Journal of Computational Physics 228 (2009) 5184–5206

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A sharp interface finite volume method for elliptic equations on Cartesian grids

M. Oevermann [a,*], C. Scharfenberg [b], R. Klein [b]

[a] Technische Universität Berlin, Institut für Energietechnik, Fasanenstr. 89, 10623 Berlin, Germany
[b] Freie Universität Berlin, Fachbereich Mathematik und Informatik, Arnimallee 14, 14195 Berlin, Germany

ARTICLE INFO

Article history:
Received 9 September 2008
Received in revised form 2 March 2009
Accepted 7 April 2009
Available online 24 April 2009

MSC:
35J25
65N12
65N30

Keywords:
Elliptic equations
Finite volume methods
Embedded interface
Variable and discontinuous coefficients
Discontinuous solution

ABSTRACT

We present a second order sharp interface finite volume method for the solution of the three-dimensional elliptic equation $\nabla \cdot (\beta(\vec{x})\nabla u(\vec{x})) = f(\vec{x})$ with variable coefficients on Cartesian grids. In particular, we focus on interface problems with discontinuities in the coefficient, the source term, the solution, and the fluxes across the interface. The method uses standard piecewise trilinear finite elements for normal cells and a double piecewise trilinear ansatz for the solution on cells intersected by the interface resulting always in a compact 27-point stencil. Singularities associated with vanishing partial volumes of intersected grid cells are removed by a two-term asymptotic approach. In contrast to the 2D method presented by two of the authors in [M. Oevermann, R. Klein, A Cartesian grid finite volume method for elliptic equations with variable coefficients and embedded interfaces, Journal of Computational Physics 219 (2006) 749–769] we use a minimization technique to determine the unknown coefficients of the double trilinear ansatz. This simplifies the treatment of the different cut-cell types and avoids additional special operations for degenerated interface topologies. The resulting set of linear equations has been solved with a BiCGSTAB solver preconditioned with an algebraic multigrid. In various testcases – including large β-ratios and non-smooth interfaces – the method achieves second order of accuracy in the L_∞ and L_2 norm.

1. Introduction

We seek solutions of the three-dimensional variable coefficient elliptic equation

$$\nabla \cdot (\beta(\vec{x})\nabla u(\vec{x})) = f(\vec{x}), \quad \vec{x} \in \overline{\Omega} \setminus \Gamma \tag{1}$$

defined in a domain $\overline{\Omega} \setminus \Gamma$ with an embedded interface Γ. For simplicity we assume $\overline{\Omega}$ to be a simple cuboid. The embedded interface Γ separates two disjoint sub-domains $\overline{\Omega}^+$ and $\overline{\Omega}^-$ with $\overline{\Omega} = (\overline{\Omega}^+ \cup \overline{\Omega}^-)$, see Fig. 1 for an illustration. Along the interface we prescribe jump conditions for the solution

$$[\![u]\!]_\Gamma = u^+(\vec{x}) - u^-(\vec{x}) = g(\vec{x}_\Gamma) \tag{2}$$

and for its gradient in the normal direction

$$[\![\beta u_n]\!]_\Gamma = \beta^+ u_n^+ - \beta^- u_n^- = h(\vec{x}_\Gamma), \tag{3}$$

with the notation $u_n = (\nabla u \cdot \vec{n})$. The unit normal vector $\vec{n}$ on Γ is defined to point from $\overline{\Omega}^+$ to $\overline{\Omega}^-$.

* Corresponding author. Tel.: +49 3031422452; fax: +49 3031422157.
E-mail address: michael.oevermann@tu-berlin.de (M. Oevermann).

0021-9991/
doi:10.1016/j.jcp.2009.04.018

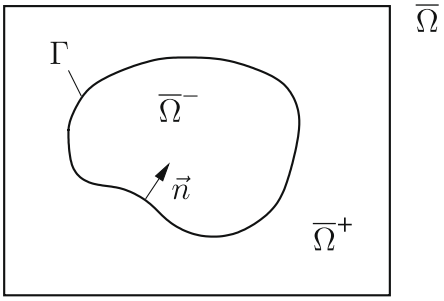

Fig. 1. Domain $\overline{\Omega}$ with sub-domains $\overline{\Omega}^+$, $\overline{\Omega}^-$, and embedded interface Γ.

Elliptic equations of type (1) with variable and discontinuous coefficients and solution discontinuities often arise as a component in modelling physical problems with embedded boundaries. Examples include incompressible two-phase flow with surface tension featuring jumps in pressure and pressure gradient across the interface, projection methods for zero Mach-number premixed combustion with jumps in the dynamic pressure and pressure gradient across the flame front, heat conduction between materials of different heat capacity and conductivity and interface diffusion processes. In the literature one can find a vast number of different approaches for the numerical solution of this type of problem. However, we limit our discussion here to methods on grids which are not aligned with the interface. These methods have the advantage that they do not need any re-meshing if the interface moves.

In Peskin's immersed boundary method [31], singular forces arising from discontinuous coefficients and jump conditions are treated as delta functions. Using discretised discrete delta functions, the discontinuity is spread over several grid cells making the method first order accurate. The method has been used for many problems in mathematical biology and fluid mechanics. Cortez and Minion [3] considerably improved Peskin's method by improving its accuracy through higher order procedures for representing boundary forces. Recent work by Tornberg and Engquist [38,39,5] generalizes the immersed boundary approach and allows for high order approximations with minimal distribution of discontinuities or singular source terms over the computational grid.

Mayo [25,26] presented a second order accurate method for Poisson's equation and the biharmonic equation on irregular domains using an integral equation formulation. The resulting Fredholm integral equations of the second kind are solved with a fast Poisson solver on a rectangular region. Although the method captures solution discontinuities at the embedded interface, continuous derivatives have been assumed to evaluate the discrete Laplacian. The method can easily be extended to fourth order accuracy.

The immersed interface method [16–18,20] is a second order finite difference method on Cartesian grids for second order elliptic and parabolic equations with variable coefficients. Discontinuities in the solution and the normal gradient at the interface are explicitly incorporated into the finite difference stencil. Second order has been achieved by including additional points near the interface into the standard 5-point stencil leading to a non-standard six-point stencil in 2D. The resulting linear equation system is sparse but not symmetric or positive definite. Based on the immersed interface method Li and Ito [19] present a second order finite difference method which satisfies the sign property on the matrix coefficients which guarantees the discrete maximum principle. The resulting linear system of equations is non-symmetric but diagonally dominant and its symmetric part is negative definite. The ideas presented in [19] have been extended to 3D in [4].

A first order finite difference method on Cartesian grids was presented by Liu et al. [22]. Interface jump conditions are explicitly incorporated into the finite difference stencil as in the immersed interface method. Applying a one-dimensional approach in each spatial direction by implicitly smearing out the gradient jump condition, standard stencils (5-point in 2D, 9-point in 3D) for the discrete Laplacian are achieved leading to a symmetric positive definite matrix for the Poisson equation. The method shows first order accuracy for the solution u in the L^∞-norm for constant coefficients $\beta^\pm$. A convergence proof of the method has been provided in [23] based on the weak formulation of the problem. Due to its simplicity and robustness the methods has been used in many engineering and scientific problems. The method has been independently developed and applied to incompressible two-phase flow in [29].

A fourth order accurate finite difference method for elliptic problems with complex boundaries has been developed by Gibou and Fedkiw in [7]. By high order extrapolation of the solution outside the domain they were able to apply high order finite difference formulas at and near the interface. Similar ideas have been used in a series of papers by Wei and coworkers [44,43,42] for elliptic problems with embedded interfaces. They developed finite difference methods of up to sixth order in 3D for smooth interfaces and up to second order for complex interfaces with sharp edges, wedges, and tips. Their methods can be viewed as a higher-order generalization of the immersed interface method. Solutions on both sides of the interface are smoothly extended beyond the interface allowing the application of standard high order finite difference formulas.

One of the first methods to model discontinuities in the finite element framework without aligning the grid with the interface has been presented in [27,1]. In the so-called extended finite element method the original finite element space is enriched by additional basis functions introducing new unknowns to the problems. The choice of additional enrichment functions depends on the type of discontinuity, e.g. step functions for solution discontinuities or distance functions for kinks

[1]. Related approaches have been presented in [9,10]. The extended finite element method has been recently applied by Groß and Reusken to model the pressure discontinuity arising from surface tension in incompressible two-phase flow [8]. They observed locally slightly better than first order of accuracy for the pressure. However, in their method they did not make use of the jump condition of the pressure gradient in normal direction.

A finite element method on triangular meshes for solving second order elliptic and parabolic equations for interface problems with $[u] = 0$ and $[\beta u_n] \neq 0$ has been proposed by Chen and Zou [2]. In their method the triangles are aligned with the interface. In the L_2-norm nearly second order accuracy ($h^2|\log h|$) has been proved. The resulting linear system of equations is symmetric and positive definite. Another finite element method based on uniform triangulations of Cartesian grids was presented by Li et al. [21]. In contrast to [2], the triangles need not to be aligned with the interface. Numerical results with non-conforming finite elements demonstrate slightly less than second order of accuracy in L_∞ and second order of accuracy with conforming finite elements for a problem with homogeneous jump conditions $[u] = 0$, $[\beta u_n] = 0$. The general case with variable coefficients and non-homogeneous interface conditions $[u] \neq 0$, $[\beta u_n] \neq 0$ has been tackled recently by Hou and Liu [11] with a finite element method. Similar to [21] they use uniform triangulations of Cartesian grids. Their method is second order accurate in L_∞ if the solution u is C^2 and the interface is C^2 or C^1. To our knowledge these methods have not been extended to three spatial dimensions.

Johansen and Colella [12] developed a second-order finite volume method on Cartesian grids for the variable coefficient Poisson equation on irregular domains with Dirichlet and Neumann boundary conditions and combined the method with an adaptive mesh refinement. Using central differencing for the gradients, their method reproduces the standard five-point stencil on regular cells. Using linear interpolation of gradients for internal edges and quadratic polynomials in normal direction to the boundary for irregular cells leads to a non-standard stencil. The final linear system is non-symmetric. Although remotely related to our work in the sense of using a finite volume method, the authors did not consider embedded boundaries with jump conditions of the solution and the normal derivative. To our knowledge, their method has not been extended to 3D. Furthermore, their method does not allow partial volumes less than 10^{-6} times the normal cell volume.

Recently the authors presented in [28] a sharp interface finite volume method on Cartesian grids for the solution of the variable coefficient Poisson equation with solution discontinuities across an embedded interface in two spatial dimensions. Using a dual bilinear solution ansatz on cells intersected by the interface the method achieves locally second order of accuracy. Singularities arising from vanishing partial volumes and certain positions of the interface relative to the underlying grid are removed in [28] by a two-term asymptotic approach.

In this paper we extend the ideas presented in [28] to 3D. The piecewise bilinear solution ansatz of our 2D method is replaced by a piecewise trilinear ansatz. In contrast to the 2D method, where we impose the jump conditions at selected points of the interface, here we use a minimization approach for the incorporation of the jump conditions and the determination of the 16 unknown coefficients of the dual piecewise trilinear solution ansatz. This leads to a unified treatment of the different types of cut-cells and avoids additional special operations for degenerated interface topologies. The only singularity arising from vanishing partial volumes of intersected cells is removed by an asymptotic approach. The asymptotic treatment of this singularity leads to a robust method which allows vanishing partial volumes down to the machine accuracy without affecting the condition number of the minimization problem. In various examples including high β ratios, complex and non-smooth interfaces the method shows locally second order of accuracy.

