Extended Discontinuous Galerkin methods for multiphase flows: the spatial discretization

By F. Kummer

1. Motivation and objectives

Over the past decades, numerous numerical methods for solving partial differential equations with a high order of accuracy have been developed. This means that the error of the numerical solution should asymptotically behave as h^p, where h denotes a characteristic length scale of the computational grid and p denotes the convergence order. In fluid dynamics, an order of at least $p = 2$ is widely accepted as a requirement for acceptable accuracy, while the need for higher orders is still part of a dispute, at least for turbulent flows,

However, common to almost all of these methods is that their convergence order is lost for low-regularity solutions, which usually occur in two-phase problems. For material interfaces, the velocity field contains a kink (i.e., a jump of the first derivative) while the pressure field contains a jump; if there is mass-transport across the interface, the velocity field also features a jump. Under such circumstances, the convergence order degenerates usually to h^1, or, even worse $h^{1/2}$ (Gross & Reusken 2007).

This work focuses on an extension of the classical Discontinuous Galerkin (DG) method which should be able to overcome the limitations mentioned above.

2. Introduction/the continuous setting

In order to formalize the two-phase flow setting, we define the following disjoint partitioning of the computational domain $\Omega \subset \mathbb{R}^2$:

$$\Omega = \mathfrak{A} \,\dot{\cup}\, \mathfrak{I} \,\dot{\cup}\, \mathfrak{B}. \tag{2.1}$$

In this work, we restrict ourselves to two-dimensional settings. Here, the sets $\mathfrak{A}$ and $\mathfrak{B}$ denote the individual fluid phases. The density ρ and the viscosity μ are piece-wise constant in $\mathfrak{A}$ and $\mathfrak{B}$:

$$\rho(\boldsymbol{x}) = \begin{cases} \rho_{\mathfrak{A}} & \text{for } \boldsymbol{x} \in \mathfrak{A} \\ \rho_{\mathfrak{B}} & \text{for } \boldsymbol{x} \in \mathfrak{B} \end{cases} \quad \text{and} \quad \mu(\boldsymbol{x}) = \begin{cases} \mu_{\mathfrak{A}} & \text{for } \boldsymbol{x} \in \mathfrak{A} \\ \mu_{\mathfrak{B}} & \text{for } \boldsymbol{x} \in \mathfrak{B} \end{cases}. \tag{2.2}$$

The set $\mathfrak{I}$ denotes the fluid interface. In mathematical terms, we assume it to be a one-dimensional manifold that has at least a continuous and globally bounded curvature. Since in this work we focus only on the spatial discretization, we do not consider a temporal evolution of the phases nor the interface. We define a normal field $\boldsymbol{n}_{\mathfrak{I}}$ on the interface $\mathfrak{I}$, with an orientation that "points from $\mathfrak{A}$ to $\mathfrak{B}$".

In this setting, the two-fluid formulation of the incompressible Navier-Stokes equation is given by the bulk equations in $\Omega \setminus \mathfrak{I}$,

$$\operatorname{div}(\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p = \mu \, \Delta \boldsymbol{u} - \boldsymbol{F}, \tag{2.3}$$

$$\operatorname{div}(\boldsymbol{u}) = 0, \tag{2.4}$$

the jump conditions at the interface $\mathfrak{I}$,

$$\llbracket \boldsymbol{u} \rrbracket = 0, \tag{2.5}$$

$$\left\llbracket p\boldsymbol{n}_{\mathfrak{I}} - \mu(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T)\boldsymbol{n}_{\mathfrak{I}} \right\rrbracket = \sigma\kappa\boldsymbol{n}_{\mathfrak{I}}, \tag{2.6}$$

where σ denotes the surface tension and κ denotes the mean curvature of $\mathfrak{I}$; furthermore, boundary conditions

$$\boldsymbol{u} = \boldsymbol{u}_{\mathrm{D}} \text{ on } \Gamma_{\mathrm{D}} \text{ and } \nabla \boldsymbol{u} \cdot \boldsymbol{n}_{\partial\Omega} - p = 0 \text{ on } \Gamma_{\mathrm{N}}, \tag{2.7}$$

where $\partial\Omega = \Gamma_{\mathrm{D}} \,\dot{\cup}\, \Gamma_{\mathrm{N}}$ denotes a disjoint decomposition of the boundary into a Dirichlet and Neumann boundary region.

Note that, from a physical point of view, the presented setting is somewhat contradictory: if the interface does not move in time, the velocity of the fluid in normal direction to the interface needs to be zero ($\boldsymbol{u} \cdot \boldsymbol{n}_{\mathfrak{I}} = 0$). From a mathematical point of view, however, we only enforce $\llbracket \boldsymbol{u} \rrbracket = 0$, but not that the interface has to move with the same speed as the flow. This may be interpreted as a singular mass source of the form $\delta_{\mathfrak{I}} \llbracket \rho \boldsymbol{u} \cdot \boldsymbol{n}_{\mathfrak{I}} \rrbracket$, where $\delta_{\mathfrak{I}}$ denotes a delta-distribution at the interface.

3. The extended Discontinuous Galerkin discretization of the two-phase problem

In this section, a formal definition of the proposed discretization of the two-phase problem is given. First, we recall some basic definitions (Section 3.1); second, we introduce the extended DG space (Section 3.2); and finally we apply this framework onto the two-phase problem (Section 3.3).

3.1. *A few definitions*

While keeping the amount of Bourbakism in this work as low as possible, for the sake of completeness, we have to recall some basic definitions and notations that form the DG framework. These are fairly standard, and can be found in similar form in many textbooks (see Hesthaven & Warburton 2008, Di Pietro & Ern 2011). We define:

- the computational domain: $\Omega \subset \mathbb{R}^2$ which must be polygonal and simply connected;
- the numerical grid: $\mathfrak{K}_h = \{K_1, \ldots, K_J\}$, with h being the maximum diameter of all cells K_j. The cells cover the whole domain ($\overline{\Omega} = \bigcup_j \overline{K_j}$), but do not overlap ($\int_{K_j \cap K_l} 1 \, \mathrm{d}\boldsymbol{x} = 0$ for $l \neq j$). We restrict ourself to non-curved grids, i.e., each cell K_j can be described as the image of a reference cell K_{ref} under an affine-linear mapping $T_j : \mathbb{R}^2 \to \mathbb{R}^2$, i.e. $T_j(K_{\mathrm{ref}}) = K_j$;
- the set containing edges of the grid: $\Gamma := \bigcup_j \partial K_j$. Furthermore, the set of all internal edges: $\Gamma_{\mathrm{int}} := \Gamma \setminus \partial\Omega$;
- a normal field $\boldsymbol{n}_\Gamma$ on Γ. On $\partial\Omega$, it represents an outer normal, i.e., on $\partial\Omega$, $\boldsymbol{n}_\Gamma = \boldsymbol{n}_{\partial\Omega}$. By $\boldsymbol{n}_{\mathfrak{I},\Gamma}$, we denote a normal field that is equal to $\boldsymbol{n}_\Gamma$ on Γ and equal to $\boldsymbol{n}_{\mathfrak{I}}$ on $\mathfrak{I}$;
- the jump and the average value operator: for $\boldsymbol{x} \in \Gamma_{\mathrm{int}}$,

$$\llbracket u \rrbracket (\boldsymbol{x}) := \lim_{\xi \searrow 0} \left(u(\boldsymbol{x} + \xi \boldsymbol{n}_\Gamma) - u(\boldsymbol{x} - \xi \boldsymbol{n}_\Gamma) \right), \tag{3.1}$$

$$\{\{u\}\} (\boldsymbol{x}) := \lim_{\xi \searrow 0} \tfrac{1}{2} \left(u(\boldsymbol{x} + \xi \boldsymbol{n}_\Gamma) + u(\boldsymbol{x} - \xi \boldsymbol{n}_\Gamma) \right). \tag{3.2}$$

