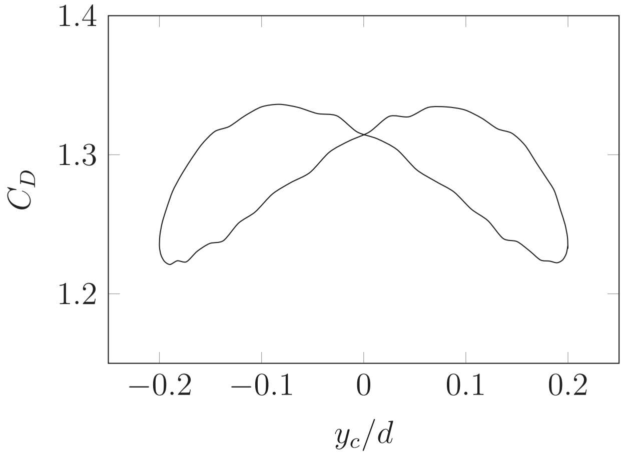


Fig. 7. Drag hysteresis for one cylinder oscillation.

Table 4
Results Particle in Shear Flow.

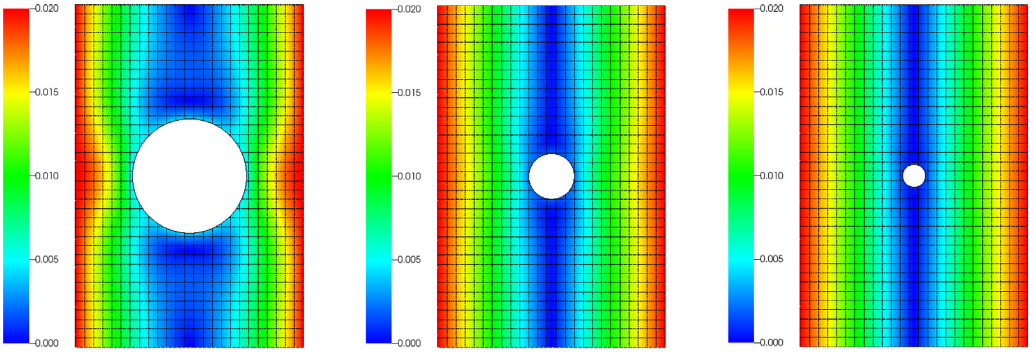
Particle radius R	Terminal angular velocity ω_j	
	Cut Cell DG	Wan and Turek [14]
1	0.0042972	0.0043148
0.4	0.0049177	0.0048697
0.2	0.0049488	0.0049584

4.3. *One rotating particle in a shear flow*

The coupling between cylinder and fluid is first tested in a fixed domain environment which was described by Wan and Turek [14]. Therefore, we place a particle between two walls, which move in opposite directions with a velocity $v_y = 0.02$ and $v_y = -0.02$ resp., creating a linear shear flow in between the walls. The channel height is $H = 6$, its width is $W = 4$ and the particle center is placed at position $\vec{X} = (2, 3)$. Top and bottom of the computational domain are bounded by using periodic boundary conditions in y-direction. The density of the particle and the fluid are chosen equally to $\rho_f = \rho_p = 1$. The viscosity of the fluid is denoted by $\mu_f = 0.01$.

The particle is fixed but can rotate around its center according to the resulting torque from the hydrodynamical forces. Initially, the particle as well as the fluid are at rest. If the radius of the particle is small enough the particle angular velocity should be close to $U_p = 0.005$ according to the vorticity field in the linear shear flow while reaching the steady state solution much faster with decreasing radius R. For our calculations we use different particle radii and measure the terminal angular velocity. At the same time we use a polynomial degree of $k = 3$ for velocity and a time step size of $\Delta t = 1$ for $R = 1$, $\Delta t = 0.5$ for $R = 0.4$ and $\Delta t = 0.25$ for $R = 0.2$. In total the system for the smallest radius has around 25 000 unknowns in space, whereas Wan and Turek [14] use 710 000 unknowns. The values obtained are compared with the results out of [14] in Table 4.

It can be seen that our results are in good agreement with the reference cited. Additionally it has to be stated that those results have never been compared to experiments. Therefore Table 4 only



(a) R=1 (b) R=0.4 (c) R=0.2

Fig. 8. Velocity magnitude in the steady state for different radii with background mesh.