The vast majority of sharp interface methods on Cartesian grids is based on finite difference methods. However, in many engineering problems such as fluid flow or heat conduction the governing equations are often derived by a control volume analysis with appropriate flux boundary and interface conditions. This point of view has led in a natural way to many conservative finite volume methods. The motivation for the finite volume approach presented in this work stems from our interest in conservative finite volume projection methods for Zero- and Low-Mach-number flow [32,15,13], with first-order accurate versions of front tracking methods for flames and contact discontinuities presented in [37,14,34,33]. The divergence constraint of the velocity field leads to an elliptic equation for the pressure in a finite volume form. The use of piecewise trilinear ansatz function for the solution u makes our method quite similar to finite element methods and allows us to construct improved exact projection methods [40,41].

Compared to the cited literature, our method differs in the following points: (i) we use a finite volume method instead of finite difference [16,22,42] or finite elements [11,21,27,1,8], (ii) compared to the second and higher order finite difference methods [16,42] we achieve always automatically a compact 27-point stencil without explicit incorporation of additional points near the interface.

In comparison to the finite element method presented in, e.g. [11,8], we present a trilinear finite element which does not develop singularities for vanishing partial volumes of intersected cells. In contrast to the cited finite element methods our method results in a non-symmetric matrix. In case of constant and equal coefficients we have a symmetric and positive definite matrix.

2. Finite volume formulation

Integrating Eq. (1) over an arbitrary control volume $\Omega \in \overline{\Omega}$ leads to

$$\int_\Omega \nabla \cdot (\beta \nabla u)\, dV = \int_\Omega f\, dV.$$

For a control volume $\Omega = \Omega^+ \cup \Omega^-$ intersected by the interface Γ we write

$$\int_{\Omega^+} \nabla \cdot (\beta^+ \nabla u^+)\, dVa + \int_{\Omega^-} \nabla \cdot (\beta^- \nabla u^-)\, dVa = \int_{\Omega} f\, dV.$$

After applying the divergence theorem on both integrals on the left hand side we get

$$\int_{\partial\Omega} \beta(\nabla u \cdot \vec{n})\, dS = \int_{\Omega} f\, dV - \int_{\Gamma_\Omega} [\![\beta u_n]\!]\, dS, \tag{4}$$

where Γ_Ω denotes the part of the embedded interface Γ lying inside Ω and $\partial\Omega = (\partial\Omega^+ \cup \partial\Omega^-) \setminus \Gamma_\Omega$. For $\Gamma_\Omega \neq 0$ we have for the source term

$$\int_{\Omega} f\, dV = \int_{\Omega^+} f^+\, dV + \int_{\Omega^-} f^-\, dV. \tag{5}$$

For a regular control volume without an embedded interface we have either $\Omega = \Omega^+ \in \overline{\Omega}^+ \wedge \Omega^- \equiv 0$ or $\Omega = \Omega^- \in \overline{\Omega}^- \wedge \Omega^+ \equiv 0$ and the last integral on the right hand side of (4) vanishes. In the following sections we describe our finite volume method to solve Eqs. (4) and (5).

3. Numerical method

We discretise Eq. (4) on a uniform Cartesian grid in three-dimensional space. Let Δx, Δy, Δz be the grid spacing in x, y, and z-direction, respectively, see Fig. 2. The values $u_{i,j,k}$ of our discretised solution are located at grid nodes with the coordinates $x_{i,j,k} = x_0 + i\Delta y$, $y_{i,j,k} = y_0 + j\Delta y$, and $z_{i,j,k} = z_0 + k\Delta z$. The control volumes $\Omega_{i,j,k}$ are centered around the corresponding grid nodes (i,j,k) having edges of length Δx, Δy, and Δz. The cuboids defined by the Cartesian grid itself are called cells in this work. We denote the cell defined by grid nodes (i,j,k), $(i+1,j,k)$, $(i+1,j+1,k)$, $(i,j+1,k)$, $(i,j,k+1)$, $(i+1,j,k+1)$, $(i+1,j+1,k+1)$, and $(i,j+1,k+1)$ as $C_{i,j,k}$, see Fig. 3. Let $\mathfrak{C}_{i,j,k}$ be the set of cells with joint grid node (i,j,k), i.e. those cells having a partial volume in $\Omega_{i,j,k}$. Furthermore, we denote with S_i^c, $c \in \mathfrak{C}_{i,j,k}$, $i \in 1, 2, 3$ the three parts of the surface of $\Omega_{i,j,k}$ lying in cell c, see Fig. 3 for an illustration. As an example, the faces with numbers 1, 8, and 11 in Fig. 3 are parts of the surface of $\Omega_{i,j,k}$.

With the notations introduced above we can now write the discrete form of (4) for control volume $\Omega_{i,j}$ as

$$\sum_{c \in \mathfrak{C}_{i,j,k}} \sum_{i=1}^{3} \int_{S_i^c} \beta(\nabla u \cdot \vec{n})\, dS = \int_{\Omega_{i,j,k}} f\, dV - \int_{\Gamma_{\Omega_{i,j,k}}} [\![\beta u_n]\!]\, dS. \tag{6}$$

To evaluate the left hand side of (6) we approximate the solution u with piecewise trilinear ansatz on each cell $c \in \mathfrak{C}_{i,j,k}$ of the grid. This allows us to evaluate the surface integrals on the left hand side of (6) analytically in terms of the node values of u and, in case of an intersected control volume, appropriate jump conditions at the interface.

4. Interface representation

We represent the interface Γ with a standard levelset approach [35,30] where the interface is implicitly defined by the zero level of a signed distance function. Geometric quantities such as interface normal vectors at the interface within a cell

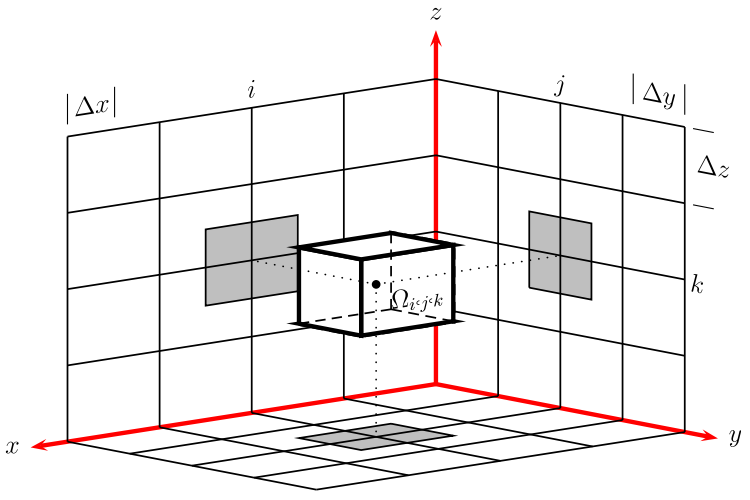

Fig. 2. Control volume $\Omega_{i,j,k}$. Discrete solution values are located at grid nodes i,j,k.

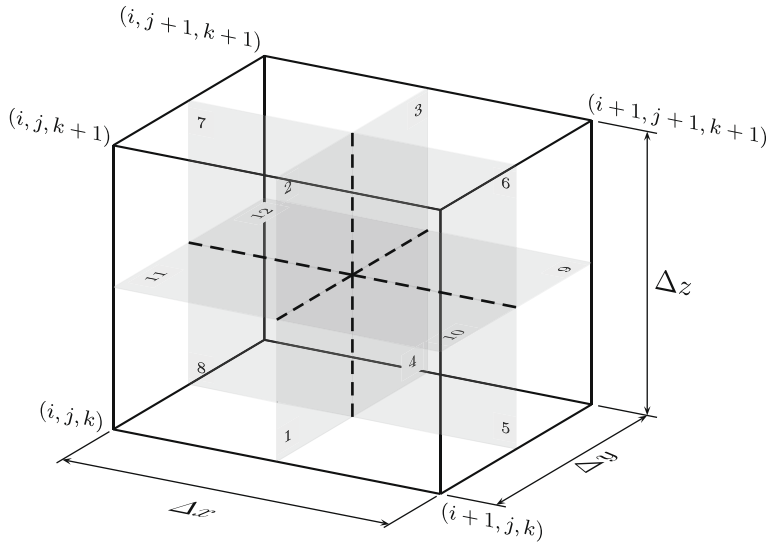

Fig. 3. Cell $C_{i,j,k}$ with node indices. The grey shaded internal faces with indices 1–12 are surface parts of the control volumes having a partial volume within the cell. The eight cuboids formed by the internal faces of the cell with edge lengths $\Delta x/2$, $\Delta y/2$, and $\Delta z/2$ belong to different control volumes and are called sub-cells.

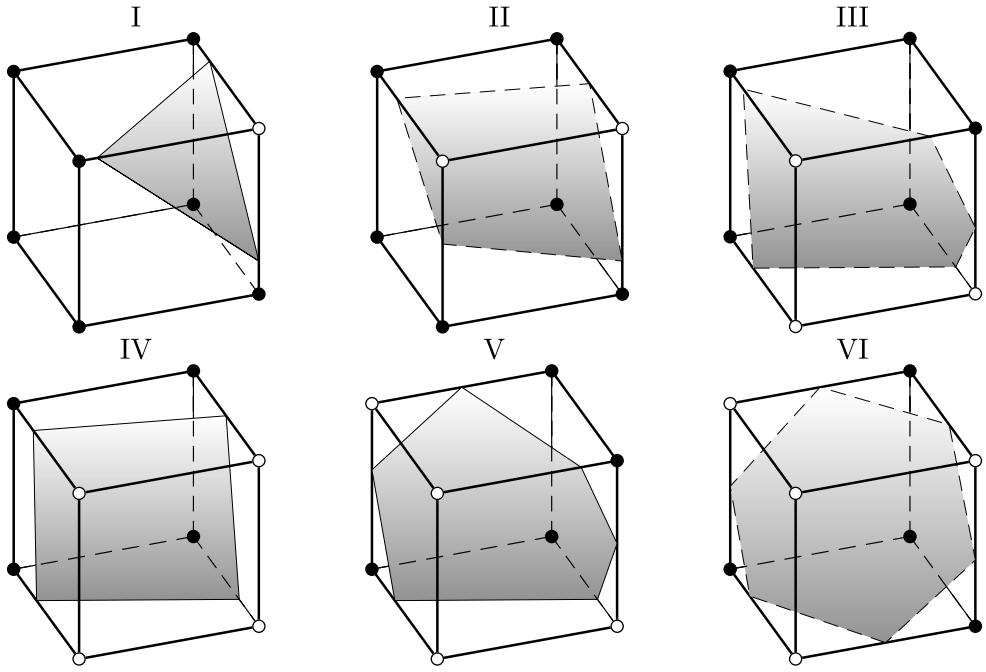

Fig. 4. Different types of cut-cells considered in this work. Nodes marked with ○ and •, respectively, are on the same side of the interface.

