INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2015; **00**:1–16
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme

A high-order Discontinuous Galerkin method for compressible flows with immersed boundaries

B. Müller*[,1,2], S. Krämer-Eis[1,2], F. Kummer[1,2], M. Oberlack[1,2,3]

[1] *Department of Fluid Dynamics, Technische Universität Darmstadt, 64287 Darmstadt, Germany*
[2] *Graduate School of Computational Engineering, Technische Universität Darmstadt, 64293 Darmstadt, Germany*
[3] *Center of Smart Interfaces, Technische Universität Darmstadt, 64287 Darmstadt, Germany*

SUMMARY

Great paper, much novelty.

BM: Write abstract

Received . . .

KEY WORDS: discontinuous galerkin method; immersed boundary; level set; quadrature; compressible flow

1. INTRODUCTION

Introduction.

Novelty compared to [1]: Higher orders, Navier-Stokes, no integration sub-cells, no artificial boundary conditions due to level set

BM: Write introduction

1.1. State of the art

IBM in general: [2], [3]

State of the art for higher order IBM:
[4] - low order - Compressible: [5] [6] [7], [8] - Incompressible multiphase: [9] [10]
- DG: - Euler: [1] - NS: [11], [12] - Poisson: [13] [14] [15]. - Two-phase: [16] [17]
- XFEM: [18] [19] [20] [21] [22] [17] [23] [24]
- high order Finite Difference: Navier-Stokes [25]

*Correspondence to: Björn Müller, Chair of Fluid Dynamics, Technische Universität Darmstadt, 64287 Darmstadt, Germany. E-mail: mueller@fdy.tu-darmstadt.de

Contract/grant sponsor: The work of S. Krämer-Eis, B. Müller and F. Kummer was supported by the 'Excellence Initiative' of the German Federal and State Governments and the Graduate School of Computational Engineering at Technische Universität Darmstadt. The work of B. Müller was supported by the German Science Foundation (DFG) through Research Grant WA 2610/2-1. The work of F. Kummer was supported by the German DFG through Research Fellowship KU 2719/1-1. Some of the numerical results in this work have been obtained using the high performance computer Lichtenberg at the Technische Universität Darmstadt.

2. GOVERNING EQUATIONS

The non-dimensional Navier-Stokes equations in conservative form read

$$\frac{\partial \vec{U}}{\partial t} + \frac{\partial \vec{F}_i^c(\vec{U})}{\partial x_i} - \frac{\partial \vec{F}_i^v(\vec{U}, \nabla \vec{U})}{\partial x_i} = 0, \tag{1}$$

with the variable vector

$$\vec{U} = \begin{pmatrix} \rho \\ \rho u_i \\ \rho E \end{pmatrix}, \tag{2}$$

the convective fluxes

$$\vec{F}_d^c = \begin{pmatrix} \rho u_i \\ \rho u_i u_j + p\delta_{ij} \\ u_i (\rho E + p) \end{pmatrix}, \tag{3}$$

and the diffusive fluxes

$$\vec{F}_i^v = \frac{1}{\mathrm{Re}} \begin{pmatrix} 0 \\ \tau_{ij} \\ \tau_{ij} u_j + \frac{\gamma}{\mathrm{Pr}(\gamma - 1)} q_i \end{pmatrix}. \tag{4}$$

The system is closed using the ideal gas law

$$p = (\gamma - 1)\rho e, \tag{5}$$

Fourier's law

$$q_i = k \frac{\partial T}{\partial x_i} \tag{6}$$

and Newtonian stress

$$\tau_{ij} = \mu \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_j} \right) - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right] \tag{7}$$

SKE: Governing equations for compressible Navier-Stokes

3. DISCRETIZATION

In the following, we briefly outline the standard Discontinuous Galerkin Method (DG) discretization of a generic conservation law, before discussing the differences to our implementation of an DG Immersed Boundary Method (IBM) scheme in Section 3.2. The most important building blocks for this scheme are the numerical integration in cut cells and a cell-agglomeration strategy, which will be discussed in Section 1 and Section 3.5, respectively.

3.1. Standard Discontinuous Galerkin discretization

For the sake of simplicity, we will introduce the basic form of the DG method using the example of the scalar conservation law

$$\frac{\partial c}{\partial t} + \nabla \cdot \vec{f}(c) = 0 \tag{8}$$

for some concentration $c = c(\vec{x}, t)$ with $\vec{x} \in \Omega \subset \mathbb{R}^D$, $t \in \mathbb{R}_0^+$ and a smooth function $\vec{f} : \mathbb{R} \to \mathbb{R}^D$, supplemented with suitable initial and boundary conditions. The extension to the Euler equations is straightforward, while viscous terms in the Navier-Stokes equations need special attention. Here, we directly follow the approach presented in [26] without any modifications, which is why we omit the details at this point.

Let Ω_h be a discretization of Ω with a characteristic mesh parameter h that represents a measure for the size of the affine cells $\{\mathcal{K}_i\}_{i=1,\dots,N}$ forming a tesselation of Ω_h. Within this setting, consider

DOI: 10.1002/nme

the set of polynomial basis and test functions $\{\Phi_{i,j}\}_{j=1,\dots,M}$ of maximum degree P, where

$$\operatorname{supp}\left(\Phi_{i,j}\right) = \mathcal{K}_i. \tag{9}$$

For each cell $\mathcal{K}_i$ with outward unit normal vector $\vec{n}$, the cell-local basis is then given by the row vector $\vec{\Phi}_i = (\Phi_{i,1}, \dots, \Phi_{i,M})$.

BM: Include assumption that $\mathcal{K}_i$ is affine linear?

We multiply Equation (8) by $\Phi_{i,j}$, integrate over the cell $\mathcal{K}_i$ and perform an integration by parts in order to obtain

$$\int_{\mathcal{K}_i} \frac{\partial c}{\partial t} \Phi_{i,j}\, dV + \int_{\partial\mathcal{K}_i} \left(\vec{f}(c)\cdot\vec{n}\right) \Phi_{i,j}\, dA - \int_{\mathcal{K}_i} \vec{f}(c)\cdot\nabla\Phi_{i,j}\, dV = 0. \tag{10}$$

We choose to discretize c by means of the modal approximation

$$c(\vec{x},t)|_{\mathcal{K}_i} \approx \tilde{c}(\vec{x},t)|_{\mathcal{K}_i} = \tilde{c}_i(\vec{x},t) = \sum_{k=0}^{M} \tilde{c}_{i,k}(t)\, \Phi_{i,k}(\vec{x}) \tag{11}$$

with the column vectors of coefficients $\vec{\tilde{c}}_i = \vec{\tilde{c}}_i(t) = \left(\tilde{c}_{i,1}(t), \dots, \tilde{c}_{i,M}(t)\right)^{\mathrm{T}}$.

Let the edges of $\mathcal{K}_i$ be denoted by $\mathcal{E}_{i,1}, \dots, \mathcal{E}_{i,E}$. Then inserting (11) into (10) and replacing $\vec{f}(c)\cdot\vec{n}$ by the numerical flux function $f = f(\tilde{c}^-, \tilde{c}^+, \vec{n})$ in the surface integral in (10) leads to the local form

$$\int_{\mathcal{K}_i} \frac{\partial \tilde{c}_i}{\partial t} \Phi_{i,j}\, dV + (\vec{f}_i)_j = 0 \tag{12}$$

of a semi-discrete set of equations with

$$(\vec{f}_i)_j \coloneqq \sum_{e=1}^{E} \int_{\mathcal{E}_e} f(\tilde{c}^-, \tilde{c}^+, \vec{n})\, \Phi_{i,j}\, dA - \int_{\mathcal{K}_i} \vec{f}(\tilde{c}_i)\cdot\nabla\Phi_{i,j}\, dV \tag{13}$$

Here,

$$\tilde{c}^- = \tilde{c}^-(\vec{x},t) \coloneqq \lim_{\epsilon\to 0^+} \tilde{c}(\vec{x} - \epsilon\vec{n}, t) \tag{14}$$

and

$$\tilde{c}^+ = \tilde{c}^+(\vec{x},t) \coloneqq \begin{cases} \tilde{c}^{\mathrm{bc}} & \mathcal{E}_{i,e} \subset \partial\Omega_h \\ \lim\limits_{\epsilon\to 0^+} \tilde{c}(\vec{x} + \epsilon\vec{n}, t) & \text{otherwise} \end{cases} \tag{15}$$

are the one-sided limits of the approximate solution on $\mathcal{E}_{i,e}$, where $\tilde{c}^+$ depends on the boundary value $\tilde{c}^{\mathrm{bc}} = \tilde{c}^{\mathrm{bc}}(\vec{x}, t, \tilde{c}^-)$ if $\mathcal{E}_{i,e}$ is a boundary edge.