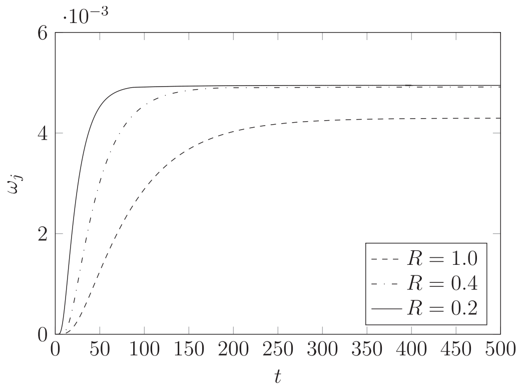


Fig. 9. Angular velocity of particle with different radii over time.

Table 5
Experimental order of h-convergence.

k/k'	EOC velocity	EOC pressure
1/0	1.4	1.3
2/1	3.3	1.9
3/2	4.0	2.8

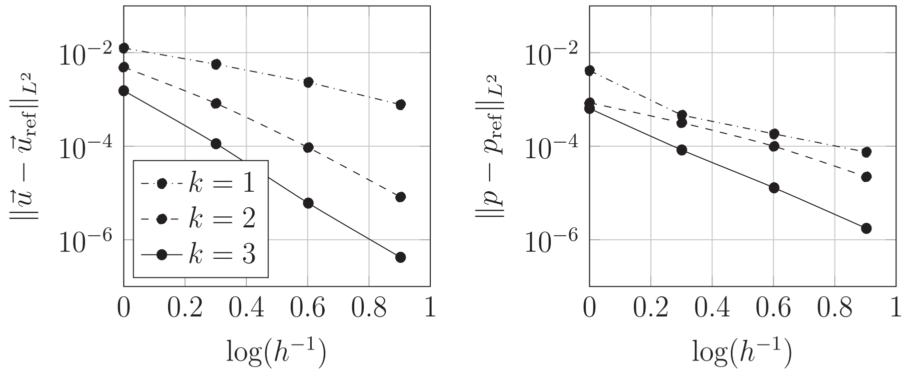
relates both methods. Nevertheless the predicted physical behavior of the increasing terminal angular velocity with shrinking diameter has been proofed. A plot of the velocity magnitude and the background mesh for all three calculations can be seen in Fig. 8. In addition, the angular velocity over time is plotted in Fig. 9 to proof the second prediction of less time needed to reach the steady state with decreasing radius. Thus it can be seen that the smaller the diameter the faster the terminal angular velocity of the particle is reached.

A modification of the present test case is also used to carry out a mesh size convergence study. For this we choose the particle radius to be $R = 1$ and turn off the coupling, such that a no slip boundary condition is imposed at the particle surface. Therefore, we yield a steady solution in time. For each polynomial degree k for velocity and $k' = k - 1$ for pressure we calculate a reference solution by using a very fine grid. Then four calculations for each polynomial degree were carried out and the L^2-Error in comparison with the aforementioned reference solution is determined. In Fig. 10 the convergence behavior can be seen. With the exception of the coarsest mesh for the pressure at polynomial degree of $k = 1$ linear decreasing functions can be observed. This is due to the fact that for the calculation mentioned above the mesh is under resolved.

We expect to yield an experimental order of convergence (EOC) of $k + 1$ for velocity and k for pressure. Because of the underresolved calculations for the coarsest mesh size those were neglected in determining the EOC in Table 5. All in all good agreement with our expectations can be seen and therefore the reliability of our method in terms of h-refinement is proven.

4.4. Single disk falling in incompressible fluid

The last testcase was calculated in order to proof the ability of our solver to handle coupled particle motion within a fluid with moving domains. Therefore we chose the testcase of a single disk



(a) Velocity (b) Pressure

Fig. 10. *h*-convergence.

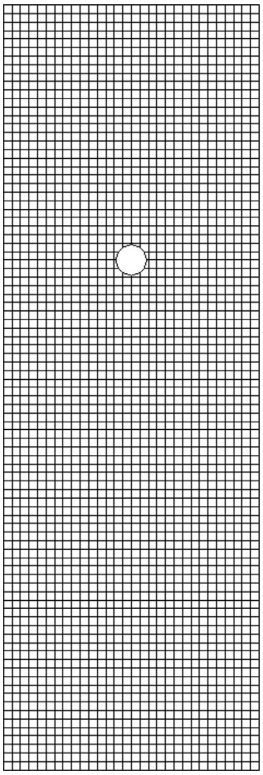


Fig. 11. Background mesh at $t = t_0$ for calculation with polynomial degree of $k = 3$.

falling in incompressible fluid due to gravity force. This testcase was also studied by Wan and Turek [14].