(i, j, k) are calculated via trilinear interpolation of second order central difference approximations at the nodes of the grid. In this study we only consider cells which are cut by a single interface. Under this constraint we can identify six different types of cut-cells which are qualitatively sketched in Fig. 4. Although we allow only cells with a single interface, control volumes might still be intersected by more than one interface.

For a unique discretisation of the interface we introduce a triangulation of the interface within each cell in the following manner: In the first step we calculate the intersections of the interface with the edges of the cell by linear interpolation of the levelset values between the nodes of the cell. In the second step we calculate a point on the surface in the interior of the cell by averaging the coordinates of the intersections determined in the first step. A third step could be the projection of this point in normal direction onto the interface assuming a trilinear distribution of the levelset function on the cell. However, in our numerical tests we did not see a noteworthy improvement of the results and therefore we left this step.

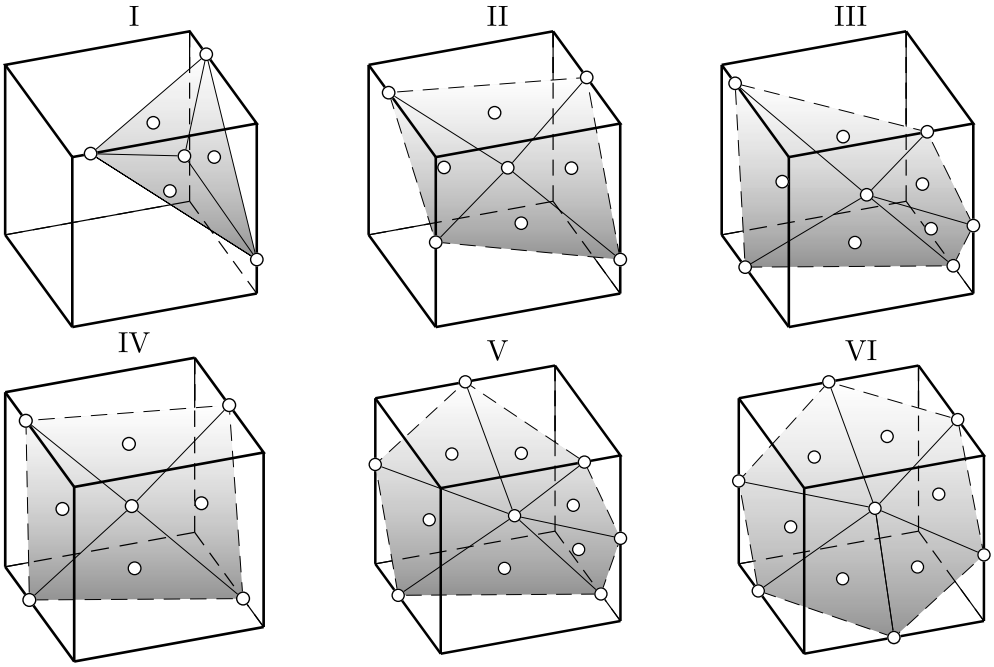

Fig. 5. Interface discretisation by triangulation for the different types of cut-cells. Discrete jump conditions are applied at the triangle corner points and the triangle barycenters which are marked with circles.

The connection of the interface midpoint point of step two with all intersections of the interface with the cell edges defines a unique triangulation of the interface within a cell. Fig. 5 shows triangulated interfaces for the different cut-cell types considered in this work. The chosen triangulation of the interface is a slight variation of the famous marching cubes algorithm [24] in the sense that we use an additional point on the interface within the cell which allows a unique representation of the interface with triangles.

In addition to the cells composed by the Cartesian grid itself we introduce so called sub-cells. Sub-cells are the (eight) parts of a cell belonging to different control volumes, see Fig. 3.

5. Piecewise trilinear ansatz

For the evaluation of the left hand side of Eq. (6) we approximate the solution u by piecewise trilinear solution ansatz functions on cells. In the following presentation we need to distinguish between cells which are intersected by an interface and those who are not.

5.1. Normal cells

We denote cells without an intersection with the interface as normal cells, and introduce the following set of local orthogonal coordinates, see Fig. 6:

$$\vec{\xi} = \begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} = \begin{bmatrix} \frac{x - x^1_{(i,j,k)}}{\Delta x} \\ \frac{y - y^1_{(i,j,k)}}{\Delta y} \\ \frac{z - z^1_{(i,j,k)}}{\Delta z} \end{bmatrix}, \tag{7}$$

where x, y, z are global coordinates and $\vec{x}^1_{(i,j,k)} = [x^1_{(i,j,k)}, y^1_{(i,j,k)}, z^1_{(i,j,k)}]^T$ denotes the global coordinates of node number one with local coordinates $\xi = \eta = \zeta = 0$, see Fig. 6. Obviously, we have $\xi, \eta, \zeta \in [0, 1]$ within a cell.

At normal cells we apply a trilinear solution ansatz $u(\xi, \eta, \zeta)$ within a cell (i, j, k). Using standard finite element shape functions we have

$$u(\xi, \eta, \zeta) = \underline{x} \cdot \underline{a} \tag{8}$$

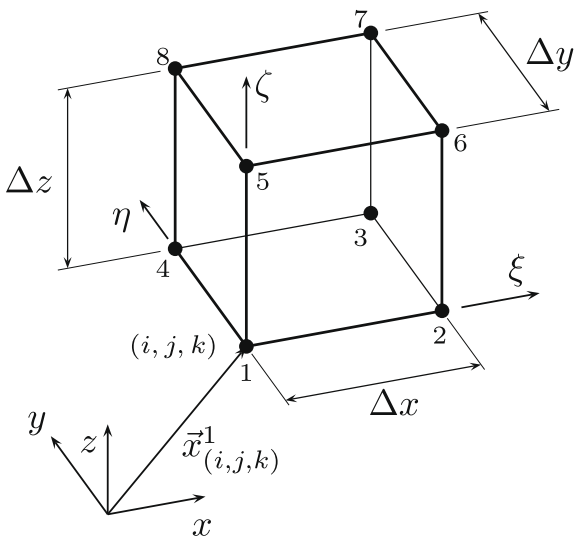

Fig. 6. Node numbering and local coordinate system for a computational cell (i,j,k). For normal cells and non-singular cut-cells the origin of the local coordinate system used for the trilinear solution ansatz is always at node number 1.

with

$$\underline{x} = \begin{bmatrix} (1-\xi) & (1-\eta) & (1-\zeta) \\ \xi & (1-\eta) & (1-\zeta) \\ \xi & \eta & (1-\zeta) \\ (1-\xi) & \eta & (1-\zeta) \\ (1-\xi) & (1-\eta) & \zeta \\ \xi & (1-\eta) & \zeta \\ \xi & \eta & \zeta \\ (1-\xi) & \eta & \zeta \end{bmatrix}^T \tag{9}$$

and

$$\underline{a} = [u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8]^T. \tag{10}$$

Here u_i denotes the solution value at node i of the computational cell (i,j,k), see Fig. 6 for the node numbering used. We have introduced the notation $\underline{a} \cdot \underline{b}$ for the inner product of two arbitrary vectors $\underline{a}$ and $\underline{b}$. In the case of a normal cell the eight node values u_i of the solution uniquely determine the trilinear ansatz.

With the trilinear ansatz (8) defined in each normal cell of the computational grid we can evaluate the surface integrals on the left side of (6) analytically in terms of the unknown nodal values. With $\beta = 1$ and $\Delta x = \Delta y = \Delta z = 1$ we get a 27-point stencil with weights as shown in Fig. 7. The stencil shown in Fig. 7 is the 3D analogue of the second order stencil of Süli [36], who proved stability and convergence of the scheme.

5.2. Cut cells

On cells intersected by the interface we use separate trilinear solution ansatz functions on both sides of the interface. We differentiate between cells with a non-vanishing interface area, called non-singular cut-cells, and cut-cells with potential singularities through vanishing interface areas and/or vanishing partial volumes. For the following presentation we assume that node number 1, Fig. 6, lies within Ω^+.

5.2.1. Non-singular cut-cells

On non-singular cut-cells we apply a dual piecewise trilinear ansatz in the form

$$\begin{aligned} u^+(\vec{\xi}) &= \underline{x} \cdot \underline{a}, \\ u^-(\vec{\xi}) &= \underline{x} \cdot \underline{b}, \end{aligned} \tag{11}$$

with

$$\underline{a} = [a_1, \ldots, a_8]^T \quad \text{and} \quad \underline{b} = [b_1, \ldots, b_8]^T, \tag{12}$$

and a shape function vector $\underline{\xi}$ as given in (9)

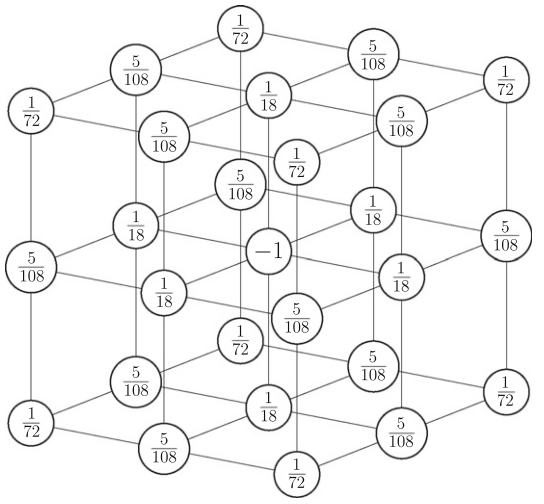

Fig. 7. Stencil elements for a control volume with $\beta = 1$, $\Delta x = \Delta y = \Delta z = 1$ where all adjacent cells are normal cells not cut by an interface.

In (11) we have defined the solutions on both sides of the interface using the same shape functions as used for normal cells. That means the 16 unknown coefficients $a_1, \ldots, a_8$ and $b_1, \ldots, b_8$ are the nodal values of the solutions u^+ and u^- at the grid nodes 1–8 of cell (i, j, k). Depending on which side of the interface the node exists these values are either real discrete solution values or extrapolated values from the other side of the interface. In order to determine the 16 unknown coefficients of the double trilinear solution ansatz we need – apart from the eight nodal values – additional constraints. These constraints will be derived from the prescribed jump conditions for the solution and its gradient in normal direction at the interface.