The temporal term in (12) can be simplified to

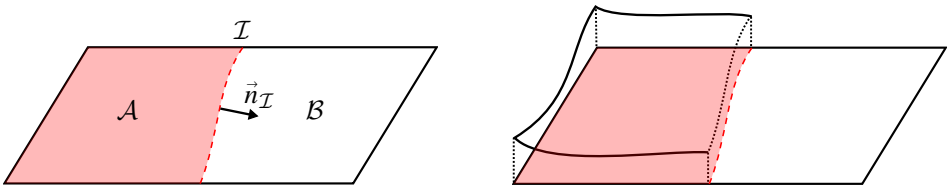
$$\int_{\mathcal{K}_i} \frac{\partial \tilde{c}_i}{\partial t} \Phi_{i,j}\, dV = \sum_{k=0}^{M} \frac{\partial \tilde{c}_{i,k}}{\partial t} \underbrace{\int_{\mathcal{K}_i} \Phi_{i,k}\Phi_{i,j}\, dV}_{=:(\mathbf{M}_i)_{k,j}} \tag{16}$$

where $\mathbf{M}_i \in \mathbb{R}^{M,M}$ denotes the cell-local, symmetric mass matrix of $\mathcal{K}_i$. It directly follows that the semi-discrete system can be summarized as

$$\mathbf{M}_i \frac{\partial \vec{\tilde{c}}_i}{\partial t} + \vec{f}_i = \vec{0}. \tag{17}$$

The structure of the mass matrix $\mathbf{M}_i$ clearly depends on the choice of the basis functions. We use a minimal orthonormal basis which implies that the mass matrix reduces to the identity matrix $\mathbf{I}$ in standard cells. This does not hold in cells intersected by the immersed boundary, though, which we will discuss in the following section.

DOI: 10.1002/nme



(a) Exemplary configuration in a cut-cell

(b) Visualuaziation of the polynomial approximation which is cut at $\mathcal{I}$

Figure 1. Illustration of the approximation in cut cells. The interface $\mathcal{I}$ (dashed red line) and its normal vector $\vec{n}_{\mathcal{I}}$ are implicitly defined by the level set function φ. Sub-domain $\mathcal{A}$ (light red) is associated with the fluid domain, while $\mathcal{B}$, is considered void.

3.2. A Discontinuous Galerkin scheme with immersed boundaries

Given an implicit representation of an immersed boundary by means of a level set function φ, we partition Ω_h into the physical region

$$\mathcal{A} = \{\vec{x} \in \Omega_h : \varphi(\vec{x}) > 0\}, \tag{18}$$

the void region

$$\mathcal{B} = \{\vec{x} \in \Omega_h : \varphi(\vec{x}) < 0\} \tag{19}$$

and the immersed boundary

$$\mathcal{I} = \{\vec{x} \in \Omega_h : \varphi(\vec{x}) = 0\}. \tag{20}$$

A natural extension of (17) can be formulated starting from the discrete weak formulation (10) by restricting the problem domain to the physical region. That is, the support of the basis/test functions $\vec{\Phi}_i$ as we all as the domains of integration in cell $\mathcal{K}_i$ are restricted to the sub-domains $\mathcal{A}_i = \mathcal{K}_i \cap \mathcal{A}$ and $\partial\mathcal{A}_i$ (cf. Figure 1), respectively, before performing the partial integration and inserting the numerical approximations. Taking into account that the surface $\partial\mathcal{A}_i$ consists of the edges $\{\mathcal{E}^{\mathcal{A}}_{i,e}\}_{e=1,\dots,E} = \{\mathcal{E}_{i,e} \cap \overline{\mathcal{A}}_i\}_{e=1,\dots,E}$ and the boundary segment $\mathcal{I}_i = \mathcal{K}_i \cap \mathcal{I}$, it directly follows that the discrete weak formulation in the context of an IBM can be written as

$$\begin{aligned} &\int_{\mathcal{A}_i} \frac{\partial \tilde{c}_i}{\partial t} \Phi_{i,j}\, dV + \sum_{e=1}^{E} \int_{\mathcal{E}^{\mathcal{A}}_{i,e}} f(\tilde{c}^-, \tilde{c}^+, \vec{n})\, \Phi_{i,j}\, dA \\ &+ \int_{\mathcal{I}_i} f(\tilde{c}^-, \tilde{c}^{\mathrm{bc}}, \vec{n}_{\mathcal{I}})\, \Phi_{i,j}\, dA - \int_{\mathcal{A}_i} \vec{f}(\tilde{c}_i) \cdot \nabla \Phi_{i,j}\, dV = 0, \end{aligned} \tag{21}$$

where $\vec{n}_{\mathcal{I}} = -\nabla\varphi / \|\nabla\varphi\|$. As a result, the semi-discrete system (17) in cells intersected by $\mathcal{I}$ is defined by

$$(\mathbf{M}_i)_{k,j} \coloneqq \int_{\mathcal{A}_i} \Phi_{i,k} \Phi_{i,j}\, dV \tag{22}$$

and

$$(\vec{f}_i)_j \coloneqq \sum_{e=1}^{E_i} \int_{\mathcal{E}^{\mathcal{A}}_{i,e}} f(\tilde{c}^-, \tilde{c}^+, \vec{n})\, \Phi_{i,j}\, dA + \int_{\mathcal{I}_i} f(\tilde{c}^-, \tilde{c}^+, \vec{n}_{\mathcal{I}})\, \Phi_{i,j}\, dA - \int_{\mathcal{A}_i} \vec{f}(\tilde{c}_i) \cdot \nabla \Phi_{i,j}\, dV \tag{23}$$

in all cells that are intersected by the boundary $\mathcal{I}$.

Even though this extension is rather straightforward from a mathematical point of view, the method strongly relies on the accurate and efficient evaluation of the integrals over $\mathcal{A}_i$ and $\mathcal{I}_i$.

DOI: 10.1002/nme

We will discuss this point in the Section 3.4. Moreover, the presence of intersected cells $\mathcal{K}_i$ with extremely small volume fractions

$$\text{frac}(\mathcal{A}_i) = \frac{\text{meas}(\mathcal{A}_i)}{\text{meas}(\mathcal{K}_i)} \tag{24}$$

inside of the physical problem domain can cause significant issues. We rectify these problems using the cell-agglomeration strategy outlined in Section 3.5.

3.3. Solution

The temporal discretization of (17) with $\vec{f}_i$ defined by (12) for standard cells and by (23) for cut cells can for example be carried out by virtue of standard Runga-Kutta schemes. Since we focus on steady-state problems within this work, we use a simple explicit Euler time discretization to drive (17) into a steady state. That is, we iterate

$$\vec{\tilde{c}}_i \leftarrow \vec{\tilde{c}}_i - \Delta t\, \mathbf{M}_i^{-1}\, \vec{f}_i(\tilde{c}) \tag{25}$$

with a pseudo time-step size Δt until a steady-state is reached.