For the fluid interface $\mathfrak{I}$ and the corresponding normal $\boldsymbol{n}_{\mathfrak{I}}$, the jump operator is defined in the same way. Note that the actual sign of the jump operator depends on the direction of the normal. Since there is no natural choice for the direction, i.e., the sign of $\boldsymbol{n}_\Gamma$ on Γ_{int}, definition of the jump operator seems quite arbitrary. However, on every occasion at which the jump operator is used, it appears as part of a product, of type $[\![-]\!]\,\boldsymbol{n}_\Gamma$ or $[\![-]\!]\,[\![-]\!]$, so that the result does not depend on the actual choice of the sign of $\boldsymbol{n}_\Gamma$. On the boundary $\partial\Omega$, jump and average are re-defined as

$$[\![u]\!]\,(\boldsymbol{x}) := \lim_{\xi\searrow 0} u(\boldsymbol{x} - \xi\boldsymbol{n}_\Gamma) \text{ and } \{\{u\}\}\,(\boldsymbol{x}) := \lim_{\xi\searrow 0} u(\boldsymbol{x} - \xi\boldsymbol{n}_\Gamma);$$

- the DG-approximation space of degree p:

$$\mathbb{P}_p(\mathfrak{K}_h) := \left\{f \in L^2(\Omega);\ \forall\ K \in \mathfrak{K}_h : f|_K \text{ is polynomial and } \deg\left(f|_K\right) \le p\right\}; \tag{3.3}$$

- the broken gradient ∇_h: for $u \in \mathcal{C}^1(\Omega \setminus \Gamma \setminus \mathfrak{I})$, $\nabla_h u$ denotes the gradient on the domain $\Omega \setminus \Gamma \setminus \mathfrak{I}$;
- the species-volume-fraction $\mathrm{fr}_\mathfrak{s}\,(K)$ for some cell $K \in \mathfrak{K}_h$: $\mathrm{fr}_\mathfrak{s}\,(K) := \int_{K\cap\mathfrak{s}} 1\ \mathrm{d}\boldsymbol{x} / \int_K 1\ \mathrm{d}\boldsymbol{x}$, for $\mathfrak{s} \in \{\mathfrak{A}, \mathfrak{B}\}$;
- the intersection of $\mathfrak{K}_h$ with $\mathfrak{s} \in \{\mathfrak{A}, \mathfrak{B}\}$: $K \wedge \mathfrak{s} := \{K \cap \mathfrak{s} \mid K \in \mathfrak{K}_h,\ \mathrm{fr}_\mathfrak{s}\,(K) > 0\}$;
- the edge-neighbors of some cell K: these are those cells $L \in \mathfrak{K}_h$ that share at least an edge (not only a vertex), i.e. $\oint_{K\cap L} 1\ \mathrm{dS} > 0$;
- the standard-basis vector $\boldsymbol{e}_d$, for $d \in \{1, 2\}$: $\boldsymbol{e}_1 = (1, 0)$, $\boldsymbol{e}_2 = (0, 1)$;

3.2. *The extended DG discretization*

In order to represent the phases $\mathfrak{A}$ and $\mathfrak{B}$ as well as the interface $\mathfrak{I}$ numerically, we employ a level-set formulation. We chose some sufficiently smooth scalar field $\varphi \in \mathbb{P}_r(\mathfrak{K}_h) \cap \mathcal{C}^n(\Omega)$, so that

$$\begin{aligned} \mathfrak{I} &= \{\boldsymbol{x} \in \Omega;\ \varphi(\boldsymbol{x}) = 0\}, \\ \mathfrak{A} &= \{\boldsymbol{x} \in \Omega;\ \varphi(\boldsymbol{x}) < 0\} \text{ and} \\ \mathfrak{B} &= \{\boldsymbol{x} \in \Omega;\ \varphi(\boldsymbol{x}) > 0\}. \end{aligned}$$

For a comprehensive overview on level-set methods we refer, for example, to the textbooks of Sethian (1996) or Osher & Fedkiw (2002). Since we consider only steady-state situations without any interface movement, level-set evolution algorithms are not relevant for this work and are not be discussed. Obviously, the interface normal can be obtained from the gradient of φ and the curvature can be calculated by Bonnet's formula:

$$\boldsymbol{n}_\mathfrak{I} = \frac{\nabla\varphi}{|\nabla\varphi|} \quad \text{and} \quad \kappa = \mathrm{div}\left(\frac{\nabla\varphi}{|\nabla\varphi|}\right).$$

The extended DG space, or XDG-space, is defined as

$$\begin{aligned} \mathbb{P}^X_p(\varphi, \mathfrak{K}_h) := \big\{f \in L^2(\Omega);\ \forall\ K \in \mathfrak{K}_h : f|_{K\cap\mathfrak{A}}, f|_{K\cap\mathfrak{B}} &\text{ are polynomial,} \\ \deg\left(f|_{K\cap\mathfrak{A}}\right) \le p \text{ and } \deg\left(f|_{K\cap\mathfrak{B}}\right) \le p\big\}. \end{aligned} \tag{3.4}$$

We note that

$$\mathbb{P}^X_p(\varphi, \mathfrak{K}_h) = \mathbb{P}_p((\mathfrak{K}_h \wedge \mathfrak{A}) \cup (\mathfrak{K}_h \wedge \mathfrak{B})). \tag{3.5}$$

We further define the continuation mapping in the cut cells,

$$\mathit{Cont}_\mathfrak{s} : \mathbb{P}^X_p(\varphi, \mathfrak{K}_h) \to \mathbb{P}_p(\mathfrak{K}_h),$$

for $\mathfrak{s} \in \{\mathfrak{A}, \mathfrak{B}\}$, which continues $u \in \mathbb{P}^X_p(\varphi, \mathfrak{K}_h)$ from 'one side' of $\mathfrak{I}$ into the domain of cut cells. Because of the polynomial structure of domain and codomain, $\mathit{Cont}_\mathfrak{s}$ is uniquely determined by the property

$$\left\{ \begin{array}{rll} \mathit{Cont}_\mathfrak{s}(u)\mathbf{1}_\mathfrak{s} & = u\ \mathbf{1}_\mathfrak{s} & \\ \mathit{Cont}_\mathfrak{s}(u) & = 0 & \text{in } \bigcup_{K\in\mathfrak{K}_h\setminus(\mathfrak{K}_h\wedge\mathfrak{s})} K \end{array} \right. ,$$

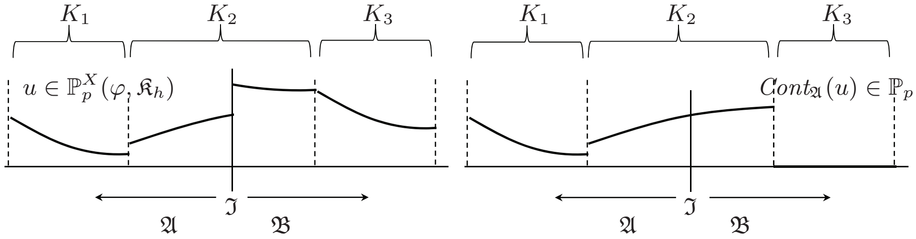

FIGURE 1. Continuation mapping. Left: $u \in \mathbb{P}_p^X(\varphi, \mathfrak{K}_h)$, right: $\mathit{Cont}_{\mathfrak{A}}(u) \in \mathbb{P}_p(\mathfrak{K}_h)$. The example shows a 1D grid consisting of three cells: $\mathfrak{K}_h = \{K_1, K_2, K_3\}$. Within K_1, since it is fully contained in $\mathfrak{A}$, (i.e., $\mathfrak{A} \cap K_1 = K_1$), $\mathit{Cont}_{\mathfrak{A}}(u)|_{K_1} = u|_{K_1}$. For K_3, which does not intersect with $\mathfrak{A}$, $\mathit{Cont}_{\mathfrak{A}}(u)|_{K_3} = 0$. Finally for K_2, which intersects with both $\mathfrak{A}$ and $\mathfrak{B}$, there is only one polynomial continuation in $\mathbb{P}_p(\mathfrak{K}_h)$ so that $\mathit{Cont}_{\mathfrak{A}}(u)|_{K_2 \cap \mathfrak{A}} = u|_{K_2 \cap \mathfrak{A}}$.

where $\mathbf{1}_X$ denotes the characteristic function for some set X. An illustration is given in Figure 1.