The gradients of u in direction of $\vec{n} = [n_x, n_y, n_z]^T$ follow directly from (11) and can be written as

$$\begin{aligned} (\nabla u^+ \cdot \vec{n}) &= u_n^+(\vec{\xi}) = \underline{y} \cdot \underline{a}, \\ (\nabla u^- \cdot \vec{n}) &= u_n^-(\vec{\xi}) = \underline{y} \cdot \underline{b}, \end{aligned} \tag{13}$$

with

$$\underline{y} = \frac{\partial \underline{x}}{\partial x} n_x + \frac{\partial \underline{x}}{\partial y} n_y + \frac{\partial \underline{x}}{\partial z} n_z. \tag{14}$$

The partial derivatives

$$\frac{\partial \underline{x}}{\partial x} = \frac{1}{\Delta x} \frac{\partial \underline{x}}{\partial \xi}, \quad \frac{\partial \underline{x}}{\partial y} = \frac{1}{\Delta y} \frac{\partial \underline{x}}{\partial \eta}, \quad \text{and} \quad \frac{\partial \underline{x}}{\partial z} = \frac{1}{\Delta z} \frac{\partial \underline{x}}{\partial \zeta}$$

are obtained by simple differentiation of (9). As an example, for the gradient in x-direction we have

$$\frac{\partial \underline{x}}{\partial x} = \frac{1}{\Delta x} \frac{\partial \underline{x}}{\partial \xi} = \frac{1}{\Delta x} \begin{bmatrix} -(1-\eta) & (1-\zeta) \\ -(1-\eta) & (1-\zeta-) \\ -\eta & (1-\zeta) \\ -\eta & (1-\zeta) \\ -(1-\eta) & \zeta \\ -(1-\eta) & \zeta \\ -\eta & \zeta \\ -\eta & \zeta \end{bmatrix}^T.$$

In addition to the eight nodal values $u_i,\ i = 1, \ldots, 8$ we provide jump conditions for the solution and the gradient in normal direction at the following points of the triangulated interface: (a) the triangle corner points and (b) the triangle barycenters, see Fig. 5. As seen in Fig. 5, the total number N of discrete points on the interface at which we prescribe jump conditions varies between 7 for cut-cells of type I and 13 for cut cells of type V and VI. Given two jump conditions per point (one for the solution and one for normal derivative), we have eight corner values plus 14 to 26 jump constraints to determine the 16 unknowns coefficients. We solve this overdetermined system by minimizing the difference between the prescribed and the calculated jump conditions under the constraint of exact nodal values.

We denote with $\vec{\xi}_k = [\xi_k, \eta_k, \zeta_k]^T$, $k = 1, \ldots, N$ the local coordinates of the kth point on the triangulated interface where we provide discrete values of jump conditions. The differences between the calculated jumps $\widehat{[\![u]\!]}$ and $\widehat{[\![\beta u_n]\!]}$ from the trilinear

ansatz and the prescribed jump values $[\![u]\!]$ and $[\![\beta u_n]\!]$ can be written in the form of jump residuals $\tilde{\underline{r}}_1 = [r_{1,1}, \ldots, r_{1,N}]^T$ and $\tilde{\underline{r}}_2 = [r_{2,1}, \ldots, r_{2,N}]^T$. For the jump residual in u we have

$$\tilde{r}_{1,k} = \widehat{[\![u]\!]}_k - [\![u]\!]_k = ((\underline{x}_k \cdot \underline{a}) - (\underline{x}_k \cdot \underline{b})) - [\![u]\!]_k \tag{15}$$

and for the gradient jump residual

$$\tilde{r}_{2,k} = \widehat{[\![\beta u_n]\!]}_l - [\![\beta u_n]\!]_k = (\beta^+(\underline{y}_k \cdot \underline{a}) - \beta^-(\underline{y}_k \cdot \underline{b})) - [\![\beta u_n]\!]_k \tag{16}$$

with $k = 1, \ldots, N$. Depending on the location of the interface, the grid spacing, and the values of β^+ and β^- the individual components of $\tilde{\underline{r}}_1$ and $\tilde{\underline{r}}_2$ can have different orders of magnitude. To give each jump condition a comparable influence on the solution of the minimization problem we introduce scaled residuals

$$r_{1,k} = w_{1,k}\tilde{r}_{1,k} \quad \text{and} \quad r_{2,k} = w_{2,k}\tilde{r}_{2,k} \tag{17}$$

with weights

$$w_{1,k} = \|[\underline{x}_k,\ \underline{x}_k]\|_2^{-1} \quad \text{and} \quad w_{2,k} = \|[\beta^{\mathfrak{A}}\underline{y}_k,\ \beta^{\mathfrak{B}}\underline{y}_k]\|_2^{-1}. \tag{18}$$

The scaled jump residuals (17) can be written in compact matrix notation as

$$\underline{r} = \underline{\underline{X}}\,\underline{c} - \underline{p} \tag{19}$$

with

$$\underline{\underline{X}} = \begin{bmatrix} \underline{\underline{X}}_\Gamma & -\underline{\underline{X}}_\Gamma \\ \beta^+\underline{\underline{Y}}_\Gamma & -\beta^-\underline{\underline{Y}}_\Gamma \end{bmatrix}, \quad \underline{c} = \begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix}, \quad \underline{p} = \begin{bmatrix} \underline{v} \\ \underline{g} \end{bmatrix},$$

$$\underline{v} = \begin{bmatrix} w_{1,1}[\![u]\!]_1 \\ \vdots \\ w_{1,N}[\![u]\!]_N \end{bmatrix}, \quad \underline{g} = \begin{bmatrix} w_{2,1}[\![\beta u_n]\!]_1 \\ \vdots \\ w_{2,N}[\![\beta u_n]\!]_N \end{bmatrix},$$

and

$$\underline{\underline{X}}_\Gamma = \begin{bmatrix} w_{1,1}\underline{x}_1 \\ \vdots \\ w_{1,N}\underline{x}_N \end{bmatrix}, \quad \underline{\underline{Y}}_\Gamma = \begin{bmatrix} w_{2,1}\underline{y}_1 \\ \vdots \\ w_{2,N}\underline{y}_N \end{bmatrix}.$$

For completeness we note that $\underline{\underline{X}} \in \mathbb{R}^{2N\times 16}$, $\underline{\underline{X}}_\Gamma,\ \underline{\underline{Y}}_\Gamma \in \mathbb{R}^{N\times 8}$, $\underline{a} \in \mathbb{R}^{8\times 1}$, $\underline{b} \in \mathbb{R}^{8\times 1}$, $\underline{c} \in \mathbb{R}^{16\times 1}$ and $\underline{p} \in \mathbb{R}^{2N\times 1}$. Eq. (19) is a set of $2N$ linear equations for the 16 unknown coefficients $\underline{a}$ and $\underline{b}$.

In our minimization approach we seek solutions for the coefficients $\underline{c} = [\underline{a}, \underline{b}]^T$ under the constraint of exact nodal values $\underline{u} = [u_1, \ldots, u_8]^T$. This constraint can be formulated as

$$\underline{\underline{B}}\,\underline{c} - \underline{u} = \underline{0}, \tag{20}$$

with

$$\underline{\underline{B}} = [\underline{\underline{B}}^+ \ \ \underline{\underline{B}}^-] \tag{21}$$

and

$$\underline{\underline{B}}^\pm = \mathrm{diag}(b_1^\pm, \ldots, b_8^\pm). \tag{22}$$

The elements of the diagonal matrices $\underline{\underline{B}}^+$ and $\underline{\underline{B}}^-$ are

$$b_i^\pm = \begin{cases} 1, & \text{if} \quad u_i \in \Omega^\pm, \\ 0, & \text{else}, \end{cases} \tag{23}$$

where u_i is the solution at node number i in cell (i, j, k).

Our constraint linear minimization problem may now be formulated using Lagrange multipliers:

$$L(\underline{c}, \underline{\lambda}) = \frac{1}{2}(\underline{r} \cdot \underline{r}) + \underline{\lambda} \cdot (\underline{\underline{B}}\,\underline{c} - \underline{u}) = \min. \tag{24}$$

A necessary condition for a minimum of $L(\underline{c}, \underline{\lambda})$ are vanishing partial derivatives

$$\frac{\partial L}{\partial \underline{c}} = 0 \quad \text{and} \quad \frac{\partial L}{\partial \underline{\lambda}} = 0.$$

This leads to following set of linear equations:

$$\underbrace{\begin{bmatrix} \underline{\underline{X}}^T\underline{\underline{X}} & \underline{\underline{B}}^T \\ \underline{\underline{B}} & \underline{\underline{0}} \end{bmatrix}}_{\underline{\underline{M}}} \begin{bmatrix} \underline{c} \\ \underline{\lambda} \end{bmatrix} = \begin{bmatrix} \underline{\underline{X}}^T\underline{p} \\ \underline{u} \end{bmatrix} \tag{25}$$

with the solution

$$\begin{bmatrix} \underline{c} \\ \underline{\lambda} \end{bmatrix} = \underbrace{\begin{bmatrix} \underline{\underline{M}}_{11}^{-1} & \underline{\underline{M}}_{12}^{-1} \\ \underline{\underline{M}}_{21}^{-1} & \underline{\underline{M}}_{22}^{-1} \end{bmatrix}}_{\underline{\underline{M}}^{-1}} \begin{bmatrix} \underline{\underline{X}}^T\underline{p} \\ \underline{u} \end{bmatrix},$$

where the $\underline{\underline{M}}_{ij}^{-1}$ are sub-matrices of $\underline{\underline{M}}$ with sizes $\underline{\underline{M}}^{-1} \in \mathbb{R}^{24\times24}$, $\underline{\underline{M}}_{11}^{-1} \in \mathbb{R}^{16\times16}$, $\underline{\underline{M}}_{12}^{-1} \in \mathbb{R}^{16\times8}$, $\underline{\underline{M}}_{21}^{-1} \in \mathbb{R}^{8\times16}$, and $\underline{\underline{M}}_{22}^{-1} \in \mathbb{R}^{8\times8}$.

We are only interested in the coefficients $\underline{c}$ – the actual values of the Lagrange multipliers are irrelevant. Thus it is sufficient to consider

$$\underline{c} = \underline{\underline{M}}_{11}^{-1}\underline{\underline{X}}^T\underline{p} + \underline{\underline{M}}_{12}^{-1}\underline{u}. \tag{26}$$

Eq. (26) relates the coefficients $\underline{c}$ of the double trilinear solution ansatz to the (yet unknown) nodal values $\underline{u}$ and the prescribed jump values at the interface $\underline{p}$.

Remark 5.1. One might ask why we include the triangle barycenters as additional points for prescribing jump conditions. Even in the case of a type II cut-cell (see Fig. 5) the four triangle corner points alone provide eight jump conditions which should be (in addition to the eight corner values) sufficient to calculate the 16 unknown coefficients. However, there is a subtle problem associated with the trilinear ansatz: On a plane interface with a constant normal vector $\vec{n} = [n_x, n_y, n_z]^T$ and $n_i = n_j = 1/\sqrt{2}$, $n_k = 0$, $i,j,k \in x,y,z$ and $i \neq j$, $k \neq i,j$ (that is a plane interface at a 45° angle to one of the coordinate axes) the gradient of the trilinear ansatz reduces to a linear function with only two degrees of freedom. That means that for a cut-cell of type II or IV under such conditions the interface triangle corner points provide five independent conditions in the value jumps but only two in the gradients amounting to a total of seven independent conditions. But also in the case of an interface that is slightly disturbed from the singular case described above the problem becomes ill conditioned making the solution numerically difficult. The jump conditions at the triangle barycenters ensure that we always provide sufficiently many independent jump conditions. We further remark that we have no analytic proof that the matrix $\underline{\underline{M}}$ never gets singular for normal cells using the prescribed number of jump conditions. However, in numerous tests we did not observe any sign of a developing singularity, e.g. an increased value of the condition number.

Remark 5.2. Instead of solving the constrained minimization problem via Lagrange multipliers we could have used a direct approach. The coefficients (12) are solution values at the nodes of the cell. Therefore, eight values of $\underline{a}$ and $\underline{b}$ are directly given by the solution of u at the nodes of the grid. Instead of enforcing exact nodal values via Lagrange multipliers in (24), the known nodal values can be used to formulate the minimization problem directly in the remaining eight unknowns. However, we have chosen the Lagrange multiplier approach here as it allows (for future applications) a more flexible handling of different types of constraints and boundary conditions.