Here, we have used the modified step size restriction

$$\Delta t \leq \frac{c_{\text{CFL}}}{2P+1} \frac{\sqrt[D]{\text{meas}(\mathcal{A}_i)}}{\|\vec{u}\| + a}. \tag{26}$$

which is strongly influenced by the (sub-)cell $\mathcal{A}_i$ with the smallest volume. We note that this formula does not take the actual shape of the sub-cells into account, which is why we would have to choose relatively low values for the constant c_{CFL}. However, this step-size restriction is alleviated significantly by the cell-agglomeration technique presented in Section 3.5, which is sufficient for the purposes of this work. However, this point requires more attention for time-dependent calculations, which will be discussed in a follow-up paper.

3.4. Numerical integration

Due to the increasing popularity of eXtended Finite Element Method (XFEM) [27] and related sharp-interface approaches, the numerical integration over implicitly defined sub-domains has received significant attention in recent years. Most approaches presented in literature deliver 2^{nd} order convergence rates in the presence of curved interfaces [1, 28–34], and are thus often supplemented by a recursive subdivision strategy in order to retain the formal convergence order of the underlying discretization scheme.

Recently, methods that aim at overcoming the need for (an excessive number of) subdivisions have gained more and more interest, especially in the context of the higher order XFEM [18, 21, 22] the Finite Cell Method (FCM) [33] and eXtended/unfitted DG methods [14, 17]. The majority of these approaches is based on the idea of either constructing a higher order approximation of the interface [21, 35, 36] or computing a mapping function that improves its piecewise linear approximation [22]. A different group of methods is based on reducing the dimension of the integrals to be computed using the idea of height functions [14, 37].

Yet another class of methods is based on solving the moment-fitting equations [38] which has originally been used for the pre-calculation of highly efficient, Gauss-like quadrature rules for arbitrary polyhedra. Later, the method has been adapted to problems involving piecewise linear *moving* interfaces [39, 40]. Here, we make use of an extension to curved interfaces [41], which we will briefly outline in the following.

Figure 2

BM: Finish quadrature section

DOI: 10.1002/nme

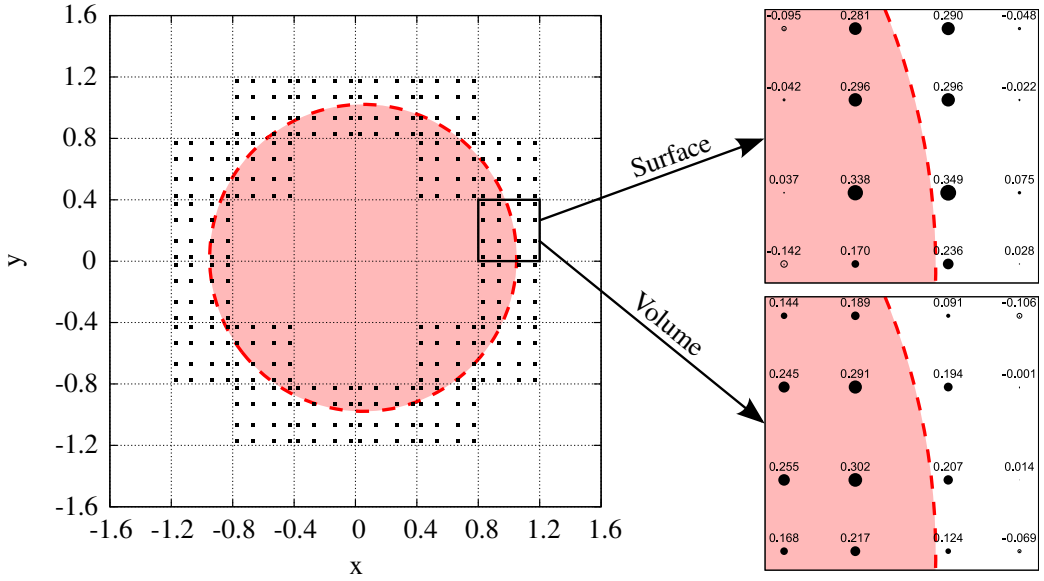
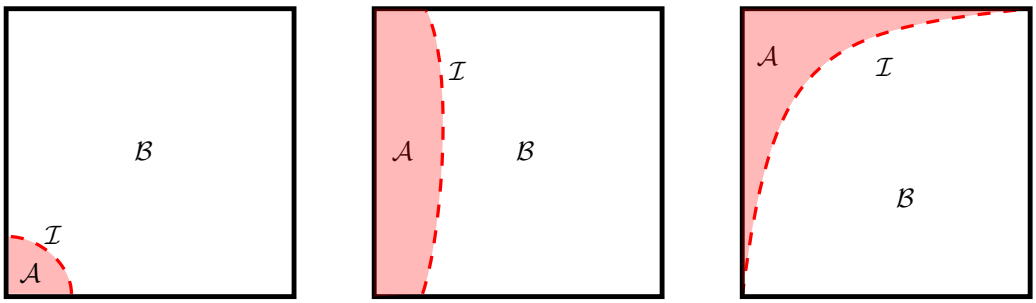


Figure 2. Quadrature nodes and weights for the example of a unit circle (light red) represented by the zero iso-contour (dashed red line) of the level set function $\varphi = x^2 + y^2 - 1$. The depicted weights have been obtained by means of the HMF using moments up to order 2. Note that we have chosen the same nodes in the volume and in the surface case for illustrative purposes. In practice, it suffices to use *less* nodes in the volume case to obtain 3$^{\text{rd}}$ order convergence for this case.



(a) Sub-cell with a small volume fraction

(b) Sliver sub-cell

(c) Strongly curved sub-cell

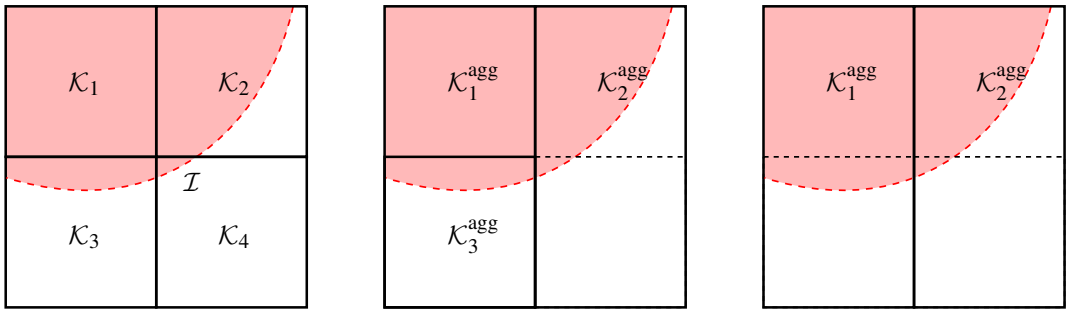
Figure 3. Configurations where cell-agglomeration is required for higher order simulations.

3.5. *Cell-agglomeration*

Up to this point, the discretization outlined in Section 26 admits arbitrarily small cut cells with arbitrary shapes. Obviously, this renders the method impractical due to the extreme time-step restriction in cells with small volume fractions (cf. Figure 3a). Moreover, it should be noted that higher order methods are strongly affected by the presence of so-called sliver elements. In Figure 3b, we give an example for a cell $\mathcal{K}$ with a volume fraction of about $\text{frac}(\mathcal{K}) \approx 0.2$. While such cells are hardly problematic for low-order simulations ($P \leq 1$), a higher-order basis $\vec{\Phi}$ defined on $\mathcal{K}$ is strongly ill-conditioned on $\mathcal{A}$ since the basis functions lose their linear independence. A simple remedy for this problem is the definition of the basis on the bounding box of $\mathcal{A}$, which has for example been proposed in [1][†] for $P \leq 2$. Unfortunately, this approach still fails if the immersed boundary is strongly curved. In Figure 3c, we give an example for a cell with approximately the

[†]We note that the authors also use a cell-agglomeration strategy which similar to the one we are using in this work

DOI: 10.1002/nme



(a) Initial mesh (b) Agglomerated mesh, α small (c) Agglomerated mesh, α large

Figure 4. Illustration of the cut-cell agglomeration strategy. For small values of α (middle), only cell $\mathcal{K}_4$ will agglomerated to the neighbor *direct* neighbor with the largest volume fraction (cell $\mathcal{K}_2$). If α is increased (right), $\mathcal{K}_3$ will additionally be agglomerated to cell $\mathcal{K}_1$.

same volume fraction as in the second example (Figure 3b), but where the bounding box of $\mathcal{A}$ coincides with $\partial\mathcal{K}$. As a result, using a rescaling of the basis to the bounding box of $\mathcal{A}$ is insufficient when higher approximation orders ($P \geq 3$) are considered.