3.3. *The spatial discretization of the two-phase problem*

In order to comply with the Ladyženskaja-Babuška-Brezzi (LBB) condition (see e.g., Babuška 1973; Brezzi 1974) in the case of equal physical parameters (i.e., $\rho_{\mathfrak{A}} = \rho_{\mathfrak{B}}$, $\mu_{\mathfrak{A}} = \mu_{\mathfrak{B}}$) we discretize velocity and pressure in XDG spaces of order k and $k' := k - 1$, respectively, i.e.,

$$(\boldsymbol{u}, p) \in \mathbb{P}_k(\varphi, \mathfrak{K}_h)^2 \times \mathbb{P}_{k-1}(\varphi, \mathfrak{K}_h) =: V_k. \tag{3.6}$$

We propose the following discretization of the problem (2.3, 2.4) with jump conditions (2.5, 2.6) and boundary conditions (2.7) in the extended DG space: find $(\boldsymbol{u}, p) \in V_k$, so that for all $(\boldsymbol{v}, \tau) \in V_k$

$$t(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{v}) + b(p, \boldsymbol{v}) = a(\boldsymbol{u}, \boldsymbol{v}) + s(\boldsymbol{v}), \tag{3.7}$$

$$-b(\tau, \boldsymbol{u}) = r(\tau), \tag{3.8}$$

or, equivalently,

$$L((\boldsymbol{u}, p), (\boldsymbol{v}, \tau)) + t(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{v}) = s(\boldsymbol{v}) + r(\tau), \tag{3.9}$$

with the symmetric linear form

$$L((\boldsymbol{u}, p), (\boldsymbol{v}, \tau)) = b(p, \boldsymbol{v}) - a(\boldsymbol{u}, \boldsymbol{v}) - b(\tau, \boldsymbol{u}). \tag{3.10}$$

Here, the (tri/bi)linear forms t, b, a, s and r, respectively, represent convectional terms ($t(-,-,-)$), pressure gradient and velocity divergence ($b(-,-)$), viscous terms ($a(-,-)$), surface tension, boundary conditions and body forces ($s(-)$), and boundary conditions for the continuity equation ($r(-)$).

The convectional terms are discretized by a local Lax-Friedrichs flux,

$$t(\boldsymbol{w}, \boldsymbol{u}, \boldsymbol{v}) = -\int_{\Omega} \rho(\boldsymbol{u} \otimes \boldsymbol{w}) : \nabla_h \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} - \oint_{\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{N}} \cup \mathfrak{I}} \left(\{\{\boldsymbol{u} \otimes \boldsymbol{w}\}\} \, \boldsymbol{n}_{\mathfrak{I},\Gamma} + (\lambda/2) [\![\boldsymbol{u}]\!] \right) \cdot [\![\rho \boldsymbol{v}]\!] \ \mathrm{dS}. \tag{3.11}$$

Details on the stabilization parameter λ are given in Section 3.3.1.

Pressure gradient and velocity divergence are discretized as

$$b(p, \boldsymbol{v}) = -\int_{\Omega} p \ \mathrm{div}\,(\boldsymbol{v}) \ \mathrm{d}\boldsymbol{x} - \oint_{(\Gamma \setminus \Gamma_{\mathrm{N}}) \cup \mathfrak{I}} [\![\boldsymbol{v}]\!] \cdot \boldsymbol{n}_{\mathfrak{I},\Gamma} \ \{\{p\}\} \ \mathrm{dS}. \tag{3.12}$$

To discretize the viscous terms, we employ an almost-standard symmetric interior penalty method (SIP), first introduced by Arnold (1982) (for an extensive analysis, see e.g., Arnold *et al.* 2002):

$$\begin{aligned} a(\boldsymbol{u},\boldsymbol{v}) = &-\int_{\Omega} \mu \nabla_h \boldsymbol{u} : \nabla_h \boldsymbol{v} \ \mathrm{d}\boldsymbol{x} + \oint_{\Gamma\setminus\Gamma_{\mathrm{N}}\setminus\mathfrak{I}} \mu \left(\{\{\nabla_h \boldsymbol{u}\}\} \llbracket \boldsymbol{v} \rrbracket + \{\{\nabla_h \boldsymbol{v}\}\} \llbracket \boldsymbol{u} \rrbracket \right) \cdot \boldsymbol{n}_{\mathfrak{I}} \ \mathrm{dS} \\ &- \sum_{\mathfrak{s}\in\{\mathfrak{A},\mathfrak{B}\}} \oint_{\Gamma\setminus\Gamma_{\mathrm{N}}\setminus\mathfrak{I}} \mu \ \eta \llbracket Cont_{\mathfrak{s}}(\boldsymbol{u}) \rrbracket \cdot \llbracket Cont_{\mathfrak{s}}(\boldsymbol{v}) \rrbracket \ \mathrm{dS} \\ &+ \oint_{\mathfrak{I}} \left(\left(\{\{\mu \nabla_h \boldsymbol{u}\}\} + \left\{\left\{\mu (\nabla_h \boldsymbol{u})^T\right\}\right\} \right) \boldsymbol{n}_{\mathfrak{I}} \right) \cdot \llbracket \boldsymbol{v} \rrbracket + \left(\{\{\mu \nabla_h \boldsymbol{v}\}\} \right) \cdot \llbracket \boldsymbol{u} \rrbracket \ \mathrm{dS} \\ &- \oint_{\mathfrak{I}} \eta \llbracket \boldsymbol{u} \rrbracket \cdot \llbracket \boldsymbol{v} \rrbracket \max\{\mu_{\mathfrak{A}}, \mu_{\mathfrak{B}}\} \ \mathrm{dS}. \end{aligned} \tag{3.13}$$

This form contains two differences in comparison to the classical SIP-method. First, we chose a modified consistency and symmetry term at the interface (fourth term of Eq. 3.13) in order to achieve consistency with the strain jump condition (Eq. 2.6). Second, the penalty term (third term of Eq. 3.13) is modified; further details are given in Section 3.3.2.

Finally, we specify the sources

$$\begin{aligned} s(\boldsymbol{v}) = &-\int_{\Omega} \boldsymbol{F} \cdot \boldsymbol{v} \ \mathrm{d}\boldsymbol{x} + \oint_{\mathfrak{I}} \kappa \ \sigma \ (\boldsymbol{n}_{\mathfrak{I}} \cdot \llbracket \boldsymbol{v} \rrbracket) \ \mathrm{dS} \\ &- \oint_{\Gamma_{\mathrm{D}}} (\boldsymbol{u}_{\mathrm{D}} \otimes \boldsymbol{u}_{\mathrm{D}}) \boldsymbol{n}_{\mathfrak{I}} \cdot \llbracket \rho \boldsymbol{v} \rrbracket \ \mathrm{dS} - \oint_{\Gamma_{\mathrm{D}}} \boldsymbol{u}_{\mathrm{D}} \cdot (\nabla_h \boldsymbol{v} \boldsymbol{n}_{\partial\Omega} - \eta \boldsymbol{v}) \ \mathrm{dS}, \end{aligned} \tag{3.14}$$

and

$$r(\tau) = \oint_{\Gamma_{\mathrm{D}}} \tau \ \boldsymbol{u}_{\mathrm{D}} \cdot \boldsymbol{n}_{\partial\Omega} \ \mathrm{dS}. \tag{3.15}$$

In this expression, the first, second, third and fourth terms represent, respectively, volume force, force induced by surface tension, Dirichlet boundary conditions for the convective part, and Dirichlet boundary conditions for the viscous part.