5.2.2. Singular cut-cells

The minimization approach in the presented form fails in cases of vanishing interface areas or, equivalently, vanishing partial volumes within a cell. This can only occur for cells of type I and II, see Figs. 4 and 8. For cut-cells of type I the interface can degenerate either into a point or a line, interfaces of type II cells can degenerate to a line. In case of a vanishing interface area some or all of the prescribed jump conditions fall onto the same point and the minimization matrix $\underline{\underline{M}}$ in (25) becomes singular. Even in cases of a non-vanishing but small partial volume, the condition number of the matrix $\underline{\underline{M}}$ becomes extremely large rendering the calculation of the inverse of $\underline{\underline{M}}$ numerically difficult or even impossible. Common approaches found in the literature, e.g. [11,8] and probably used in many codes is to simply limit the minimal allowed interface area within a cell, which is not satisfying as it introduces new and unnecessary sources of truncation errors.

In order to obtain a robust method without artificially limiting the minimal interface area within a cell, we follow the ideas presented in [28] and remove those singularities by a two-term asymptotic approach. For this approach we need to introduce a second coordinate system with an origin at a node in the vanishing partial volume.

Let $\mathfrak{n} \in 1,\ldots,8$ be the node in cell (i,j,k) with global coordinates $\vec{x}^{\mathfrak{n}}_{(i,j,k)} = [x^{\mathfrak{n}}_{(i,j,k)}, y^{\mathfrak{n}}_{(i,j,k)}, z^{\mathfrak{n}}_{(i,j,k)}]^T$, see Fig. 6. We introduce the following set of local orthogonal coordinates with origin at node $\mathfrak{n}$ of cell (i,j,k):

$$\vec{\xi}^{\mathfrak{n}}_{(i,j,k)} = \begin{bmatrix} \xi^{\mathfrak{n}}_{(i,j,k)} \\ \eta^{\mathfrak{n}}_{(i,j,k)} \\ \zeta^{\mathfrak{n}}_{(i,j,k)} \end{bmatrix} = \begin{bmatrix} \frac{x-x^{\mathfrak{n}}_{(i,j,k)}}{\Delta x} \\ \frac{y-y^{\mathfrak{n}}_{(i,j,k)}}{\Delta y} \\ \frac{z-z^{\mathfrak{n}}_{(i,j,k)}}{\Delta z} \end{bmatrix}, \quad \mathfrak{n} \in 1,\ldots,8. \tag{27}$$

For $\mathfrak{n} = 1$ we have the same coordinate system as used in the minimization approach for non-singular cells.

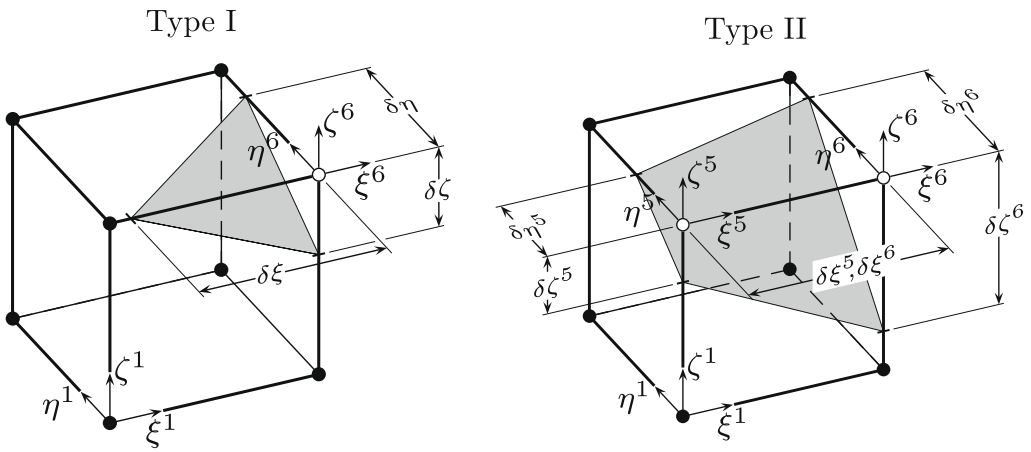

Fig. 8. Coordinate systems and naming conventions for the two-term asymptotic solution approach. The pictures show exemplary configurations for a cut-cell of type I with $\mathfrak{n} = 6$ on the left, and $\mathfrak{n}_1 = 5$, $\mathfrak{n}_2 = 6$ for a type II cut-cell on the right.

Without loss of generality we choose Ω^- to be the vanishing partial volume of cell (i, j, k). For cut-cells of type I only one node with the number $\mathfrak{n}$ is element of Ω^- ($\mathfrak{n} = 6$ in the exemplary configuration shown in Fig. 8). Cut cells of type II have two nodes $\mathfrak{n}_1$ and $\mathfrak{n}_2$ within the vanishing partial volume of the cell ($\mathfrak{n}_1 = 5$ and $\mathfrak{n}_2 = 6$ in the example of Fig. 8). Let $\hat{\xi} \in \xi, \eta, \zeta$ be the coordinate direction connecting nodes $\mathfrak{n}_1$ and $\mathfrak{n}_2$. Without loss of generality we assume that $\mathfrak{n}_1$ and $\mathfrak{n}_2$ are ordered in such a way that $\hat{\xi}_{\mathfrak{n}_1} = 0$ and $\hat{\xi}_{\mathfrak{n}_2} = 1$, and set $\mathfrak{n} = \mathfrak{n}_1$ in (27) for type II cut-cells.

Furthermore, we define $[\![u]\!]_{\mathfrak{n}_i}$ with $\mathfrak{n}_i = \mathfrak{n}$ for type I cells and $\mathfrak{n}_i = \mathfrak{n}_1, \mathfrak{n}_2$ for type II cells to be the prescribed value jump on the triangulated interface closest to the point $\mathfrak{n}_i$. This ensures that we always use the correct jump of u in the leading order solution in the case of a vanishing partial volume Ω^-, see below.

With the definitions above we introduce the following two-term ansatz for the solutions u^+ and u^- on both sides of the interface:

$$\begin{aligned} u^+(\vec{\xi}^1) &= u^{(+,0)}(\vec{\xi}^1) + \varepsilon u^{(+,1)}(\vec{\xi}^1), \\ u^-(\vec{\xi}^{\mathfrak{n}}, \tilde{\xi}) &= u^{(-,0)}(\vec{\xi}^{\mathfrak{n}}) + \varepsilon u^{(-,1)}(\tilde{\vec{\xi}}), \end{aligned} \tag{28}$$

where $u^{(+,0)}$, $u^{(-,0)}$ are leading order solutions, $u^{(+,1)}$, $u^{(-,1)}$ are corrections to the leading order solutions, and ε is a measure of the singularity of the cell to be defined below. With $\tilde{\vec{\xi}}$ we denote scaled coordinates within the vanishing partial volume Ω^- which will be introduced below. We remark that we always use coordinates $\vec{\xi}^1 = \vec{\xi}$ for the solution in Ω^+, whereas the origin for the solution ansatz in Ω^- varies with the orientation of the interface.

5.2.2.1. Leading order solution. The leading order solution is the solution for a vanishing partial volume Ω^-. This situation corresponds to an interface touching only node $\mathfrak{n}$ for cut-cells of type I or nodes $\mathfrak{n}_1$ and $\mathfrak{n}_2$ for type II cells. In both cases the whole domain of the cell is effectively part of Ω^{st}. The solution values $u^+_{\mathfrak{n}}$ for type I cells, $u^+_{\mathfrak{n}_1}$ and $u^+_{\mathfrak{n}_2}$ for type II cells are directly given by

$$u^+_{\mathfrak{n}_i} = u^-_{\mathfrak{n}_i} + [\![u]\!]_{\mathfrak{n}_i} = u_{\mathfrak{n}_i} + [\![u]\!]_{\mathfrak{n}_i} \tag{29}$$

where $u_{\mathfrak{n}_i}$ is the solution value at node $\mathfrak{n}_i$ of the cell. However, with (29) we have eight corner values for u^+ on hand and we can set the leading order solution in Ω^+ to be a full trilinear function:

$$u^{(+,0)} = \underline{x} \cdot \underline{a}^{(0)} \tag{30}$$

with

$$a^{(0)}_i = u^+_i = \begin{cases} u_i + [\![u]\!]_{\mathfrak{n}_i}, & \text{if } i = \mathfrak{n}_i \\ u_i, & \text{else} \end{cases}, \quad i = 1, \ldots, 8,$$

and $\underline{x}$ defined in (9). This can be written in a slightly different form:

$$\underline{a}^{(0)} = \underline{u} + \underline{\underline{D}} \cdot \underline{[\![u]\!]} \tag{31}$$

with

$$\underline{\underline{D}} = \mathrm{diag}(d_1, \ldots, d_N), \quad \text{and} \quad d_j = \delta_{\mathfrak{n}_i j}, \quad \mathfrak{n}_i = \mathfrak{n}_1, \mathfrak{n}_2.$$

For type I cells the spatial distribution of the leading order solution in Ω^- has no importance and is set to be a constant, i.e.

$$u_I^{(-,0)} = u_{\mathfrak{n}_1}. \tag{32}$$

For type II cells we set the leading order solution to be a linear function between the two nodes $\mathfrak{n}_1$, $\mathfrak{n}_2 \in \Omega^-$, which can be written as

$$u_{II}^{(-,0)} = (1-\hat{\xi})\, u_{\mathfrak{n}_1} + \hat{\xi}\, u_{\mathfrak{n}_2}. \tag{33}$$

Here, $\hat{\xi} \in \xi, \eta, \zeta$ is the coordinate pointing from node $\mathfrak{n}_1$ to $\mathfrak{n}_2$ ($\hat{\xi} = \xi$ in Fig. 8). For later purpose we write the leading order solution in Ω^- in the following form:

$$u^{(-,0)} = \underline{x}^- \cdot \underline{b}^{(0)} \tag{34}$$

with

$$\underline{b}^{(0)} = \underline{u}$$

and

$$\underline{x}^- = [x_1^-, \ldots, x_8^-]^T, \tag{35}$$

with

$$x_i^- = \delta_{i\mathfrak{n}}, \quad i \in 1, \ldots, 8 \tag{36}$$

for type I cut-cells and

$$x_i^- = \begin{cases} (1-\hat{\xi}), & \text{if} \quad i = \mathfrak{n}_1, \\ \hat{\xi}, & \text{if} \quad i = \mathfrak{n}_2, \\ 0, & \text{else}, \end{cases} \qquad i \in 1, \ldots, 8 \tag{37}$$

for type II cells.