In order to overcome these issues, we employ the cell-agglomeration strategy proposed in [17] which we will outline briefly in the following, before discussing the details of its implementation in Section 3.5.2 and assessing its effect on the discretization in Section 3.5.3.

3.5.1. Agglomeration strategy In the first step of the algorithm, we determine the list of agglomeration *source* cells $\{\mathcal{K}_s^{\text{src}}\}_{s=1,\dots,S} = \{\mathcal{K}_i : \text{frac}(\mathcal{K}_i) \leq \alpha\}$ with a user-defined threshold α ($0 \leq \alpha < 1$). Second, we determine the agglomeration *target* cell $\mathcal{K}_s^{\text{tar}}$ for each source cell $\mathcal{K}_s^{\text{src}}$ by finding the edge neighbor $\{\mathcal{N}_{s,e}\}_{e=1,\dots,E} = \{\mathcal{K}_s \cap \mathcal{K}_i = \mathcal{E}_{s,e}\}$ with the largest volume fraction, i.e.

$$\mathcal{K}_s^{\text{tar}} = \underset{\mathcal{N}_{s,e}}{\text{argmax}}\, \text{frac}(\mathcal{N}_{s,e}). \tag{27}$$

A major advantage of the DG method is the weak coupling of neighboring cells via fluxes. As a result, each basis $\vec{\Phi}_i$ can be chosen independently, which simply allows us to extend the basis of the target cell $\mathcal{K}_s^{\text{tar}}$ into the agglomeration source cell $\mathcal{K}_s^{\text{src}}$. The source cell is then formally eliminated from the discretization, thus defining the new computational mesh $\{\mathcal{K}_i^{\text{agg}}\}_{i=1,\dots,N-S}$. This does however not reflect our actual implementation (cf. Section 3.5.2), which exploits the locality of the involved operations and hence only requires few additional *cell-local* matrix-vector products per time-step that do not affect the parallel efficiency.

The cell-agglomeration process is illustrated in Figure 4 using the example of the depicted in Figure 4a which initially consists of 4 cells. Depending on the selected threshold α, the resulting mesh consists of 3 (Figure 4b) or 2 cells (Figure 4c). Here, it is worth noting that cell $\mathcal{K}_4$ is not agglomerated to cell $\mathcal{K}_1$ since they do not share a common edge. However, further increasing α would lead to a situation where $\mathcal{K}_2^{\text{agg}}$ from Figure 4c would in turn be agglomerated to $\mathcal{K}_1^{\text{agg}}$, which would indeed link cells $\mathcal{K}_4$ to cell $\mathcal{K}_1$. While this case can be handled easily via a recursive strategy, we did not encounter such a situation during our numerical experiments presented in Section 26 which is why we abstain from discussing the details here.

3.5.2. Implementation For the discussion of the implementation, we limit ourselves a single agglomeration pair consisting of the source $\mathcal{K}^{\text{src}}$ and the target cell $\mathcal{K}^{\text{tar}}$ that is merged to form $\mathcal{K}^{\text{agg}}$. Let $\vec{\Phi}^{\text{tar}} = (\vec{\Phi}_1^{\text{tar}}, \dots, \vec{\Phi}_M^{\text{tar}})$ and $\vec{\Phi}^{\text{src}} = (\vec{\Phi}_1^{\text{src}}, \dots, \vec{\Phi}_M^{\text{src}})$ be the row vectors of basis functions of $\mathcal{K}^{\text{tar}}$ and $\mathcal{K}^{\text{src}}$, respectively. Note that the individual basis functions are non-zero in the corresponding cells only. We then construct a coupling matrix $\mathbf{Q} \subset \mathbb{R}^{M,M}$ such that $\vec{\Phi}^{\text{src}}\mathbf{Q}$ is the smooth extension

DOI: 10.1002/nme

of $\vec{\Phi}^{\text{tar}}$ into $\mathcal{K}^{\text{src}}$. The basis of $\mathcal{K}^{\text{agg}}$ is then defined as

$$\vec{\Phi}^{\text{agg}} = \vec{\Phi}^{\text{tar}} + \vec{\Phi}^{\text{src}}\mathbf{Q}, \tag{28}$$

which is computed once at the beginning of a simulation.

In order to obtain $\mathbf{Q}$, we project the smooth extension of the basis functions $\vec{\Phi}^{\text{tar}}$ into $\mathcal{K}^{\text{src}}$, denoted by $\underline{\vec{\Phi}}^{\text{tar}}$, onto $\mathcal{K}^{\text{src}}$. That is, we compute

$$\mathbf{Q} = \begin{pmatrix} \int\limits_{\mathcal{K}^{\text{src}}} \underline{\vec{\Phi}}_1^{\text{tar}} \vec{\Phi}_1^{\text{src}}\, dV & \dots & \int\limits_{\mathcal{K}^{\text{src}}} \underline{\vec{\Phi}}_M^{\text{tar}} \vec{\Phi}_1^{\text{src}}\, dV \\ \vdots & & \vdots \\ \int\limits_{\mathcal{K}^{\text{src}}} \underline{\vec{\Phi}}_1^{\text{tar}} \vec{\Phi}_M^{\text{src}}\, dV & \dots & \int\limits_{\mathcal{K}^{\text{src}}} \underline{\vec{\Phi}}_M^{\text{tar}} \vec{\Phi}_M^{\text{src}}\, dV \end{pmatrix} \tag{29}$$

from which it directly follows that

$$\underline{\vec{\Phi}}^{\text{tar}} = \vec{\Phi}^{\text{src}}\mathbf{Q}. \tag{30}$$

The benefits of definition (28) are two-fold. Firstly, it greatly simplifies the extension of an existing implementation of a DG scheme because all operators defined in Section 3.1 and Section 3.2 can be reused without changing the underlying data structures. We will outline this process in the following. Secondly, the introduction of $\mathbf{Q}$ greatly simplifies the handling of dynamic (de-)agglomeration of cells in transient calculations with moving boundaries. The second point is out of scope of this work and we will thus not discuss it here.

In the setup phase of the computation, we need to compute the mass matrix $\mathbf{M}^{\text{agg}}$ as well as the initial condition for the agglomerated cell. It is easy to verify that these quantities can easily be computed from the respective quantities defined on the original grid (cf. Appendix A). The mass matrix follows from

$$\mathbf{M}^{\text{agg}} = \mathbf{M}^{\text{tar}} + \mathbf{Q}^{\text{T}}\mathbf{M}^{\text{src}}\mathbf{Q}, \tag{31}$$

which allows us to compute the projection of the initial condition via

$$\vec{c}^{\,\text{agg}} = (\mathbf{M}^{\text{agg}})^{-1}(\mathbf{M}^{\text{tar}}\vec{c}^{\,\text{tar}} + \mathbf{Q}^{\text{T}}\mathbf{M}^{\text{src}}\vec{c}^{\,\text{src}}). \tag{32}$$