3.3.1. *Choice of the Local-Lax-Friedrichs parameter λ*

The stabilization parameter for the convectional terms is chosen as

$$\lambda := \max\left\{ |\varrho|; \varrho \in \operatorname{spec}\left(\boldsymbol{Q}\left(\overline{\boldsymbol{u}^{\mathrm{in}}} \right) \right) \cup \operatorname{spec}\left(\boldsymbol{Q}\left(\overline{\boldsymbol{u}^{\mathrm{out}}} \right) \right) \right\},$$

where spec($\boldsymbol{Q}$) denotes the spectrum of the matrix $\boldsymbol{Q}$; the matrix $\boldsymbol{Q}(\boldsymbol{u})$ is given as

$$\boldsymbol{Q}(\boldsymbol{u}) = \nabla_{\boldsymbol{u}} \left((\boldsymbol{u} \otimes \boldsymbol{u}) \boldsymbol{n}_{\mathfrak{I},\Gamma} \right) = \begin{bmatrix} 2u_1 n_1 + u_2 n_2 & u_1 n_2 \\ u_2 n_1 & u_1 n_1 + 2 u_2 n_2 \end{bmatrix},$$

and $\overline{\boldsymbol{u}^{\mathrm{in}}}$, $\overline{\boldsymbol{u}^{\mathrm{out}}}$ denote the mean values of $\boldsymbol{u}^{\mathrm{in}}$, $\boldsymbol{u}^{\mathrm{out}}$ within two cells adjacent to the respective edge $E \subset \Gamma$.

3.3.2. *Choice of the symmetric interior penalty parameter η*

Note that the discussion within this section is preliminary and more extensive analysis is required.

Regarding the classical SIP-method, it is well known (see e.g. Shahbazi 2005) that the penalty parameter η must be chosen large enough to ensure coercivity of the form $a(-,-)$; however, a large penalty parameter increases the condition number of the problem. Thus, it is important to chose η as small as possible, yet large enough. It is also well known that for some cell face $E \subset \Gamma$, η (respectively, the lower bound of η which ensures coercivity) scales as

$$\eta \sim \frac{k^2}{h_E},$$

where k denotes the DG-polynomial degree and h_E the diameter of E. Therefore, η is usually chosen differently for each edge, depending on the local geometric properties of the grid, i.e., $\eta = \eta(\boldsymbol{x})$, for $\boldsymbol{x} \in \Gamma \cup \mathfrak{I}$.

At all cells K which are not cut by the interface $\mathfrak{I}$, we use the explicit lower-bound estimate proposed by Shahbazi (2005). In cut cells, however, h_E may become very small, causing an undesirably high condition number. Our main tool which we use to address this issue is cell-agglomeration (discussed in Section 5). Numerical experiments (see Sections 5 and 6) may indicate that agglomeration eliminates the need for scaling η with the local size of the cut edges. Indeed, none of the experimental results given below contains any such scaling; in cut cells, we only scale by the factor $1/h_K$, where h_K denotes the minimum diameter of the uncut cell. As noted above, we are not yet able to present any theoretical analysis on that.

In addition to cell-agglomeration, we employ another "trick", which to our knowledge was first proposed and analyzed by Massing *et al.* (2012). This actually regards the third term of Eq. (3.13). Instead of penalizing $[\![u]\!]$, for some $u \in \mathbb{P}^X_k(\varphi, \mathfrak{K}_h)$, the expression $[\![Cont_{\mathfrak{s}}(u)]\!]$ is penalized. For the following discussion, w.l.o.g., we assume that $u|_{\mathfrak{B}} = 0$. Then u can be represented as $u = \mathbf{1}_{\mathfrak{A}}\ u_{\mathfrak{A}}$, with $u_{\mathfrak{A}} \in \mathbb{P}_k(\mathfrak{K}_h)$, and $u_{\mathfrak{A}} \equiv 0$ for all cells K fully occupied by species $\mathfrak{B}$ (i.e., $\mathfrak{B} \cap K = K$); therefore, $Cont_{\mathfrak{A}}(u) = u_{\mathfrak{A}}$. For some cell-face $E \subset \Gamma$, this implies the following:

- if E is fully contained in either $\mathfrak{A}$ or $\mathfrak{B}$, $[\![Cont_{\mathfrak{A}}(u)]\!] = [\![u]\!]$ and the penalty term reduces to the classical form that is used in the SIP-method;
- on cut cell faces (i.e. $\mathfrak{I} \cap E \neq \{\}$), $[\![Cont_{\mathfrak{A}}(u)]\!] = [\![u_{\mathfrak{A}}]\!]$. In other words, instead of penalizing $[\![u]\!]$ on the domain $\mathfrak{A} \cap E$, the continuation $u_{\mathfrak{A}}$ is penalized on the whole face E;

4. Numerical integration

One actual challenge for implementing of XDG-methods is the integration of functions that are discontinuous at the interface $\mathfrak{I}$, or equivalently the numerical evaluation of the integrals

$$\oint_{\partial K \cap \mathfrak{A}} f \ \mathrm{dS}, \ \oint_{\partial K \cap \mathfrak{B}} f \ \mathrm{dS}, \ \oint_{K \cap \mathfrak{I}} f \ \mathrm{dS}, \ \int_{K \cap \mathfrak{A}} f \ \mathrm{d}\boldsymbol{x}, \ \int_{K \cap \mathfrak{B}} f \ \mathrm{d}\boldsymbol{x}$$

for some sufficiently smooth $f \in \mathcal{C}^n(K)$, on a reference cell $K \subset \mathbb{R}^2$, in order to evaluate the forms defined in the previous section.

We address this problem by a method proposed by Müller *et al.* (2013), with slight modifications that will be outlined below. For numerical integration/quadrature, using

$$\text{nodes } \underline{\boldsymbol{x}} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_L) \ \text{ and weights } \underline{w} = (w_1, \dots, w_L)$$

of a function $f(\boldsymbol{x})$, we define the notation

$$\int_{(\underline{\boldsymbol{x}}, \underline{w})}^{num} f := \sum_{l=1}^{L} w_l f(\boldsymbol{x}_l). \tag{4.1}$$

The starting point of our examination is the Gauss theorem, applied onto an orthonormal basis $\underline{\phi} = (\phi_1, \ldots \phi_N)$ of $\mathbb{P}_q(\{K\})$,

$$\int_{K\cap\mathfrak{A}} \operatorname{div}(\phi_n \boldsymbol{e}_d) \, \mathrm{d}\boldsymbol{x} - \oint_{K\cap\mathfrak{I}} \phi_n \boldsymbol{e}_d \cdot \boldsymbol{n}_{\mathfrak{I}} \, \mathrm{dS} = \oint_{\partial K\cap\mathfrak{A}} \phi_n \boldsymbol{e}_d \cdot \boldsymbol{n}_{\partial K} \, \mathrm{dS}, \tag{4.2}$$

respectively, its discrete counterpart

$$\int_{(\underline{\boldsymbol{x}},\underline{w}^{\mathfrak{A}})}^{num} \operatorname{div}(\phi_n \boldsymbol{e}_d) - \int_{(\underline{\boldsymbol{x}},\underline{w}^{\mathfrak{I}})}^{num} \phi_n \boldsymbol{e}_d \cdot \boldsymbol{n}_{\mathfrak{I}} = \int_{(\underline{\boldsymbol{x}}^{\partial},\underline{w}^{\partial})}^{num} \phi_n \boldsymbol{e}_d \cdot \boldsymbol{n}_{\partial K}. \tag{4.3}$$

Here, three different quadrature rules are involved: for integration on the 'bulk' $K\cap\mathfrak{A}$ the rule $(\underline{\boldsymbol{x}}, \underline{w}^{\mathfrak{A}})$, for integration on the interface $K \cap \mathfrak{I}$ the rule $(\underline{\boldsymbol{x}}, \underline{w}^{\mathfrak{I}})$, and for integration on the cell boundary $\partial K \cap \mathfrak{A}$ the rule $(\underline{\boldsymbol{x}}^{\partial}, \underline{w}^{\partial})$. As suggested by Müller *et al.* (2013), we use the same quadrature nodes $\underline{\boldsymbol{x}}$ for bulk and interface integration.