For the construction of the correction solution in the next section we summarise the coefficients of the leading order solution as:

$$\underline{c}^{(0)} = [\underline{a}^{(0)}\ \underline{b}^{(0)}]^T = \underline{\underline{U}}\,\underline{u} + \underline{\underline{P}}\,\underline{p}, \tag{38}$$

with

$$\underline{\underline{U}} = [\underline{\underline{I}} \quad \underline{\underline{I}}]^T, \quad \underline{\underline{I}} \in \mathbb{R}^{8\times 8},$$
$$\underline{p} = [[\![u]\!], [\![\beta u_n]\!]]^T$$

and

$$\underline{\underline{P}} = \begin{bmatrix} \underline{\underline{D}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} \end{bmatrix}.$$

5.2.2.2. Correction solution. For partial volumes $|\Omega^-| > 0$ we need to correct the leading order solution. For the correction solution we make a full trilinear ansatz on both sides of the interface:

$$\begin{aligned} u^{(+,1)} &= \underline{x} \cdot \underline{a}^{(1)}, \\ u^{(-,1)} &= \tilde{\underline{x}} \cdot \underline{b}^{(1)}, \end{aligned} \tag{39}$$

with $\underline{x}$ defined in (11) and $\tilde{\underline{x}}$ similarly defined with ξ, η, ζ replaced with the scaled coordinates $\tilde{\xi}$, $\tilde{\eta}$, $\tilde{\zeta}$. The scaled coordinates for the correction solution $u^{(-,1)}$ in Ω^- are given as

$$\vec{\tilde{\xi}} = \begin{bmatrix} \tilde{\xi} \\ \tilde{\eta} \\ \tilde{\zeta} \end{bmatrix} = \begin{bmatrix} \xi^{\mathfrak{n}}/\delta\xi \\ \eta^{\mathfrak{n}}/\delta\eta \\ \zeta^{\mathfrak{n}}/\delta\zeta \end{bmatrix}. \tag{40}$$

The scaling factors $\delta\xi$, $\delta\eta$, $\delta\zeta$ are defined as the distances from nodes $\mathfrak{n}_1$ or $\mathfrak{n}_2$ to the interface along the edges of the cell in direction of $\xi^{\mathfrak{n}}$, $\eta^{\mathfrak{n}}$, $\zeta^{\mathfrak{n}}$ axes, respectively, see Fig. 8. For cut-cells of type I we define

$$\delta\xi = \delta\xi^{\mathfrak{n}}, \quad \delta\eta = \delta\eta^{\mathfrak{n}}, \quad \delta\zeta = \delta\zeta^{\mathfrak{n}}, \tag{41}$$

and for type II cells we define

$$\begin{aligned} \delta\xi &= \max(\delta\xi^{\mathfrak{n}_1}, \delta\xi^{\mathfrak{n}_2}), \\ \delta\eta &= \max(\delta\eta^{\mathfrak{n}_1}, \delta\eta^{\mathfrak{n}_2}), \\ \delta\zeta &= \max(\delta\zeta^{\mathfrak{n}_1}, \delta\zeta^{\mathfrak{n}_2}) \end{aligned} \tag{42}$$

and remark that the scaling factor in the connecting direction between the nodes $\mathfrak{n}_1$ and $\mathfrak{n}_2$ is defined to be unity. With this scaling we always achieve $O(\tilde{\xi}) = O(\tilde{\eta}) = O(\tilde{\zeta}) = 1$ within an Ω^- and avoid situations of $\tilde{\xi}, \tilde{\eta}, \tilde{\zeta} \to \infty$ for cut-cells of type II. In addition we define the small parameter ε as

$$\varepsilon = \min(\delta\xi, \delta\eta, \delta\zeta). \tag{43}$$

The introduction of scaled coordinates ensures that the distance between the different points of prescribed jump conditions in $\tilde{\xi}, \tilde{\eta}, \tilde{\zeta}$ space is independent of ε and always of order $O(1)$. This guarantees that the prescribed jump conditions always lead to a set of linearly independent equations for the minimization.

The gradients in direction of $\vec{n}$ follow from (28) as

$$\begin{aligned}(\nabla u^+ \cdot \vec{n})(\vec{\tilde{\xi}}) &= \underline{y} \cdot \underline{a}^{(0)} + \varepsilon\,(\underline{y} \cdot \underline{a}^{(1)}),\\ (\nabla u^- \cdot \vec{n})(\vec{\tilde{\xi}}^{\mathfrak{n}}, \vec{\tilde{\xi}}) &= \underline{y}^- \cdot \underline{b}^{(0)} + \varepsilon\,(\underline{\tilde{y}} \cdot \underline{b}^{(1)}),\end{aligned} \tag{44}$$

where $\underline{y}$ is defined in (14). The leading order solution for type I cells is a constant leading to

$$\underline{y}^- = \underline{0}$$

for such cells. The leading order gradient in Ω^- for type II cells follows from (33) as

$$\underline{y}^- \cdot \underline{b}^{(0)} = (-\hat{\xi} u_{\mathfrak{n}_1} + \hat{\xi} u_{\mathfrak{n}_2})\, n_{\hat{\xi}}$$

with $n_{\hat{\xi}} \in n_x, n_y, n_z$ corresponding to alignment of $\hat{\xi}$.

For the solution gradient in normal direction we have in scaled coordinates

$$\underline{\tilde{y}} = \frac{\partial \underline{\tilde{x}}}{\partial x} n_x + \frac{\partial \underline{\tilde{x}}}{\partial y} n_y + \frac{\partial \underline{\tilde{x}}}{\partial z} n_z, \tag{45}$$

where

$$\frac{\partial \underline{\tilde{x}}}{\partial x} = \frac{1}{\Delta x\, \delta\xi} \frac{\partial \underline{\tilde{x}}}{\partial \tilde{\xi}}, \quad \frac{\partial \underline{\tilde{x}}}{\partial y} = \frac{1}{\Delta y\, \delta\eta} \frac{\partial \underline{\tilde{x}}}{\partial \tilde{\eta}}, \quad \text{and} \quad \frac{\partial \underline{\tilde{x}}}{\partial z} = \frac{1}{\Delta z\, \delta\zeta} \frac{\partial \underline{\tilde{x}}}{\partial \tilde{\zeta}}.$$

As an example, for the gradient in x-direction we have

$$\frac{\partial \underline{\tilde{x}}}{\partial x} = \frac{1}{\Delta x\, \delta\xi} \begin{bmatrix} -(1-\tilde{\eta}) & (1-\tilde{\zeta}) \\ -(1-\tilde{\eta}) & (1-\tilde{\zeta}) \\ -\tilde{\eta} & (1-\tilde{\zeta}) \\ -\tilde{\eta} & (1-\tilde{\zeta}) \\ -(1-\tilde{\eta}) & \tilde{\zeta} \\ -(1-\tilde{\eta}) & \tilde{\zeta} \\ -\tilde{\eta} & \tilde{\zeta} \\ -\tilde{\eta} & \tilde{\zeta} \end{bmatrix}^T.$$

Note that $\tilde{y} \to \infty$ for $\varepsilon \to 0$. However, this singularity is canceled by the ε term in the asymptotic expansion (44).

Given the two-term ansatz (28) we can write the jump residual in u and βu_n as

$$\tilde{r}_{1,k} = \widehat{[\![u]\!]}_k - [\![u]\!]_k = \widehat{[\![u]\!]}_k^{(0)} + \varepsilon \widehat{[\![u]\!]}_k^{(1)} - [\![u]\!]_k = ((\underline{x}_k \cdot \underline{a}^{(0)}) - (\underline{x}_k^- \cdot \underline{b}^{(0)})) + \varepsilon((\underline{x}_k \cdot \underline{a}^{(1)}) - (\underline{\tilde{x}}_k \cdot \underline{b}^{(1)})) - [\![u]\!]_k, \tag{46}$$

$$\begin{aligned}\tilde{r}_{2,k} &= \widehat{[\![\beta u_n]\!]}_k - [\![\beta u_n]\!]_k = \widehat{[\![\beta u_n]\!]}_k^{(0)} + \varepsilon \widehat{[\![\beta u_n]\!]}_k^{(1)} - [\![\beta u_n]\!]_k \\ &= (\beta^+(\underline{y}_k \cdot \underline{a}^{(0)}) - \beta^-(\underline{y}_k^- \cdot \underline{b}^{(0)})) + \varepsilon(\beta^+(\underline{y}_k \cdot \underline{a}^{(1)}) - \beta^-(\underline{\tilde{y}} \cdot \underline{b}^{(1)})) - [\![\beta u_n]\!]_k.\end{aligned} \tag{47}$$

To give each individual component of the residuals a comparable influence on the minimization solution we introduce, as before, the scaled residuals

$$r_{1,k} = w_{1,k}\, \tilde{r}_{1,k} \quad \text{and} \quad r_{2,k} = w_{2,k}\, \tilde{r}_{2,k} \tag{48}$$

with weights

$$w_{1,k} = \|\varepsilon\,[\underline{x}_k,\ \underline{x}_k]\|_2^{-1} \quad \text{and} \quad w_{2,k} = \|\varepsilon\,[\beta^+ \underline{y}_k,\ \beta^- \underline{\tilde{y}}_k]\|_2^{-1}. \tag{49}$$

The scaled jump residuals (48) can be written in compact matrix notation as

$$\underline{r} = \underline{\underline{\tilde{X}}}\, \underline{c}^{(1)} + \underline{\underline{X}}\, \underline{c}^{(0)} - \underline{p} \tag{50}$$

with

$$\underline{\underline{X}} = \begin{bmatrix} \underline{\underline{X}}_\Gamma & -\underline{\underline{X}}_\Gamma^- \\ \beta^+ \underline{\underline{Y}}_\Gamma & -\beta^- \underline{\underline{Y}}_\Gamma^- \end{bmatrix}, \quad \underline{\underline{\tilde{X}}} = \begin{bmatrix} \underline{\underline{X}}_\Gamma & -\underline{\underline{\tilde{X}}}_\Gamma \\ \beta^+ \underline{\underline{Y}}_\Gamma & -\beta^- \underline{\underline{\tilde{Y}}}_\Gamma \end{bmatrix}, \quad \underline{c}^{(0)} = \begin{bmatrix} \underline{a}^{(0)} \\ \underline{b}^{(0)} \end{bmatrix}, \quad \underline{c}^{(1)} = \begin{bmatrix} \underline{a}^{(1)} \\ \underline{b}^{(1)} \end{bmatrix}, \quad \underline{p} = \begin{bmatrix} \underline{v} \\ \underline{g} \end{bmatrix}, \quad \underline{v} = \begin{bmatrix} w_{1,1}[\![u]\!]_1 \\ \vdots \\ w_{1,N}[\![u]\!]_N \end{bmatrix},$$

$$\underline{g} = \begin{bmatrix} w_{2,1}[\![\beta u_n]\!]_1 \\ \vdots \\ w_{2,N}[\![\beta u_n]\!]_N \end{bmatrix}$$

and

$$\underline{\underline{X}}_\Gamma = \begin{bmatrix} w_{1,1}\underline{x}_1 \\ \vdots \\ w_{1,N}\underline{x}_N \end{bmatrix}, \quad \underline{\underline{Y}}_\Gamma = \begin{bmatrix} w_{2,1}\underline{y}_1 \\ \vdots \\ w_{2,N}\underline{y}_N \end{bmatrix}, \quad \underline{\underline{X}}_\Gamma^- \begin{bmatrix} w_{1,1}\underline{x}_1^- \\ \vdots \\ w_{1,N}\underline{x}_N^- \end{bmatrix}, \quad \underline{\underline{Y}}_\Gamma^- = \begin{bmatrix} w_{2,1}\underline{y}^- \\ \vdots \\ w_{2,N}\underline{y}^- \end{bmatrix}, \quad \underline{\underline{\tilde{X}}}_\Gamma = \begin{bmatrix} w_{1,1}\tilde{\underline{x}}_1 \\ \vdots \\ w_{1,N}\tilde{\underline{x}}_N \end{bmatrix}, \quad \underline{\underline{\tilde{Y}}}_\Gamma = \begin{bmatrix} w_{2,1}\tilde{\underline{y}}_1 \\ \vdots \\ w_{2,N}\tilde{\underline{y}}_N \end{bmatrix}.$$

The unknowns in (50) are the coefficients $\underline{c}^{(1)} = [\underline{a}^{(1)}, \underline{b}^{(1)}]^T$, whereas $\underline{c}^{(0)}$ is already known from the leading order solution. In accordance with the procedure for normal cells in Section 5.2.1 we want to minimize the residual $\underline{r}$ under the constraint of exact nodal values. Since the leading order solution fulfills already the condition of exact nodal values, the correction solution has to be zero at the nodes of the cell, i.e.

$$\underline{\underline{B}}\underline{c}^{(1)} = \underline{0}, \tag{51}$$

where $\underline{\underline{B}}$ is given in (21).