After this projection, the solution stays continuous within $\mathcal{K}^{\text{agg}}$ for all times. As a result, injecting the agglomerated solution $\vec{c}^{\,\text{agg}}$ into the original mesh via

$$\vec{c}^{\,\text{tar}} = \vec{c}^{\,\text{agg}} \tag{33}$$

and

$$\vec{c}^{\,\text{src}} = \mathbf{Q}\vec{c}^{\,\text{agg}} \tag{34}$$

leads to an equivalent problem formulation. The updated solution is thus computed by means of

$$\vec{c}^{\,\text{tar}} \leftarrow \vec{c}^{\,\text{tar}} - \Delta t\,(\mathbf{M}^{\text{agg}})^{-1}\,\vec{f}^{\,\text{agg}} \tag{35}$$

$$\vec{c}^{\,\text{src}} \leftarrow \mathbf{Q}\vec{c}^{\,\text{tar}}. \tag{36}$$

Here, the flux vector $\vec{f}^{\,\text{agg}}$ can be computed from the standard fluxes $\vec{f}^{\,\text{tar}}$ and $\vec{f}^{\,\text{src}}$ for $\mathcal{K}^{\text{tar}}$ and $\mathcal{K}^{\text{src}}$ by means of

$$\vec{f}^{\,\text{agg}} = \vec{f}^{\,\text{tar}} + \mathbf{Q}^{\text{T}}\,\vec{f}^{\,\text{src}}. \tag{37}$$

All in all, the variable update using cell-agglomeration only requires two additional, cell-local matrix-vector products per time-step, which allows for a simple extension of an existing DG scheme.

3.5.3. Influence of the agglomeration threshold Empty

BM: Add test for condition number vs approximation quality when varying α

DOI: 10.1002/nme

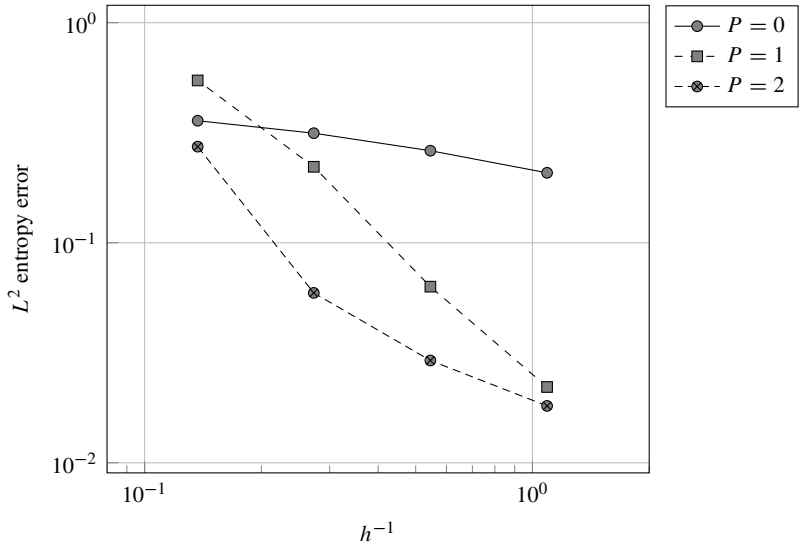
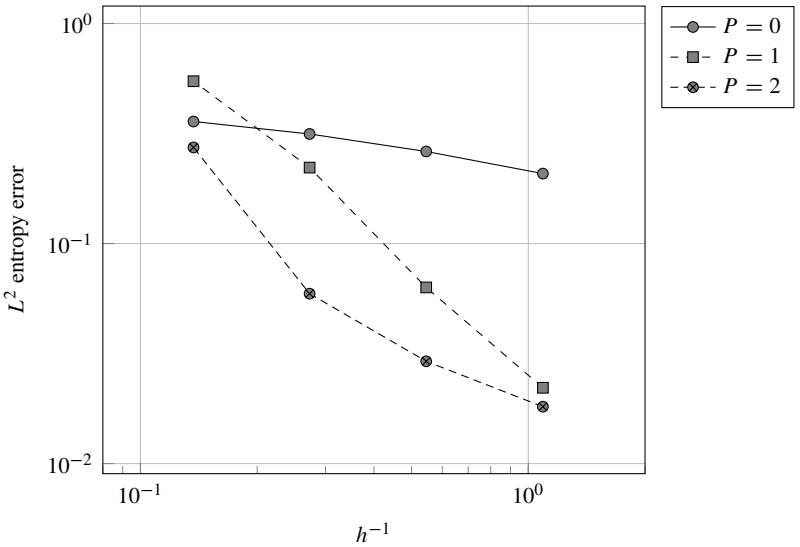


Figure 5. To do: Study mass matrix conditioning as a function of α



P	EOC
0	0.26
1	1.57
2	1.28

Figure 6. Results of the h-convergence study for the flow around cylinder using an IBM based on HMF(8)

4. NUMERICAL RESULTS

Applied boundary conditions: Slip wall [42] Wall [26] and free-stream (Dirichlet)
experimental order of convergence (EOC)

4.1. Inviscid flow around a cylinder

Study the influence of cell-agglomeration

$$\varphi(\vec{x}) = x^2 + y^2 - 1 \tag{38}$$

BM: Add description and results for cylinder flow

Figure 7
Figure 8

DOI: 10.1002/nme

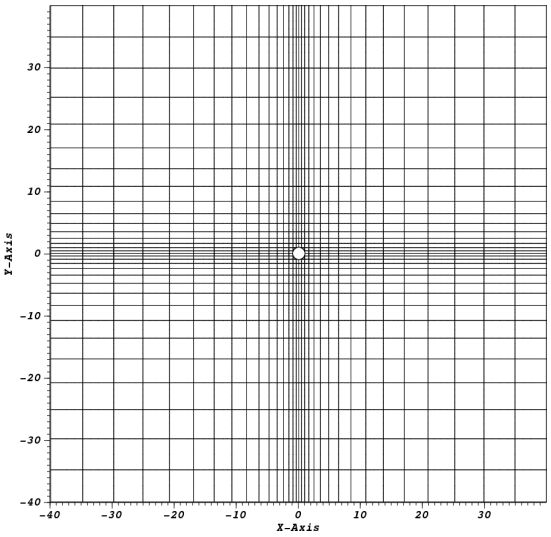
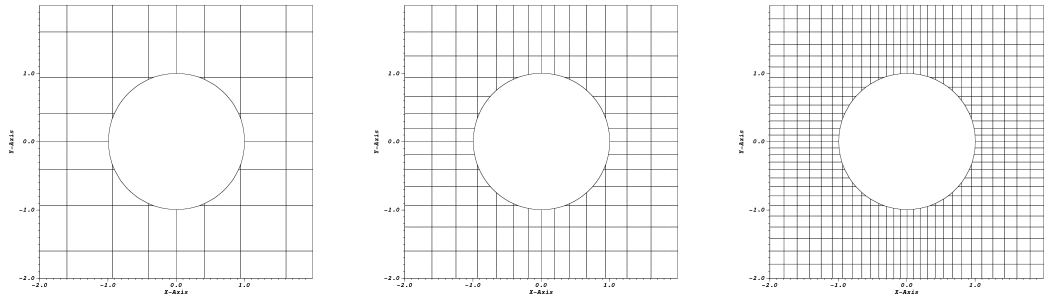


Figure 7. Coarsest mesh for the flow around a cyinder consisting of 32 × 32 rectangular cells. Note that the cylinder has been added during post-processing for illustrative purposes.



(a) 32 × 32 cells (b) 64 × 64 cells (c) 128 × 128 cells

Figure 8. Zoom into the region around the cylinder on the three coarsest grids. The smallest cut-cells have a volume fraction of about 5 % in all cases. Note that the cylinder has been added during post-processing for illustrative purposes.