In the original approach the discrete Gauss theorem (Eq. 4.3) is tested only with vector-valued functions

$$\boldsymbol{f}_n := \frac{1}{2}\left(\int \phi_n \, \mathrm{d}x, \int \phi_n \, \mathrm{d}y\right),$$

instead of the functions $\phi_n \boldsymbol{e}_d$ which we propose here. This yields a lower number of equations, but Eq. (4.3) is only *approximately* fullfilled. In order to obtain a consistent discretization based on (3.7, 3.8), Eq. (4.3) needs to be *exactly* fulfilled for polynomial degree $q = 2k + 1$ and all $d \in \{1, 2\}$, up to numerical round-off errors.

The rules $(\underline{\boldsymbol{x}}, \underline{w}^{\mathfrak{A}})$ and $(\underline{\boldsymbol{x}}, \underline{w}^{\mathfrak{I}})$, which fulfill Eq. (4.3), can be constructed as follows: In order to determine the right-hand side of Eq. (4.3), we follow Müller's original approach and construct the quadrature rule $(\underline{\boldsymbol{x}}^{\partial}, \underline{w}^{\partial})$ by finding the roots of the level-set field φ on the one-dimensional set ∂K. The quadrature nodes $\underline{\boldsymbol{x}}$ may be chosen arbitrarily in K. Then, Eq. (4.3) represents a linear system of equations in the unknowns $\underline{w}^{\mathfrak{I}}$ and $\underline{w}^{\mathfrak{A}}$. Since the number of equations is, in two dimensions, $2N$ (in three dimensions, $3N$) and the number of unknowns is $2L$, a necessary condition for the existence of an exact solution to Eq. (4.3) is $L \geq N$ (in three dimensions, $2L \geq 3N$). Then Eq. (4.3) represents a underdetermined system, which may be solved for instance with the LAPACK function `dgesly`.

Finally, to obtain a quadrature rule for the domain $K \cap \mathfrak{B}$, we use the identity

$$\int_{K\cap\mathfrak{B}} f \, \mathrm{d}\boldsymbol{x} = \int_K f \, \mathrm{d}\boldsymbol{x} - \int_{K\cap\mathfrak{A}} f \, \mathrm{d}\boldsymbol{x}.$$

Then, if $(\underline{\boldsymbol{x}}^K, \underline{w}^K)$ denotes a quadrature rule – of at least an order q – for domain K, a rule for $K \cap \mathfrak{B}$ is given by

$$\left([\underline{\boldsymbol{x}}^K, \underline{\boldsymbol{x}}], [\underline{w}^K, -\underline{w}^{\mathfrak{A}}]\right),$$

where $[\underline{a}, \underline{b}]$ denotes the concatenation of the vectors $\underline{a}$ and $\underline{b}$. By this choice, we ensure that the sum over the numerical integrals over both domains $K \cap \mathfrak{A}$ and $K \cap \mathfrak{B}$ is indeed equal to the integral over K. The boundary integral $\partial K \cap \mathfrak{B}$ can be handled in analog fashion.

5. Cell-agglomeration, choice of basis and condition number

In general, the linear system (3.7,3.8) may still have a very high condition number if there are some cells K with either a small $\mathfrak{A}$- or $\mathfrak{B}$ - fraction (i.e., $\mathrm{fr}_{\mathfrak{A}}(K) \ll 1$ or

$\mathrm{fr}_{\mathfrak{B}}(K) \ll 1$). Numerical experiments indicate that the agglomeration of small cut cells to larger neighbors may be some way to effectively overcome this issue.

First, we determine a list A_α of cell-pairs to agglomerate, for an agglomeration threshold $0 \leq \alpha < 1$, by the following algorithm:

```
set $A_\alpha := \{\}$
**for all** species $\mathfrak{s} \in \{\mathfrak{A}, \mathfrak{B}\}$ **do**
    **for all** cells $K \in \mathfrak{K}_h$, with $\mathrm{fr}_{\mathfrak{s}}(K) > 0$ **do**
        **if** $\mathrm{fr}_{\mathfrak{s}}(K) < \alpha$ **then**
            select cell $L_{\max}$ form all edge-neighbors $L$ of $K$, where $\mathrm{fr}_{\mathfrak{s}}(L)$ is maximal.
            $A_\alpha \leftarrow A_\alpha \cup \{\{K, L_{\max}\}\}$, i.e. add the pair $\{K, L_{\max}\}$ to $A_\alpha$
        **end if**
    **end for**
**end for**
```

Then, we define the agglomerated XDG space

$$\mathbb{P}_p^X(\varphi, \mathfrak{K}_h, A_\alpha) := \left\{ f \in \mathbb{P}_p^X(\varphi, \mathfrak{K}_h) \mid \forall\, K, L \in A_\alpha : f|_{K \cup L} \text{ is polynomial} \right\}. \tag{5.1}$$

By analogy to Eq. (3.6), we define the agglomerated $(k, k-1)$ – velocity/pressure space

$$\mathbb{P}_k(\varphi, \mathfrak{K}_h, A_\alpha)^2 \times \mathbb{P}_{k-1}(\varphi, \mathfrak{K}_h, A_\alpha) =: V_k^\alpha, \tag{5.2}$$

which we finally use to discretize (3.7,3.8).

Below, we perform an experimental study of the condition number of the Stokes system

$$L((\boldsymbol{u}, p), (\boldsymbol{v}, \tau)) = s(\boldsymbol{v}) + r(\tau), \tag{5.3}$$

which can be obtained by neglecting the convective part $t(-,-,-)$ in Eq. (3.9), in the agglomerated space V_k^α. Therefore, we chose basis functions $\boldsymbol{\phi}_{j,\mathfrak{s},d,n}$ of V_k^α, with the following properties:

- $\mathrm{supp}(\boldsymbol{\phi}_{j,\mathfrak{s},d,n}) = \overline{K_j \cap \mathfrak{s}}$, i.e., j denotes a cell-index and $\mathfrak{s}$ the species. For some fixed j, $\mathfrak{s}$ may be any species in $\{\mathfrak{A}, \mathfrak{B}\}$ which is actually present in cell K_j, i.e., $\mathfrak{s} \in \{\mathfrak{r} \in \{\mathfrak{A}, \mathfrak{B}\} \mid \mathrm{fr}_{\mathfrak{r}}(K_j) > 0\}$;
- d denotes the velocity components ($d \in \{1, 2\}$) resp. the pressure ($d = 3$), i.e., $\boldsymbol{\phi}_{j,\mathfrak{s},d,n} \cdot \boldsymbol{e}_l = 0$ for $d \neq l$;
- n denotes a polynomial index and depends on d, since for the pressure space ($d = 3$) the polynomial is one degree lower than that for the velocity;

Given such a basis, one may express the matrix of $L(-,-)$ as

$$\mathcal{M}_{(j,\mathfrak{s},n,d),(i,\mathfrak{r},m,l)} = L(\boldsymbol{\phi}_{j,\mathfrak{s},n,d}, \boldsymbol{\phi}_{i,\mathfrak{r},m,l}). \tag{5.4}$$

Here, the tuple $(j, \mathfrak{s}, n, d)$ actually denotes a bijection between all valid combinations of $j, \mathfrak{s}, n, d$, as specified above and the set $\{1, 2, \ldots, \dim(V_k^\alpha)\}$.

In order to minimize any influence on the matrix condition number that is actually caused by the specific choice of the basis, a symmetric block-Jacobi-preconditioning is applied to $\mathcal{M}$: i.e.,

$$\mathcal{M}^{pc} := P^T \mathcal{M} P, \tag{5.5}$$

such that the $((j, \mathfrak{s}, -, d), (j, \mathfrak{s}, -, d))$-diagonal blocks of $\mathcal{M}^{pc}$ are identity matrices, for $d = 1, 2$, respectively.