The problem of minimizing the residual vector $\underline{r}$ under constraint (51) can now be stated as

$$L(\underline{c}^{(1)}, \underline{\lambda}) = \frac{1}{2}(\underline{r} \cdot \underline{r}) + \underline{\lambda} \cdot (\underline{\underline{B}}\underline{c}^{(1)}) = \min. \tag{52}$$

The necessary conditions for a minimum of $L(\underline{c}^{(1)}, \underline{\lambda})$ are

$$\frac{\partial L}{\partial \underline{c}^{(1)}} = 0 \quad \text{and} \quad \frac{\partial L}{\partial \underline{\lambda}} = 0$$

leading to the following set of linear equations:

$$\underbrace{\begin{bmatrix} \underline{\underline{\tilde{X}}}^T \underline{\underline{\tilde{X}}} & \underline{\underline{B}}^T \\ \underline{\underline{B}} & \underline{\underline{0}} \end{bmatrix}}_{\underline{\underline{\tilde{M}}}} \begin{bmatrix} \underline{c}^{(1)} \\ \underline{\lambda} \end{bmatrix} = -\begin{bmatrix} \underline{\underline{\tilde{X}}}^T \underline{\underline{X}} \underline{c}^{(0)} \\ \underline{0} \end{bmatrix} + \begin{bmatrix} \underline{\underline{\tilde{X}}}^T \underline{p} \\ \underline{0} \end{bmatrix}. \tag{53}$$

Using (38) we can write for the second order solution

$$\underline{c}^{(1)} = \left\{ -\underline{\underline{\tilde{M}}}_{11}^{-1} \underline{\underline{\tilde{X}}}^T \underline{\underline{X}} \underline{c}^{(0)} + \underline{\underline{\tilde{M}}}_{12}^{-1} \underline{\underline{\tilde{X}}}^T \underline{p} \right\} = \{ -\underline{\underline{\tilde{M}}}_{11}^{-1} \underline{\underline{\tilde{X}}}^T \underline{\underline{X}} \underline{\underline{U}} \underline{u} - (\underline{\underline{\tilde{M}}}_{11}^{-1} \underline{\underline{\tilde{X}}}^T \underline{\underline{X}} \underline{\underline{P}} - \underline{\underline{\tilde{M}}}_{12}^{-1} \underline{\underline{\tilde{X}}}^T) \underline{p} \}, \tag{54}$$

where $\underline{\underline{\tilde{M}}}_{11}^{-1}$ and $\underline{\underline{\tilde{M}}}_{12}^{-1}$, are sub-matrices of $\underline{\underline{\tilde{M}}}^{-1}$ in (53):

$$\underline{\underline{\tilde{M}}}^{-1} = \begin{bmatrix} \underline{\underline{\tilde{M}}}_{11}^{-1} & \underline{\underline{\tilde{M}}}_{12}^{-1} \\ \underline{\underline{\tilde{M}}}_{21}^{-1} & \underline{\underline{\tilde{M}}}_{22}^{-1} \end{bmatrix},$$

with $\underline{\underline{\tilde{M}}}^{-1} \in \mathbb{R}^{24\times 24}$, $\underline{\underline{\tilde{M}}}_{11}^{-1} \in \mathbb{R}^{16\times 16}$, $\underline{\underline{\tilde{M}}}_{12}^{-1} \in \mathbb{R}^{16\times 8}$, $\underline{\underline{\tilde{M}}}_{21}^{-1} \in \mathbb{R}^{8\times 16}$, and $\underline{\underline{\tilde{M}}}_{22}^{-1} \in \mathbb{R}^{8\times 8}$.

Fig. 9, left, demonstrates the effectivity of our two-term asymptotic approach. The figure compares the condition number of the minimization matrix $\underline{\underline{M}}$ using a single step minimization as for non-singular cut-cells and the two-term asymptotic approach using the scaling factors introduced in Eq. (49). For the figure we have used a singular cut-cell of type I with $\varepsilon = \delta\xi = \delta\eta = \delta\zeta$ and different ratios of the coefficients $\beta^+ : \beta^-$. The condition number for the single step minimization matrix according to (25) quickly becomes extremely large making the computation of the inverse prone to errors or even impossible. In contrast, down to the machine accuracy the condition number for the two-term asymptotic approach in (53) is essentially independent of the small parameter ε and, due to an appropriate scaling, for small ε also independent of the ratio of coefficients $\beta^+ : \beta^-$. Fig. 9, right, demonstrates the influence of the ratio of coefficients using scaling factors $w_{1,k} = \varepsilon^{-1}$ and $w_{2,k} = 1$ instead of (49). This scaling preserves the influence of the ratio of coefficients and effectively removes ε in front of the correction solution in (46) and keeps ε in front of the gradient residual (47) which, in turns, cancels the singularity in $\underline{y}$. Eq. (53) defines a constraint linear least square problem and we expect a condition number which scales with the square of the condition number of the original set of equations which is basically the ratio of coefficients $\beta^+ : \beta^-$. Fig. 9, right, clearly demonstrates this scaling property.

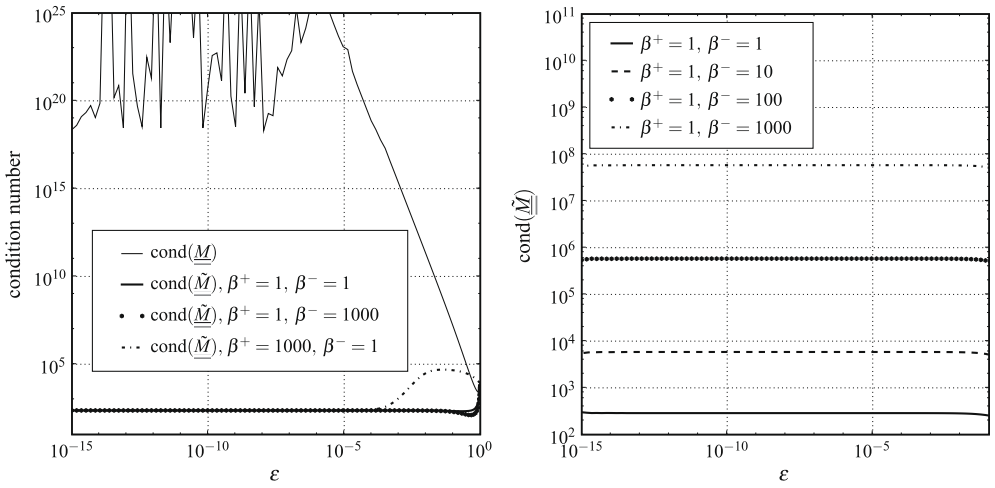

Fig. 9. Condition number of the minimization matrices $\underline{\underline{M}}$ for the single step solution and $\tilde{\underline{\underline{M}}}$ for the two step asymptotic solution according to (25) and (53), respectively, for a singular cut-cell of type I with $\varepsilon = \delta\tilde{\xi} = \delta\eta = \delta\zeta$. In the plot on the left side we use the scaling factors defined in (49) for the two-step asymptotic approach whereas the plot on the right side uses scaling factors $w_{1,k} = \varepsilon^{-1}$ and $w_{2,k} = 1$.

All results presented in this paper have been obtained using the two-step asymptotic approach for singular cut-cells and the scaling factors given in (49).

5.2.2.3. Composite solution. With the results from the preceding sections we can now summarise the complete solution (28) as

$$\begin{aligned} u^+(\vec{\tilde{\xi}}) &= \underline{x} \cdot \underline{a}^{(0)} + \varepsilon(\underline{x} \cdot \underline{a}^{(1)}), \\ u^-(\vec{\tilde{\xi}}^{\mathfrak{n}}, \tilde{\tilde{\xi}}) &= \underline{x}^{\mathfrak{n}} \cdot \underline{b}^{(0)} + \varepsilon(\tilde{\underline{x}} \cdot \underline{b}^{(1)}), \end{aligned} \tag{55}$$

where $\underline{x}$, $\underline{x}^{\mathfrak{n}}$, $\underline{a}^{(0)}$, $\underline{a}^{(1)}$, $\underline{b}^{(0)}$, and $\underline{b}^{(1)}$ are given by (9), (35), (37), (38) and (54), respectively. The shape function vector $\tilde{\underline{x}}$ is formally identical to (9) with ξ, η, ζ replaced with the scaled coordinates $\tilde{\xi}$, $\tilde{\eta}$, $\tilde{\zeta}$.

5.3. Evaluation of the discrete Laplacian

With (8)–(10) for normal cells, (11) with (26) for non-singular cut-cells, and (55) for singular cut-cells we can express the piecewise trilinear distribution of u on each cell of the grid as function of discrete node values and jump conditions. This allows us to evaluate the surface integrals on the left hand side of (6) analytically which leads to a sparse system of linear equations for the solution values $u_{i,j,k}$ on the nodes of the grid.

If the surface element S_i^c in (6) has an intersection with the interface we split the integral in two parts,

$$\int_{S_i^c} \beta \nabla u \cdot \vec{n}\, dS = \int_{S_i^{c,+}} \beta^+ \nabla u^+ \cdot \vec{n}\, dS + \int_{S_i^{c,-}} \beta^- \nabla u^{\mathfrak{B}} \cdot \vec{n}\, dS, \tag{56}$$

and evaluate each part of the integral analytically using (11) and (26) or (55) for the solutions on the different sides of the interface.

For the evaluation of the integrals on the right hand side of (56) we assume the intersection between the interface Γ and S_i^c to be a straight line, i.e. $S_i^{c,+}$ and $S_i^{c,+}$ can always be represented by triangles and/or rectangles, see Fig. 10 for an illustration.

The dependence of the piecewise trilinear solution on the jump conditions at the interface leads – in addition to the terms on the right hand side of (6) – to a contribution of surface integrals to the right hand side of the global linear system of equations. However, for normal cells as well as for cut cells we always get a compact 27-point stencil.

For $\beta^+ = \beta^-$ we end up with a symmetric and diagonally dominant matrix. However, in the general case of $\beta^+ \neq \beta^-$ the resulting matrix is non-symmetric. That also means that we cannot a priori guarantee that our methods fulfills a discrete maximum principle.