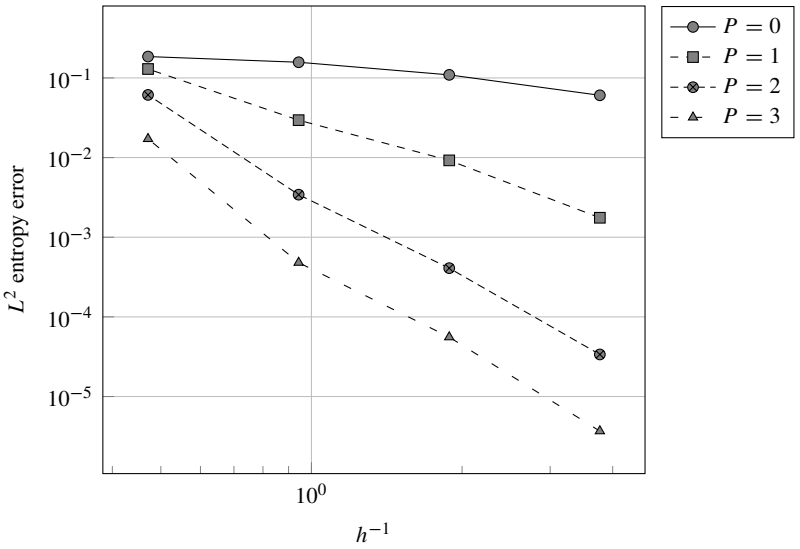
4.2. Inviscid flow over a Gaussian bump

In this test case, the flow over a smooth bump is investigated. It is used to study the robustness of the scheme w.r.t movements of the level set. The entropy s is constant in the flow field and the L_2-norm of the entropy error is then used as an indicator of solution accuracy since the analytical solution is unknown. The smooth bump is given by the following level set equation

$$\varphi(\vec{x}) = (y - \epsilon_2) - 0.01 + e^{-0.5(x-\epsilon_1)^2}, \tag{39}$$

where ϵ_1 and ϵ_2 controls the level-set movements in x- and y-direction, respectively. Due to the fact that a high order boundary representation is needed for accurate results, the level set field order is set to $P_{\mathrm{LS}} = 8$. The computational domain is a $[-12, 12] \times [0, 12]$ box with a uniform Cartesian grid. Slip wall boundary conditions are applied at the smooth bump, everywhere else uniform free stream conditions with $\mathrm{M}_\infty = 0.5$. In Figure 9, we present a comparison of the error in the L_2-norm under a uniform grid refinement for $P = 0, \ldots, 4$ without any movement of the level-set function, i.e $\epsilon_1 = \epsilon_2 = 0$. We call this the base case. For all degree P, we observe that the error converges towards zero with the expected rate $\mathcal{O}\left(h^{P+1}\right)$.

DOI: 10.1002/nme



P	EOC
0	0.54
1	2.02
2	3.55
3	3.96

Figure 9. Results of the h-convergence study for the flow over a smooth bump with $\epsilon_1 = \epsilon_2 = 0$

4.3. *Inviscid flow over a NACA0012 profile*

In this example, we consider the subsonic inviscid flow around a NACA0012 airfoil, where the upper and lower surface of the airfoil is specified by

$$y = \pm 0.6 \left(0.2969\sqrt{x} - 0.126x - 0.3516x^2 + 0.2843x^3 - 0.1306x^4\right). \tag{40}$$

To obtain a purely polynomial level set function, equation (40) is reordered and squared which yields to two level set equations

$$\varphi(\vec{x}) = \left(\pm y + 0.6\left(0.126x^2 + 0.3516x^3 - 0.2843x^3 + 0.1306x^4\right)\right)^2 - 0.03173x, \tag{41}$$

for the upper (+) and lower (−) part. According to this level set equation, we set the order of the level set field to $P_{LS} = 8$. The bounding box is 100 chord length away from the airfoil. The free stream conditions are set to $\mathrm{M}_\infty = 0.5$. Figure 40 shows a zoom into the airfoil. We specify a refined region around the airfoil. Traditionally, the angle of attack α is specified by rotating the free stream conditions by the value of α. Here, we keep the free stream conditions the same, but rotate the level set field to demonstrate the geometrical flexibility of the IBM scheme. Again, the L_2-norm of the entropy error is used as an indicator of solution accuracy. Additionally, the lift c_L and drag c_D coefficients are compared to the values of [**?**].

Study efficiency in terms of influence of HMF

SKE: Add description and results for NACA (inviscid)

4.4. *Viscous flow over a NACA0012 profile*

Show that viscosity also works; steady or unsteady?

SKE: Add description and results for NACA (viscous)

5. CONCLUSION

Curved elements are dead.

BM: Add conclusion

DOI: 10.1002/nme

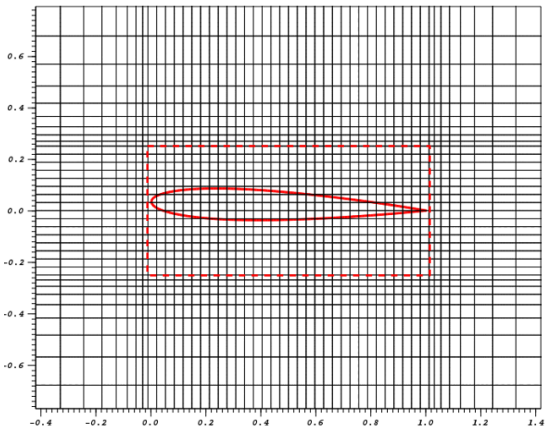


Figure 10. Zoom into the region around the NACA0012 airfoil for a medium mesh: Solid red line shows the zero contour of the level set function (41) for an airfoil with $\alpha = 2^\circ$. The dashed red line indictaed the refined region.

A. CELL-AGGLOMERATION ALGEBRA

In this section, we derive the relations following from definition (28) that have used in Section 3.5.2. Within this section, we use the Einstein index summation.

Equation (31) for mass matrix of the agglomerated cell $\mathbf{M}^{\text{agg}}$ follows from

$$(\mathbf{M}^{\text{agg}})_{ij} = \int_{\mathcal{K}^{\text{agg}}} (\vec{\Phi}^{\text{agg}})_i (\vec{\Phi}^{\text{agg}})_j \, dV \tag{42}$$

$$= \int_{\mathcal{K}^{\text{tar}}} (\vec{\Phi}^{\text{tar}})_i (\vec{\Phi}^{\text{tar}})_j \, dV + \int_{\mathcal{K}^{\text{src}}} (\vec{\Phi}^{\text{src}}\mathbf{Q})_i (\vec{\Phi}^{\text{src}}\mathbf{Q})_j \, dV \tag{43}$$

$$= (\mathbf{M}^{\text{tar}})_{ij} + \int_{\mathcal{K}^{\text{src}}} (\vec{\Phi}^{\text{src}})_k (\mathbf{Q})_{ki} (\vec{\Phi}^{\text{src}})_l (\mathbf{Q})_{lj} \, dV \tag{44}$$

$$= (\mathbf{M}^{\text{tar}})_{ij} + (\mathbf{Q})_{ki} \left(\int_{\mathcal{K}^{\text{src}}} (\vec{\Phi}^{\text{src}})_k (\vec{\Phi}^{\text{src}})_l \, dV \right) (\mathbf{Q})_{lj} \tag{45}$$

$$= (\mathbf{M}^{\text{tar}})_{ij} + (\mathbf{Q}^{\text{T}})_{ik} (\mathbf{M}^{\text{src}})_{kl} (\mathbf{Q})_{lj}. \tag{46}$$

BM: Flux agglomeration?