Note that the symmetric block-Jacobi-preconditioning is indeed equivalent to a change of the DG basis: assume an N-dimensional vector space W over $\mathbb{R}$, and two different bases

$$\underline{\phi} = (\phi_1, \dots \phi_N) \text{ and } \underline{\psi} = (\psi_1, \dots \psi_N),$$

linked by an $N \times N$ – change-of-basis matrix A, i.e.,

$$\underline{\psi} = \underline{\phi} A, \quad \text{resp.,}\ \psi_n = \sum_m A_{mn} \phi_m.$$

Let $\mathcal{R}$ and $\mathcal{Q}$ be the matrices associated with some real-valued bilinear form $f(-,-)$ on $W \times W$, with respect to $\underline{\phi}$ and $\underline{\psi}$, i.e.,

$$\mathcal{R}_{ij} = f(\phi_i, \phi_j) \text{ and } \mathcal{Q}_{ij} = f(\psi_i, \psi_j).$$

A simple calculation shows that

$$\mathcal{Q}_{ij} = f\left(\sum_m A_{mi} \phi_m, \sum_n A_{nj} \phi_n \right) = (A^T \mathcal{R} A)_{ij}.$$

Therefore, the preconditioning applied in Eq. 5.5 is indeed equivalent to a change of the DG-basis, and every non-singular block-diagonal matrix is a possible choice for the preconditioning P. In our experiments, the particular choice mentioned above (chose P such that the diagonal blocks of $P^T \mathcal{M} P$ are identity matrices for each cell, each variable, and each species) usually gave the lowest condition number. Other variants – including re-scaling the basis in every cell and orthonormalizing the cut-cell basis – yield overall larger condition numbers.

The actual numerical study is performed on an equidistant 18×18 – cells' discretization of the domain $\Omega = (-1.5, 1.5)^2$, the DG-Polynomial order $k = 3$. The level set field is chosen as $\varphi = x^2 + y^2 - R^2$, which can be exactly represented in $\mathbb{P}_2(\mathfrak{K}_h)$. The physical parameters were chosen to be those of air and water ($\sigma = 0.072$, $\rho_{\mathfrak{A}} = 10^3$, $\mu_{\mathfrak{A}} = 10^{-3}$ $\rho_{\mathfrak{B}} = 1.2$, $\mu_{\mathfrak{B}} = 17.1 \cdot 10^{-6}$; because of the 2D-situation, we do not specify units here). We test various radii R against different agglomeration thresholds α, namely,

$$\begin{aligned} R \in \{&0.69711, 0.70611, 0.70711, 0.70811, 0.71711, 0.75468, 0.80226, \\ &0.83984, 0.84884, 0.84984, 0.85084, 0.85984\} \text{ and} \\ \alpha \in \{&0, 0.01, 0.02, 0.05, 0.1\}. \end{aligned} \tag{5.6}$$

The different radii are chosen so that multiple rather small cuts occur (see Figure 2). The numerical experiment indicates that the choice $\alpha \approx 0.1$ yields a rather constant condition number among all tested radii (see Table 1 and Figure 3). Condition numbers of $\mathcal{M}^{pc}$ were computed by the MATLAB-function `condest`.

6. Numerical results

In this section, we study the solution of ellipsoidal, static droplets in the domain $\Omega = (-1.5, 1.5)^2$. The level - set function is given by

$$\varphi(x, y) = \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1, \tag{6.1}$$

which can obviously be exactly represented in $\mathbb{P}_2(\mathfrak{K}_h)$. As noted in the introduction, this may be contradictory from a physical point of view. It is, however, an interesting model problem to study important numerical properties of the scheme, such as the convergence order.

α	min. cond. no.	max. cond. no.
0	$6.45 \cdot 10^6$	$3.74 \cdot 10^{29}$
0.05	$6.25 \cdot 10^6$	$2.11 \cdot 10^{10}$
0.01	$6.45 \cdot 10^6$	$3.3 \cdot 10^{12}$
0.02	$6.45 \cdot 10^6$	$2.11 \cdot 10^{10}$
0.05	$6.25 \cdot 10^6$	$2.11 \cdot 10^{10}$
0.1	$6.25 \cdot 10^6$	$1.29 \cdot 10^7$

TABLE 1. Maximum condition number of $\mathcal{M}^{pc}$ over all tested radii for various agglomeration thresholds α.

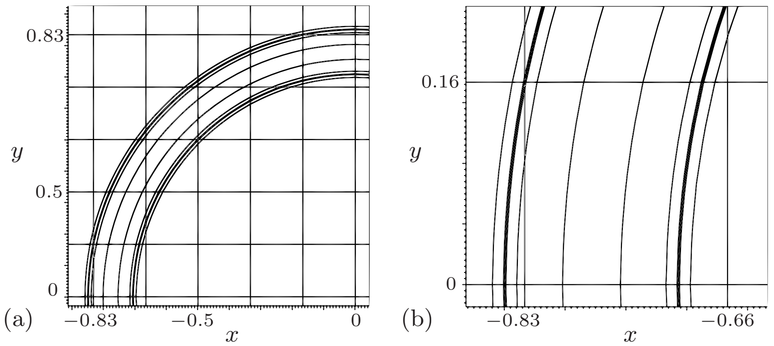

FIGURE 2. The grid and the level sets which are used for the condition number test. (a) Overall view on $x < 0/y > 0$ - quadrant. (b) Detailed view.

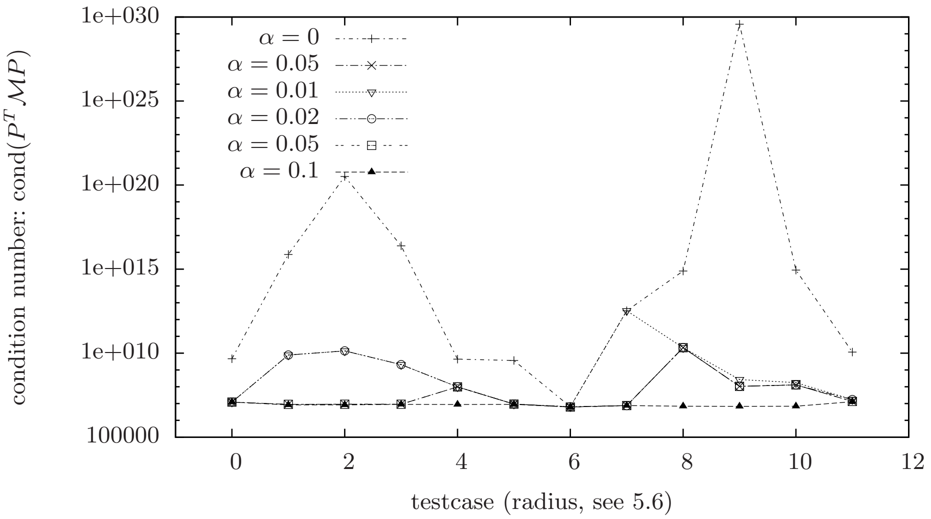

FIGURE 3. Condition number of $\mathcal{M}^{pc}$ over all tested radii for various agglomeration thresholds α (horizontal axis: different testcases, i.e., level-set radii, as lined up in sequence 5.6).

A final comment is needed on the evaluation of the curvature: since we chose a level - set function that can be exactly represented in the DG polynomial space and is in $\mathcal{C}^\infty(\Omega)$, the curvature can be evaluated exactly by Bonnet's formula. Therefore, the integral $\oint_{\mathfrak{I}} \kappa\sigma \boldsymbol{n}_{\mathfrak{I}} \,\mathrm{dS}$, which represents the force induced by surface tension, can be evaluated with sufficient accuracy. For unsteady problems, however, it can no longer be assumed that $\varphi \in \mathcal{C}^\infty(\Omega)$; indeed, by using a DG-based algorithm for level - set evolution, one cannot even assume $\varphi \in \mathcal{C}^0(\Omega)$. In such cases, a more sophisticated treatment of surface tension is required, to be investigated in future work.