The dimensions of the computational domain are $W = 2$ and $H = 6$ and the particle with diameter of $d = 0.25$ is centered at $X_p = (1, 4)$ at time $t = t_0$. The density ratio of the particle and fluid is denoted with $\frac{\rho_p}{\rho_f} = 1.25$. For validation purposes the particle Reynolds number is calculated every timestep using $Re_p = |\vec{U}| \cdot d \cdot \rho_p / \nu$. Again calculations have been conducted for three different polynomial degrees, keeping the total number of unknowns almost fixed. The background mesh for the calculation with $k = 3$ can be seen in Fig. 11 at initial time. The mesh is equally distributed in the whole computational domain and the resolution is around four cells in direction of the particle diameter.

Table 6 shows results of our calculations. Here, a decrease of particle Reynolds number can be observed if the polynomial degree is increased. Thus the drag force acting on the particle is underestimated with a low order approximation and the particle speed increases. The same observation has also been made by Wan and Turek [14]. Additionally we compare our results to Glowinski et al. [9], who use a implicit immersed boundary approach with Lagrange multipliers. In summary, using an explicit coupling approach yields a smaller particle Reynolds number than using implicit coupling. However, the total falling velocity of our method appears to be in a similar range as the results from the references cited above. Again, we can yield similar results by decreasing the total number of unknowns by a factor of more than two, which confirms the accuracy of our method also for coupled motion.

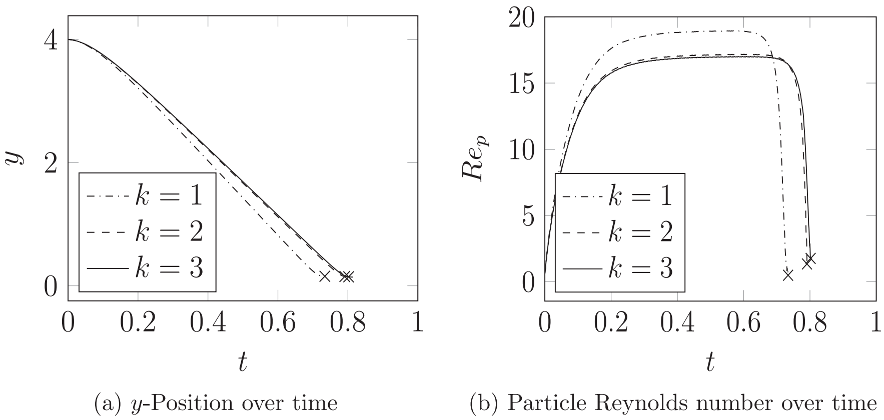
Table 6
Results single particle falling in incompressible fluid.

Method	Δt	k	Unknowns	max Re_p
Cut Cell DG	10^{-3}	1	73 000	18.95
	10^{-3}	2	72 000	17.18
	10^{-3}	3	70 500	17.00
Wan and Turek [14]			139 000	17.42
			354 000	17.15
Glowinski et al. [9].	7.5 x 10^{-4}		1 768 000	17.31

In Fig. 12 the particle position and the particle Reynolds number are plotted over time. The crosses indicate the termination time of the calculation when the particle comes too close to the lower wall and the simulation was stopped, because no collision model is implemented so far. It can be observed, that the coarser the background mesh the earlier the simulation has to stop despite using a higher polynomial degree for the approximation or in other words: There has to be a particular amount of background cells between the particle surface and the wall.

For conducting a convergence study in time the timespan from release of the disk until $t = 0.02$ was used and divided by an increasing number of time steps, starting by choosing two. Also the same configuration as for the setting with a polynomial degree of three was being used. For each calculation the number of time steps were divided by 2 and a reference solution was calculated by using 256 time steps. The L^2-Error for velocity can be seen in Fig. 13.

As it is prescribed in the previous section, because of the splitting approach our method renders to be first order in time only. We use a BDF2-scheme and therefore have to start the calculation with one step of BDF1, which also renders our method of first order in time only. Resulting out of those reasons the expected experimental order of convergence can be confirmed by Fig. 13.



(a) y-Position over time
(b) Particle Reynolds number over time

Fig. 12. Results of a single disk falling in incompressible fluid.