Equation (32) for the agglomerated solution follows from the orthogonality condition

$$\int_{\mathcal{K}^{\text{agg}}} (\tilde{c} - c^{\text{agg}})(\vec{\Phi}^{\text{agg}})_i \, dV \stackrel{!}{=} 0 \tag{47}$$

DOI: 10.1002/nme

which leads to

$$(\mathbf{M}^{\text{agg}}\vec{c}^{\text{agg}})_i = \int_{\mathcal{K}^{\text{tar}}} c^{\text{tar}}(\vec{\Phi}^{\text{tar}})_i \, dV + \int_{\mathcal{K}^{\text{src}}} c^{\text{src}}(\vec{\Phi}^{\text{src}}\mathbf{Q})_i \, dV \tag{48}$$

$$= (\vec{c}^{\text{tar}})_j \int_{\mathcal{K}^{\text{tar}}} (\vec{\Phi}^{\text{tar}})_j (\vec{\Phi}^{\text{tar}})_i \, dV + (\vec{c}^{\text{src}})_j \int_{\mathcal{K}^{\text{src}}} (\vec{\Phi}^{\text{src}})_j (\vec{\Phi}^{\text{src}}\mathbf{Q})_i \, dV \tag{49}$$

$$= (\vec{c}^{\text{tar}})_j (\mathbf{M}^{\text{tar}})_{ji} + (\vec{c}^{\text{src}})_j \int_{\mathcal{K}^{\text{src}}} (\vec{\Phi}^{\text{src}})_j (\vec{\Phi}^{\text{src}})_k (\mathbf{Q})_{ki} \, dV \tag{50}$$

$$= (\vec{c}^{\text{tar}})_j (\mathbf{M}^{\text{tar}})_{ji} + (\vec{c}^{\text{src}})_j (\mathbf{M}^{\text{src}})_{jk} (\mathbf{Q})_{ki} \tag{51}$$

$$= (\mathbf{M}^{\text{tar}})_{ij} (\vec{c}^{\text{tar}})_j + (\mathbf{Q}^{\text{T}})_{ik} (\mathbf{M}^{\text{src}})_{kj} (\vec{c}^{\text{src}})_j, \tag{52}$$

where the last line exploits the symmetry of $\mathbf{M}^{\text{src}}$.

BM: Make indices conformal, i.e. avoid using i (reserved for cells), introduce Einstein convention and use commans between indices

Equations (33) and (34) for the injection (?) of the agglomerated solution into the non-agglomerated space follow from the orthogonality conditions

$$\int_{\mathcal{K}^{\text{tar}}} (c^{\text{agg}} - c^{\text{tar}})(\vec{\Phi}^{\text{tar}})_i \, dV \stackrel{!}{=} 0 \tag{53}$$

and

$$\int_{\mathcal{K}^{\text{src}}} (c^{\text{agg}} - c^{\text{src}})(\vec{\Phi}^{\text{src}})_i \, dV \stackrel{!}{=} 0, \tag{54}$$

respectively. The first condition simply leads to

$$\vec{c}^{\text{tar}} = \vec{c}^{\text{agg}}, \tag{55}$$

while the second condition yields

$$(\mathbf{M}^{\text{src}}\vec{c}^{\text{src}})_i = \int_{\mathcal{K}^{\text{src}}} c^{\text{agg}}(\vec{\Phi}^{\text{src}})_i \, dV \tag{56}$$

$$= \int_{\mathcal{K}^{\text{src}}} (\vec{\Phi}^{\text{agg}})_j (\vec{c}^{\text{agg}})_j (\vec{\Phi}^{\text{src}})_i \, dV \tag{57}$$

$$= \int_{\mathcal{K}^{\text{src}}} (\vec{\Phi}^{\text{src}}\mathbf{Q})_j (\vec{c}^{\text{agg}})_j (\vec{\Phi}^{\text{src}})_i \, dV \tag{58}$$

$$= \int_{\mathcal{K}^{\text{src}}} (\vec{\Phi}^{\text{src}})_k (\mathbf{Q})_{kj} (\vec{c}^{\text{agg}})_j (\vec{\Phi}^{\text{src}})_i \, dV \tag{59}$$

$$= \left(\int_{\mathcal{K}^{\text{src}}} (\vec{\Phi}^{\text{src}})_i (\vec{\Phi}^{\text{src}})_k \, dV \right) (\mathbf{Q})_{kj} (\vec{c}^{\text{agg}})_j \tag{60}$$

$$= (\mathbf{M}^{\text{src}})_{ik} (\mathbf{Q})_{kj} (\vec{c}^{\text{agg}})_j. \tag{61}$$

DOI: 10.1002/nme

B. CELL-AGGLOMERATION EXAMPLE

For simplicity, consider a one-dimensional domain $\Omega = [-1, 3]$ without any immersed boundaries, discretized using two cells, $\mathcal{K}_1 = [-1, 1]$ and $\mathcal{K}_2 = [1, 3]$, and 2^{nd} order polynomials. Let the basis in $\mathcal{K}_1$ be given by

$$\Phi_{1,1} = 1|_{\mathcal{K}_1}, \quad \Phi_{1,2} = x|_{\mathcal{K}_1}, \quad \Phi_{1,3} = x^2\Big|_{\mathcal{K}_1} \tag{62}$$

and the basis in $\mathcal{K}_2$

$$\Phi_{2,1} = 1|_{\mathcal{K}_2}, \quad \Phi_{2,2} = x - 2|_{\mathcal{K}_2}, \quad \Phi_{2,3} = (x-2)^2\Big|_{\mathcal{K}_2}. \tag{63}$$

Obviously this basis is neither orthonormal nor numerically favorable, but this is irrelevant for this example. Assume furthermore that $\mathcal{K}_2$ shall be agglomerated to cell $\mathcal{K}_1$. Then, the basis in the agglomerated cell $\mathcal{K}^{\text{agg}}$ is given by

$$\Phi_{1,1} = 1|_{\mathcal{K}_1\cap\mathcal{K}_2}, \quad \Phi_{1,2} = x|_{\mathcal{K}_1\cap\mathcal{K}_2}, \quad \Phi_{1,3} = x^2\Big|_{\mathcal{K}_1\cap\mathcal{K}_2} \tag{64}$$

BM: Finish cell-agglomeration example?

References

1. Qin R, Krivodonova L. A discontinuous galerkin method for solutions of the euler equations on cartesian grids with embedded geometries. *Journal of Computational Science* Jan 2013; **4**(1-2):24–35, doi:10.1016/j.jocs.2012.03.008.
2. Burman E, Claus S, Hansbo P, Larson MG, Massing A. CutFEM: discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering* Dec 2014; :n/a–n/adoi:10.1002/nme.4823.
3. Fries T, Belytschko T. The extended/generalized finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering* Aug 2010; :253–304doi:10.1002/nme.2914.
4. Mittal R, Iaccarino G. Immersed boundary methods. *Annual Review of Fluid Mechanics* 2005; **37**:239–261, doi:doi:10.1146/annurev.fluid.37.061903.175743.
5. Lew AJ, Buscaglia GC. A discontinuous-Galerkin-based immersed boundary method. *International Journal for Numerical Methods in Engineering* Oct 2008; **76**(4):427–454, doi:10.1002/nme.2312.
6. Liu J, Qiu J, Hu O, Zhao N, Goman M, Li X. Adaptive Runge-Kutta discontinuous galerkin method for complex geometry problems on cartesian grid. *International Journal for Numerical Methods in Fluids* Jul 2013; :n/a–n/adoi:10.1002/fld.3825.
7. Fechter S, Munz C. A discontinuous galerkin-based sharp-interface method to simulate three-dimensional compressible two-phase flow. *International Journal for Numerical Methods in Fluids* Mar 2015; doi:10.1002/fld.4022.
8. Ferrari A, Munz C, Weigand B. A high order Sharp-Interface method with local time stepping for compressible multiphase flows. *Communications in Computational Physics* 2010; doi:10.4208/cicp.090310.050510a.
9. Marchandise E, Remacle J. A stabilized finite element method using a discontinuous level set approach for solving two phase incompressible flows. *Journal of Computational Physics* Dec 2006; **219**(2):780–800, doi:10.1016/j.jcp.2006.04.015.
10. Pochet F, Hillewaert K, Geuzaine P, Remacle J, Marchandise E. A 3D strongly coupled implicit discontinuous galerkin level set-based method for modeling two-phase flows. *Computers & Fluids* 2013; doi:10.1016/j.compfluid.2013.04.010.
11. Fidkowski KJ, Darmofal DL. A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier–Stokes equations. *Journal of Computational Physics* Aug 2007; **225**(2):1653–1672, doi:10.1016/j.jcp.2007.02.007.
12. Modisette JM, Darmofal DL. Toward a robust, Higher-Order Cut-Cell method for viscous flows. Orlando, Florida, 2010.
13. Brandstetter G, Govindjee S. A high-order immersed boundary discontinuous-Galerkin method for poisson's equation with discontinuous coefficients and singular sources. *International Journal for Numerical Methods in Engineering* Dec 2014; doi:10.1002/nme.4835.
14. Saye RI. High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles. *SIAM Journal on Scientific Computing* 2015; .
15. Bastian P, Engwer C. An unfitted finite element method using discontinuous galerkin. *International Journal for Numerical Methods in Engineering* Sep 2009; **79**(12):1557–1576, doi:10.1002/nme.2631.
16. Heimann F, Engwer C, Ippisch O, Bastian P. An unfitted interior penalty discontinuous galerkin method for incompressible Navier-Stokes two-phase flow. *International Journal for Numerical Methods in Fluids* Jan 2013; **71**(3):269—293, doi:10.1002/fld.3653.
17. Kummer F. Extended discontinuous galerkin methods for two-phase flows the spatial discretization. *International Journal for Numerical Methods in Engineering* 2015; .