For all test cases given below, the direct solver PARDISO (see Schenk 2002, 2004; Schenk & Gärtner 2006) is used to solve the discrete saddle-point problem which arises from the XDG-discretization of either the Stokes- (Eq. 5.3) or the Navier-Stokes (Eq. 3.9) problem.

6.1. *A static droplet*

First, we investigate a static droplet, where $a = b = 0.8$. The exact solution in this case is given by

$$\boldsymbol{u}_{ex} = (0, 0), \quad p_{ex} = 0 \text{ in } \mathfrak{A} \text{ and } p_{ex} = \frac{-\sigma}{0.8} \text{ in } \mathfrak{B}.$$

The purpose of this test case is to study "spurious velocities" introduced by the discretization in a static case. The material parameters for $\mathfrak{A}$ and $\mathfrak{B}$ are chosen to be those of water and air ($\sigma = 0.072$, $\rho_{\mathfrak{A}} = 10^3$, $\mu_{\mathfrak{A}} = 10^{-3}$ $\rho_{\mathfrak{B}} = 1.2$, $\mu_{\mathfrak{B}} = 17.1 \cdot 10^{-6}$). An 18×18-equidistant grid is chosen for the discretization of Ω, with an agglomeration factor $\alpha = 0.1$ and the DG polynomial degree $k = 3$. For the numerical solution $\boldsymbol{u}, p$ of the Stokes problem (Eq. 5.3) we obtain the bounds

$$\|\boldsymbol{u} - \boldsymbol{u}_{ex}\|_\infty \leq 1.6 \cdot 10^{-11} \text{ and } \|p - p_{ex}\|_\infty \leq 1.2 \cdot 10^{-12}.$$

6.2. *Convergence study on the Stokes problem*

We study the convergence order of the Stokes problem (Eq. 5.3) on a sequence of meshes with

$$6 \times 6, \;\; 12 \times 12, \;\; 24 \times 24, \;\; 48 \times 48 \text{ and } 96 \times 96$$

cells. In order to obtain a non-zero solution, we set

$$a = 0.8\sqrt{1.01} \approx 0.80399 \text{ and } b = 0.8\sqrt{0.99} \approx 0.79599,$$

while material parameters and agglomeration factor are set like in the previous section. A visualization of the velocity and pressure field is given in Figure 4. Since we are not aware of an exact solution to this problem, we assume $(\boldsymbol{u}_{ex}, p_{ex})$ to be the solution observed on the 96×96-cells grid. The convergence trend is plotted in Figure 5. As an average convergence order, we observe the following:

$k/k' = k - 1$	velocity	pressure	grids
2/1	2.57	1.51	12×12 – 48×48 .
3/2	3.51	1.25	12×12 – 48×48

While the convergence order for velocity falls into the expected range – at least greater than k – the convergence order for the pressure is disappointing. One reason may be that, for the chosen example, the pressure is almost piecewise-constant (see Figure 4) and therefore not ideal for convergence measurements.

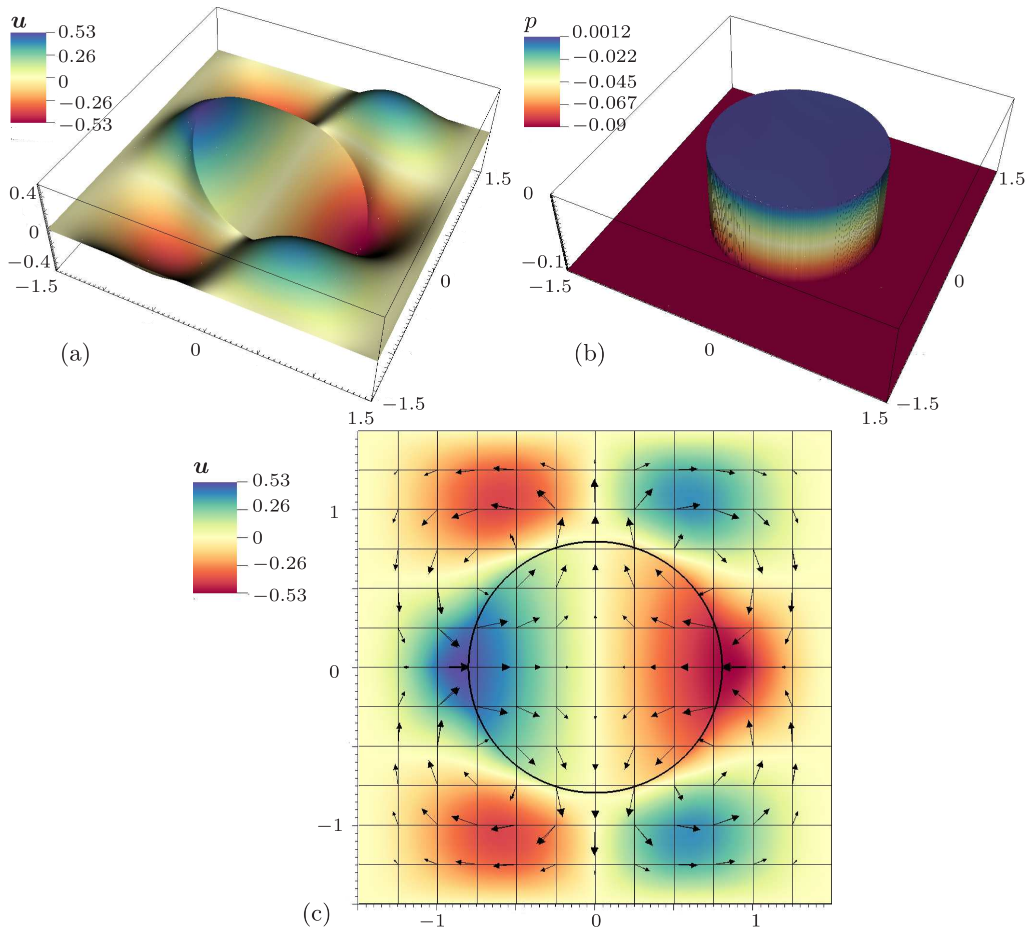

FIGURE 4. Velocity in x-direction (a), pressure (b) and velocity vectors/interface position/grid (c) for the steady ellipsoidal droplet.

6.3. *A steady Navier-Stokes solution*

Finally, we demonstrate the solution of a steady Navier-Stokes problem. A solution $(\boldsymbol{u}, p)$ of the nonlinear problem (3.9) is obtained as the limit of an iterative sequence

$$(\boldsymbol{u}_0, p_0) = 0, (\boldsymbol{u}_1, p_1), (\boldsymbol{u}_2, p_2), \ldots \rightarrow (\boldsymbol{u}, p). \tag{6.2}$$

For $\vartheta = 1, 2, \ldots$, the intermediate solution $(\boldsymbol{u}_{\vartheta+1}, p_{\vartheta+1})$ is found by an under-relaxed fix-point iteration:

$$\begin{aligned} \text{Find } (\tilde{\boldsymbol{u}}, \tilde{p}) \text{ s. t.} \qquad & L((\tilde{\boldsymbol{u}}, \tilde{p}), (\boldsymbol{v}, \tau)) + t(\boldsymbol{u}_\vartheta, \tilde{\boldsymbol{u}}, \boldsymbol{v}) = s(\boldsymbol{v}) \quad \forall (\boldsymbol{v}, \tau) \in V_k^\alpha \\ \boldsymbol{u}_{\vartheta+1} &= (1 - \beta_u)\tilde{\boldsymbol{u}} + \beta_u \ \boldsymbol{u}_\vartheta \\ p_{\vartheta+1} &= (1 - \beta_p)\tilde{p} + \beta_p \ p_\vartheta, \end{aligned}$$