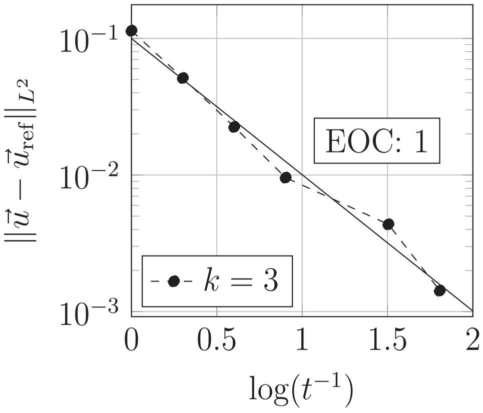


Fig. 13. *t*-convergence.

5. Conclusion

In this paper we have presented a 2D Discontinuous Galerkin immersed boundary method with first applications to moving body flow problems. Our method has been included in the software package *BoSSS* [30]. We have obtained good agreement with literature for fixed and moving boundary test cases. Due to our higher-order DG ansatz we could yield similar results, while at the same time using a considerable smaller number of unknowns, which is the main advantage of the proposed scheme. Additionally we could point out strong correlations between the choice of the SIP penalty parameter and physical quantities like lift and drag forces, which is important to state. For our method also less pressure oscillations could be observed. Also we have proven the reliability of our method by conducting convergence studies for time and space. Nevertheless, despite of all advantages, the splitting approach in our method renders to be only first order in time.

This leads to ongoing and further extensions of our solver. A key advantage of using the hierarchical-moment fitting strategy to calculate hydrodynamic forces is that we are not restricted to circular shaped bodies. In addition to bodies with different size and shape it is planned to extend the solver in near future to 3D flow problems. As a consequence, our method has to be accelerated to handle the increasing computational effort effectively in three-dimensional computations. On top of that, we want to increase the order of time integration and therefore some tests have already been made using alternatives and improvements for the splitting type approach. Also, a first solution to tackle the strong dependency of the penalty parameter came up. To investigate this idea, further test calculations have to be done in the future.

Acknowledgments

This work is supported by the 'Excellence Initiative' of the German Federal and State Governments and the Graduate School of Computational Engineering at Technische Universität Darmstadt. F. Kummer is supported by the German DFG through Collaborative Research Centre 1194/B06.