18. Legrain G, Chevaugeon N, Dréau K. High order X-FEM and levelsets for complex microstructures: Uncoupling geometry and approximation. *Computer Methods in Applied Mechanics and Engineering* 2012; **241–244**(0):172–189, doi:10.1016/j.cma.2012.06.001.
19. Annavarapu C, Hautefeuille M, Dolbow JE. Stable imposition of stiff constraints in explicit dynamics for embedded finite element methods: Embedded constraints in explicit dynamics. *International Journal for Numerical Methods in Engineering* Oct 2012; **92**(2):206–228, doi:10.1002/nme.4343.
20. Johansson A, Larson MG. A high order discontinuous galerkin nitsche method for elliptic problems with fictitious boundary. *Numerische Mathematik* Sep 2012; **123**(4):607–628, doi:10.1007/s00211-012-0497-1.
21. Fries T, Omerović S. Higher-order accurate integration of implicit geometries. *International Journal for Numerical Methods in Engineering* 2015; :n/a–n/adoi:10.1002/nme.5121.
22. Lehrenfeld C. High order unfitted finite element methods on level set domains using isoparametric mappings. *Computer Methods in Applied Mechanics and Engineering* 2015; **Preprint**.
23. Massing A, Larson MG, Logg A, Rognes ME. A stabilized nitsche overlapping mesh method for the stokes problem. *Numerische Mathematik* 2014; **128**(1):73–101, doi:10.1007/s00211-013-0603-z.
24. Massing A, Larson MG, Logg A, Rognes ME. A stabilized nitsche fictitious domain method for the stokes problem. *Journal of Scientific Computing* 2014; **61**(3):604—628, doi:10.1007/s10915-014-9838-9.
25. Brehm C, Hader C, Fasel H. A locally stabilized immersed boundary method for the compressible Navier–Stokes equations. *Journal of Computational Physics* Aug 2015; **295**:475–504, doi:10.1016/j.jcp.2015.04.023.
26. Hartmann R, Houston P. Symmetric interior penalty DG methods for the compressible Navier-Stokes equations i: Method formulation. *International Journal of Numerical Analysis and Modeling* Jul 2005; .
27. Moës N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering* Sep 1999; **46**(1):131–150, doi:10.1002/(SICI)1097-0207(19990910)46:1⟨131::AID-NME726⟩3.0.CO;2-J.
28. Smereka P. The numerical approximation of a delta function with application to level set methods. *Journal of Computational Physics* Jan 2006; **211**:77–90, doi:10.1016/j.jcp.2005.05.005.
29. Ventura G. On the elimination of quadrature subcells for discontinuous functions in the eXtended Finite-Element method. *International Journal for Numerical Methods in Engineering* Apr 2006; **66**(5):761–795, doi:10.1002/nme.1570.
30. Min C, Gibou F. Geometric integration over irregular domains with application to level-set methods. *Journal of Computational Physics* 2007; **226**(2):1432–1443, doi:16/j.jcp.2007.05.032.
31. Zahedi S, Tornberg A. Delta function approximations in level set methods by distance function extension. *Journal of Computational Physics* 2010; **229**(6):2199–2219, doi:10.1016/j.jcp.2009.11.030.
32. Müller B, Kummer F, Oberlack M, Wang Y. Simple multidimensional integration of discontinuous functions with application to level set methods. *International Journal for Numerical Methods in Engineering* 2012; **92**(7):637–651, doi:10.1002/nme.4353.
33. Sudhakar Y, Moitinho de Almeida J, Wall WA. An accurate, robust, and easy-to-implement method for integration over arbitrary polyhedra: Application to embedded interface methods. *Journal of Computational Physics* May 2014; doi:10.1016/j.jcp.2014.05.019.
34. Ventura G, Benvenuti E. Equivalent polynomials for quadrature in heaviside function enriched elements. *International Journal for Numerical Methods in Engineering* May 2014; :n/a–n/adoi:10.1002/nme.4679.
35. Cheng KW, Fries T. Higher-order XFEM for curved strong and weak discontinuities. *International Journal for Numerical Methods in Engineering* 2010; **82**(5):564–590, doi:10.1002/nme.2768.
36. Kudela L, Zander N, Bog T, Kollmannsberger S, Rank E. Efficient and accurate numerical quadrature for immersed boundary methods. *Advanced Modeling and Simulation in Engineering Sciences* Dec 2015; **2**(1), doi:10.1186/s40323-015-0031-y.
37. Wen X. High order numerical methods to two dimensional delta function integrals in level set methods. *Journal of Computational Physics* Jun 2009; **228**(11):4273–4290, doi:10.1016/j.jcp.2009.03.004.
38. Bremer J, Gimbutas Z, Rokhlin V. A nonlinear optimization procedure for generalized gaussian quadratures. *SIAM Journal on Scientific Computing* 2010; **32**(4):1761–1788, doi:10.1137/080737046.
39. Mousavi SE, Sukumar N. Generalized gaussian quadrature rules for discontinuities and crack singularities in the extended finite element method. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(49–52):3237–3249, doi:10.1016/j.cma.2010.06.031.
40. Mousavi SE, Sukumar N. Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons. *Computational Mechanics* 2011; **47**(5):535–554, doi:10.1007/s00466-010-0562-5.
41. Müller B, Kummer F, Oberlack M. Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *International Journal for Numerical Methods in Engineering* Nov 2013; **96**(8):512–528, doi:10.1002/nme.4569.
42. Bassi F, Bartolo CD, Hartmann R, Nigro A. A discontinuous galerkin method for inviscid low mach number flows. *Journal of Computational Physics* 2009; **228**(11):3996–4011.
43. Wang Z, Fidkowski K, Abgrall R, Bassi F, Caraeni D, Cary A, Deconinck H, Hartmann R, Hillewaert K, Huynh H, *et al.*. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids* 2013; :811–845doi:10.1002/fld.3767.

TODO LIST

BM: Write abstract 1
BM: Write introduction 1
SKE: Governing equations for compressible Navier-Stokes 2

DOI: 10.1002/nme

BM: Include assumption that $\mathcal{K}_i$ is affine linear? . . . 3
BM: Finish quadrature section . . . 5
BM: Add test for condition number vs approximation quality when varying α . . . 8
BM: Add description and results for cylinder flow . . . 9
SKE: Add description and results for NACA (inviscid) . . . 11
SKE: Add description and results for NACA (viscous) . . . 11
BM: Add conclusion . . . 11
BM: Flux agglomeration? . . . 12
BM: Make indices conformal, i.e. avoid using i (reserved for cells), introduce Einstein convention and use commans between indices . . . 13
BM: Finish cell-agglomeration example? . . . 14

DOI: 10.1002/nme