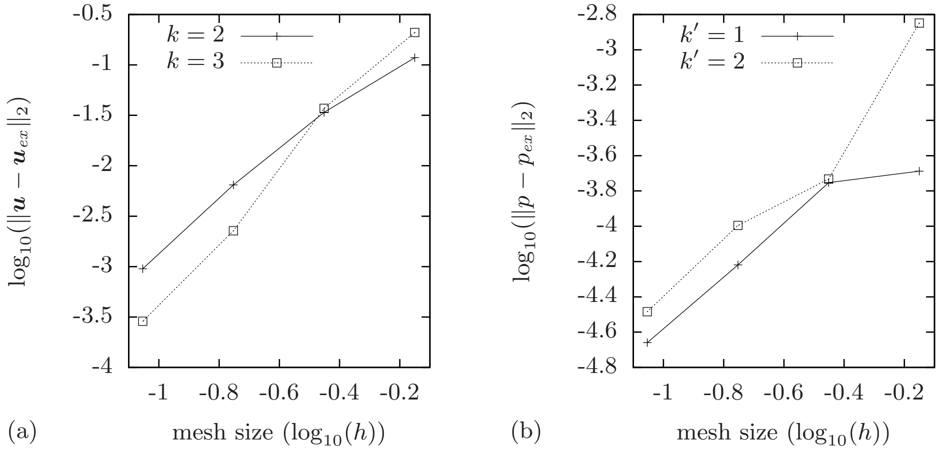

FIGURE 5. Convergence study on velocity (a, polynomial degree k) and pressure error (b, polynomial degree $k' = k - 1$). Horizontal axis: logarithm of grid resolution, i.e. $\log_{10}(h)$, where h denotes the grid resolution. Vertical axis: logarithm of error in the L^2-norm, i.e. $\log_{10}(\|\boldsymbol{u}-\boldsymbol{u}_{ex}\|_2)$ resp. $\log_{10}(\|p - p_{ex}\|_2)$.

where β_u and β_p denote the under-relaxation factors for velocity and pressure. The chosen parameters are

$$\sigma = 0.072,\ \rho_{\mathfrak{B}} = 1.2,\ \rho_{\mathfrak{A}} = 10^3,\ \mu_{\mathfrak{B}} = 10^{-3},\ \mu_{\mathfrak{A}} = 5.0 \cdot 10^{-3},$$
$$a = 0.8 \cdot \sqrt{1.0001},\ b = 0.8 \cdot \sqrt{0.9999},$$
$$\beta_u = 0.1,\ \beta_p = 0.1.$$

The convergence history of the iteration sequence (6.2) is displayed in Figure 6. Since the solution of the steady Navier-Stokes droplet behaves qualitatively in the same way as those of the steady Stokes droplet (see Figure 4), we do not provide separate plots of the solution.

7. Summary and outlook

The spatial discretization of a two-phase flow problem by means of a level-set/extended DG discretization has been demonstrated. The numerical results show a promising level of accuracy, especially on low-resolution grids, and indicate that the convergence properties of the DG method can be recovered in a two-phase setting by the XDG extension. Furthermore, cell-agglomeration seems to be an appropriate tool to handle “difficult” cut situations.

Several open topics remain to be resolved.

Of primary importance will be the investigation of dynamic problems in two dimensions, without surface tension. One test case may be the classic Rayleigh-Taylor instability.

Although not discussed here, the spatial discretization shows a high sensitivity to errors on the curvature. Preliminary results indicate that a level set evolution algorithm, where errors in the curvature do not accumulate in time, is required.

Furthermore, the performance of the implementation is still quite low to date. One reason is that the saddle-point problem is currently solved by a direct sparse solver; for

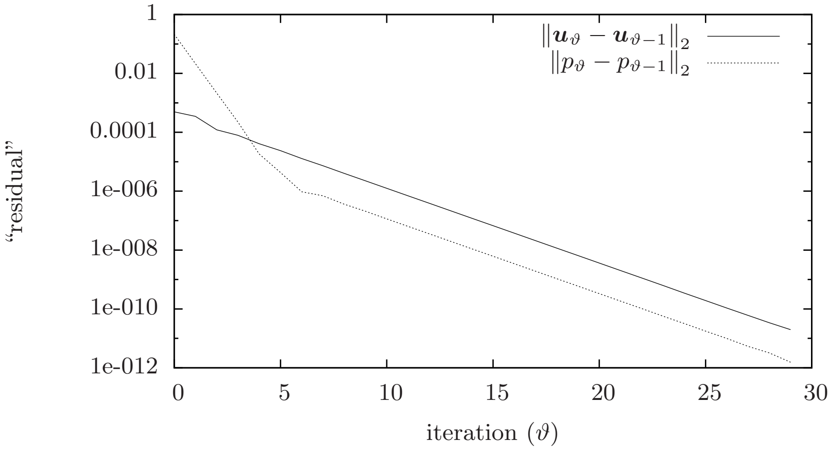

FIGURE 6. Convergence history of the velocity (solid line) and pressure (dotted line) "residual" over the first 30 iterations of the fix-point iteration. (By the therm "residual", we actually indicate the difference between two iterations, i.e., $\|\boldsymbol{u}_\vartheta - \boldsymbol{u}_{\vartheta-1}\|_2$, resp., $\|p_\vartheta - p_{\vartheta-1}\|_2$.)

larger systems, this approach is obviously not feasible, especially for three-dimensional problems. Future work is needed to address the issue of efficient preconditioning and solving of those systems.

Acknowledgements

This project is funded by the German Research Association (Deutsche Forschungsgemeinschaft, DFG) under fund KU-2719/1-1. I would like to thank Prof. Tim Warburton for fruitful and challenging discussions, as well as for his support for my visit at the CAAM department at Rice University, Houston, TX.

REFERENCES

ARNOLD, D. N. 1982 An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.* **19** (4), 742.

ARNOLD, D. N., BREZZI, F., COCKBURN, B. & MARINI, L. D. 2002 Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.* **39** (5), 1749–1779.

BABUŠKA, I. 1973 The finite element method with Lagrangian multipliers. *Numerische Mathematik* **20** (3), 179–192.

BREZZI, F. 1974 On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique* **8** (R2), 129–151.

DI PIETRO, D. A. & ERN, A. 2011 *Mathematical Aspects of Discontinuous Galerkin Methods*. *Mathématiques et Applications* 69. Springer.

GROSS, S. & REUSKEN, A. 2007 Finite element discretization error analysis of a surface tension force in two-phase incompressible flows. *SIAM J. Numer. Anal.* **45** (4), 1679–1700.

HESTHAVEN, J. S. & WARBURTON, T. 2008 *Nodal Discontinuous Galerkin Methods:*

Algorithms, Analysis, and Applications. *Texts in Applied Mathematics* 54. Springer-Verlag.

Massing, A., Larson, M. G., Logg, A. & Rognes, M. E. 2012 A stabilized Nitsche overlapping mesh method for the Stokes problem. *arXiv preprint arXiv:1205.6317* .

Müller, B., Kummer, F. & Oberlack, M. 2013 Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *Int. J. Numer. Meth. Eng.* pp. n/a–n/a.

Osher, S. J. & Fedkiw, R. P. 2002 *Level Set Methods and Dynamic Implicit Surfaces*. Springer.

Schenk, O. 2002 Two-level dynamic scheduling in PARDISO: improved scalability on shared memory multiprocessing systems. *Parallel Computing* **28** (2), 187–197.

Schenk, O. 2004 Solving unsymmetric sparse systems of linear equations with PARDISO. *Future Generation Computer Systems* **20** (3), 475–487.

Schenk, O. & Gärtner, K. 2006 On fast factorization pivoting methods for sparse symmetric indefinite systems. *Electronic Transactions on Numerical Analysis* **23**, 158–179.

Sethian, J. A. 1996 *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press.

Shahbazi, K. 2005 An explicit expression for the penalty parameter of the interior penalty method. *J. Comput. Phys.* **205**, 401–407.