References

[1] Hu HH, Joseph DD, Crochet MJ. Direct simulation of fluid particle motions. Theor Comput Fluid Dyn 1991;3(5):285–306.
[2] Hu HH. Direct simulation of flows of solid-liquid mixtures. Int J Multiphase Flow 1996;22(2):335–52.
[3] Maury B. Characteristics ALE method for the unsteady 3D navier-stokes equations with a free surface. Int J Comut Fluid Dyn 1996;6(3):175–88.
[4] Maury B. Direct simulations of 2D fluid-particle flows in biperiodic domains. J Comput Phys 1999;156(2):325–51.
[5] Peskin CS. Flow patterns around heart valves: A digital computer method for solving the equations of motion, New York: Yeshiva University; 1972. Dissertation.
[6] Mittal R, Iaccarino G. Immersed boundary methods. Annu Rev Fluid Mech 2005;37(1):239–61.
[7] Hou G, Wang J, Layton A. Numerical methods for fluid-structure interaction — a review. Commun Comput Phys 2012;12(August (2)):337–77.
[8] Glowinski R, Pan TW, Hesla TI, Joseph DD. A distributed lagrange multiplier/fictitious domain method for particulate flows. Int J Multiphase Flow 1999;25(5):755–94.
[9] Glowinski R, Pan TW, Hesla TI, Joseph DD, Périaux J. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. J Comput Phys 2001;169(2):363–426.
[10] Patankar NA, Singh P, Joseph DD, Glowinski R, Pan TW. A new formulation of the distributed lagrange multiplier/fictitious domain method for particulate flows. Int J Multiphase Flow 2000;26(9):1509–24.
[11] Fogelson AL, Peskin CS. A fast numerical method for solving the three-dimensional stokes' equations in the presence of suspended particles. J Comput Phys 1988;79(1).
[12] Lai M-C, Peskin CS. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. J Comput Phys 2000;160(2):705–19.
[13] Höfler K, Schwarzer S. Navier–Stokes simulation with constraint forces: finite-difference method for particle-laden flows and complex geometries. Physical Review E 2000;61(6):7146–60.
[14] Wan D, Turek S. Direct numerical simulation of particulate flow via multigrid FEM techniques and the fictitious boundary method. Int J Numer Methods Fluids 2006;51(5):531–66.
[15] Duchanoy C, Jongen TRG. Efficient simulation of liquid–solid flows with high solids fraction in complex geometries. Comput Fluids 2003;32(10):1453–71.
[16] Uhlmann M. An immersed boundary method with direct forcing for the simulation of particulate flows. J Comput Phys 2005;209(2):448–76.
[17] Peskin CS. The immersed boundary method. Acta Numerica 2002;11:479–517.
[18] Feng Z-G, Michaelides EE. The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems. J Comput Phys 2004;195(2):602–28.
[19] Luo H, Dai H, Ferreira de Sousa P. A hybrid formulation to suppress the numerical oscillations caused by immersed moving boundaries. APS Division of Fluid Dynamics Meeting Abstracts; 2009.
[20] Luo H, Yin B, Dai H, Doyle J. A 3D computational study of the flow-structure interaction in flapping flight. 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition. American Institute of Aeronautics and Astronautics; 2010.
[21] Liao C-C, Chang Y-W, Lin C-A, McDonough JM. Simulating flows with moving rigid boundary using immersed-boundary method. Comput Fluids 2010;39(1):152–67.
[22] Mittal R, Dong H, Bozkurttas M, Najjar FM, Vargas A, von Loebbecke A. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. J Comput Phys 2008;227(10):4825–52.
[23] Seo JH, Mittal R. A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations. J Comput Phys 2011;230(19):7347–63.
[24] Reed WH, Hill TR. Triangular mesh methods for the neutron transport equation. Tech. Rep.. Los Alamos Scientific Lab., N.Mex. (USA); 1973.
[25] Kummer F. Extended discontinuous Galerkin methods for two-phase flows: the spatial discretization. Int J Numer Methods Eng 2016. n/a–n/a
[26] Müller B, Kummer F, Oberlack M. Highly accurate surface and volume integration on implicit domains by means of moment-fitting. Int J Numer Methods Eng 2013;96(8):512–28.
[27] Moës N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. Int J Numer Methods Eng 1999;46(1):131–50.
[28] Groß S, Reusken A. An extended pressure finite element space for two-phase incompressible flows with surface tension. J Comput Phys 2007;224(1):40–58.
[29] Bastian P, Engwer C. An unfitted finite element method using discontinuous galerkin. Int J Numer Methods Eng 2009;79(12):1557–76.
[30] Kummer F. The BoSSS Discontinuous Galerkin solver for incompressible fluid dynamics and an extension to singular equations.. Darmstadt: Technische Universität; 2012.
[31] Schenk O, Gärtner K. Two-level dynamic scheduling in PARDISO: improved scalability on shared memory multiprocessing systems. Parallel Comput 2002;28(2):187–97.
[32] Schenk O, Gärtner K. Solving unsymmetric sparse systems of linear equations with PARDISO. Future Generation Computer Systems 2004;20(3):475–87.
[33] Schenk O, Gärtner K. On fast factorization pivoting methods for sparse symmetric indefinite systems. Electron Trans Numer Anal 2006;23(1):158–79.
[34] Müller B, Krämer-Eis S, Kummer F, Oberlack M. A high-order discontinuous Galerkin method for compressible flows with immersed boundaries. Int J Numer Methods Eng 2016. n/a–n/a
[35] Babuška I. The finite element method with lagrangian multipliers. Numerische Mathematik 1973;20(3):179–92.
[36] Brezzi F. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. ESAIM: Math Modell Numer Anal - Modélisation Mathématique et Analyse Numérique 1974;8(R2):129–51.
[37] Girault V, Rivière B, Wheeler M. A discontinuous Galerkin method with nonoverlapping domain decomposition for the Stokes and Navier–Stokes problems. Math Comput 2005;74(249):53–84.

[38] Schäfer M, Turek S, Durst F, Krause E, Rannacher R. Benchmark computations of laminar flow around a cylinder. In: Hirschel PDEH, editor. Flow Simulation with High-Performance Computers II. Notes on Numerical Fluid Mechanics (NNFM). Vieweg+Teubner Verlag; 1996. p. 547–66. doi:10.1007/978-3-322-89849-4_39.

[39] Meneghini JR, Bearman PW. Numerical simulation of high amplitude oscillatory flow about a circular cylinder. J Fluids Struct 1995;9(4):435–55.

[40] Lu XY, Dalton C. Calculation of the timing of vortex formation from an oscillating cylinder. J Fluids Struct 1996;10(5):527–41.

[41] Kajishima T, Takiguchi S. Interaction between particle clusters and particle-induced turbulence. Int J Heat Fluid Flow 2002;23(5):639–46.

[42] Kajishima T, Takiguchi S, Hamasaki H, Miyake Y. Turbulence structure of particle-laden flow in a vertical plane channel due to vortex shedding. JSME Int J Ser B 2001;44(4):526–35.