5.4. Evaluation of source terms

For a second order approximation of (5) we use

$$\int_\Omega f\, dV \approx |\Omega^+| f^+(\boldsymbol{x}_s^+) + |\Omega^-| f^-(\boldsymbol{x}_s^-), \tag{57}$$

where $\boldsymbol{x}_s^\pm$ denotes the barycenter of $\Omega^\pm$.

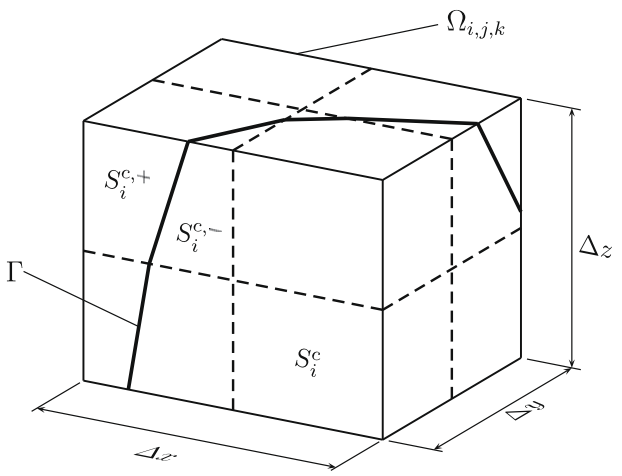

Fig. 10. Piecewise linear representation of the intersections between the boundary of control volume $\Omega_{i,j,k}$ and the interface Γ. The parts of the surface bounded by the dashed lines are the inner faces of cells around node (i,j,k) as shows in Fig. 3.

For the integral of the gradient jump over the interface Γ we use a triangulation of the interface as described in Section 4 based on sub-cells. Let $\overline{\mathcal{C}}_{i,j,k}$ denote the set of sub-cells composing control volume $\Omega_{i,j,k}$. If $\mathcal{T}_c$, $c \in \overline{\mathcal{C}}_{i,j,k}$ is the set of interface triangles in sub-cell c we can write for the second integral on the right hand side of (5)

$$\int_{\Gamma_{\Omega_{i,j,k}}} [\![\beta u_n]\!]\, dS = \sum_{c\in\overline{\mathcal{C}}_{\cdot,\cdot,\cdot}} \sum_{\tau\in\mathcal{T}_c} \int_\tau [\![\beta u_n]\!]\, dS \approx \sum_{c\in\overline{\mathcal{C}}_{\cdot,\cdot,\cdot}} \sum_{\tau\in\mathcal{T}_c} |\tau|\, [\![\beta u_n]\!]_\tau, \tag{58}$$

where $[\![\beta u_n]\!]_\tau$ is the discrete value of the gradient jump in the barycenter of triangle τ. Eq. (58) is a standard second order approximation of an integral over a triangulated surface.

6. Results

In the following examples we compare numerical results with given analytic solutions $u^+(\vec{x})$ and $u^-(\vec{x})$ and prescribed coefficients $\beta^+(\vec{x})$ and $\beta^-(\vec{x})$. We use the analytic solutions to provide values for $f^\pm(\vec{x})$ in the barycenter(s) of Ω^+ and/or Ω^+ of (57) and to evaluate discrete values of the jump conditions $[\![u]\!]$ and $[\![\beta u_n]\!]$. Furthermore, the analytic solutions provide Dirichlet boundary conditions on $\partial\overline{\Omega}$.

All geometric information about the interface is evaluated to second order of accuracy using standard levelset techniques as described in textbooks [35,30]. The interface is defined by the zero level of the signed normal distance function $\phi(\vec{x})$. We set Ω^+ and Ω^- to be the region with $\phi(\vec{x}) > 0$ and $\phi(\vec{x}) < 0$, respectively. Intersections of the interface with the edges of cells and sub-cells have been evaluated assuming linear distributions of ϕ between grid points. The unit normal vector pointing from Ω^+ to Ω^- is given by $\vec{n} = -\frac{\nabla\phi}{|\nabla\phi|}$. The normal vectors needed at various locations of the triangulated interface, see Fig. 5, are evaluated by trilinear interpolation of second order node centered normal vectors.

The arising linear systems of equations have been solved with the `hypre` library [6] using a BiCGSTAB solver preconditioned with an algebraic multigrid solver using the standard parameters for the multigrid solver as described in the manuals of the `hypre` library.

We evaluate the performance of our method by convergence studies in the usual L_∞ and L_2 norms. Furthermore, we provide convergence results for an interface norm L_B, which is the L_2-norm of all grid points of intersected cells. The order of convergence for all examples below has been evaluated by a linear least square fit of the error results on successively refined grid from $10\times10\times10$ to $120\times120\times120$ grid points. Usually we omit the results on the coarsest grid levels for the linear least square fit as the results on very coarse grids are usually not suitable for convergence studies. Furthermore, in all examples with a geometric symmetry of the interface we have shifted the interface by a small value $\varepsilon_{x|y|z} < \Delta x, \Delta y, \Delta z$ in all coordinate directions to avoid the (positive) influence of symmetries on the solutions.

6.1. Example 1

The first example is the 3D analogue of a 2D test case used in [11,28]. We solve (1) in the domain $-1 \leqslant x, y, z \leqslant 1$. The interface is a simple sphere with radius 0.5 and midpoint at $(0,0,0)$. The analytic solutions $u^\pm$ and the coefficients $\beta^\pm$ are given as:

$$u^+ = \ln(x^2+y^2+z^2), \quad u^- = \sin(x+y+z),$$
$$\beta^+ = \sin(x+y+z)+2, \quad \beta^- = \cos(x+y+z)+2.$$

The solution as well as the normal derivative are discontinuous across the interface. This example is characterized by a simple geometry of the interface with constant curvature, nonlinear solutions on both sides of the interface, and a relatively small ratio of the coefficients β^+ and β^-.

Fig. 11 shows on the left side the solution at $z = 0$. The figure demonstrates the sharp and discontinuous representation of the solution without any peaks. The convergence results for this example are summarised on the right hand side of Fig. 11.

For this test case the method shows second order of accuracy in the L_2, L_∞, and the boundary norm L_B. Not surprising, the error in the L_∞-norm is the largest and about one order of magnitude larger than the error in the L_B-norm, but the order of convergence is ≈ 2.0 in allnorms.

6.2. Example 2

This case follows an example investigated by Li in [18,28] in 2D. We use again a spherical interface radius 0.5 and midpoint at $(0,0,0)$. The analytic solutions on the computational domain $0 \leqslant x,y,z \leqslant 1$ are given as

$$u^+ = \frac{r^4 + C_0 \log(2r)}{\beta^+}, \quad u^- = \frac{r^2}{\beta^-},$$
$$\beta^+ = \text{const.}, \qquad \beta^- = \text{const.},$$

with $r = \sqrt{x^2+y^2+z^2}$ and $C_0 = -0.1$. The solution and the gradient are discontinuous across the interface, however, the jump of the gradient in normal direction $[\![\beta u_n]\!]$ is independent of the β ratio here. We use this test case to evaluate the performance of our scheme under various ratios of the coefficients $\beta^+ : \beta^-$. For large values of $\beta^\pm$ the solution becomes nearly constant in $\Omega^\pm$.

Fig. 12 shows convergence results for different ratios of the coefficients $\beta^+ : \beta^-$. For all combinations of $\beta^+ : \beta^-$ we see second order (or above for $\beta^+ : \beta^- = 1000 : 1$) of accuracy in the L_∞-, L_B-, and L_2-norms. It can further be observed from Fig. 12 that the absolute value of the error at a certain grid resolution seems to depend more on the values of β v itself than on their ratio. For the β-ratios investigated here we see an increasing difference between the errors in the L_∞- and L_B-norm with increasing β-ratio. Furthermore, at large β ratios we see occasionally a non-smooth convergence in the L_∞ norm.

Fig. 13 shows CPU times for solving the set of linear equations as a function of grid size and β-ratio. All results have been obtained on an Intel 8400 Core 2 Duo processor running at 3 GHz clock speed. The BiCGSTAB solver of the `hypre` package preconditioned with the algebraic multigrid solver BOOMER has been used here. The convergence tolerance for the solver has been set to $1e{-}13$. The scaling factor s in Fig. 13 has been obtained by a linear least square fit. The results show a slightly less than linear scaling of the CPU time over the number of grid points for all β-ratios. It is a known fact that the condition number of the resulting set of linear equations strongly depends on the ratio of the coefficient β and that problems of the type considered here become notoriously difficult to solve numerically for limitingly large ratios of the coefficients. In our examples we made the experience that the algebraic multigrid solver BOOMER used as a direct solver works very reliably and fast for low ratios of the coefficient β. However, for high β-ratios it was necessary to use a preconditioned solver as mentioned above.

The CPU times in Fig. 13 for the high β-ratio cases $\beta^+ : \beta^- = 1 : 1000$ and $\beta^+ : \beta^- = 1000 : 1$ are roughly 2–3 times longer than for $\beta^+ : \beta^- = 1 : 1$ on the same grid. The CPU times shown in Fig. 13 are representative for all other results presented in this paper. For brevity we omit the detailed CPU times for most other examples.

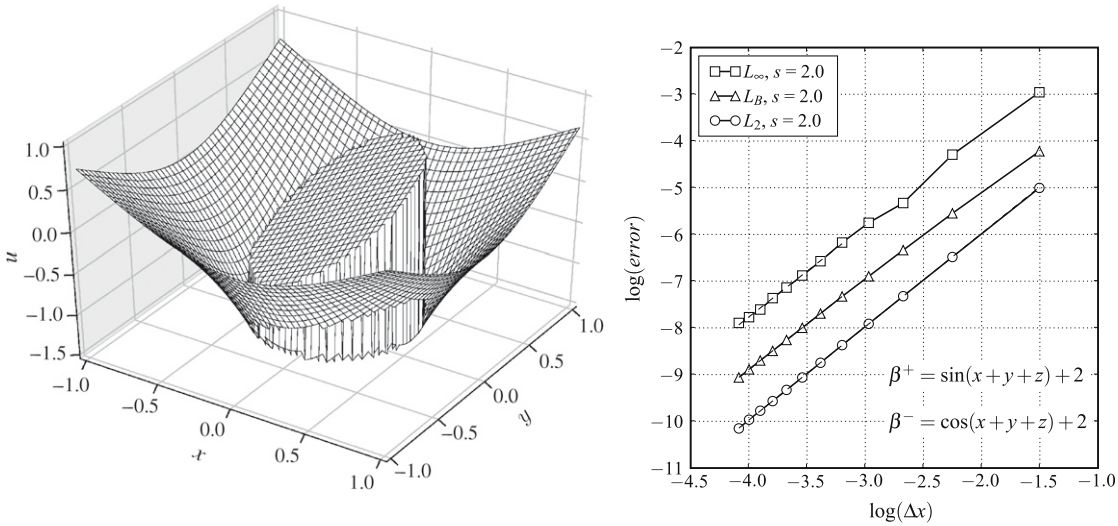

Fig. 11. Solution for example 1 at $z = 0$ (left) and convergence results (right). The slope s of the linear least square fit is the order of convergence.

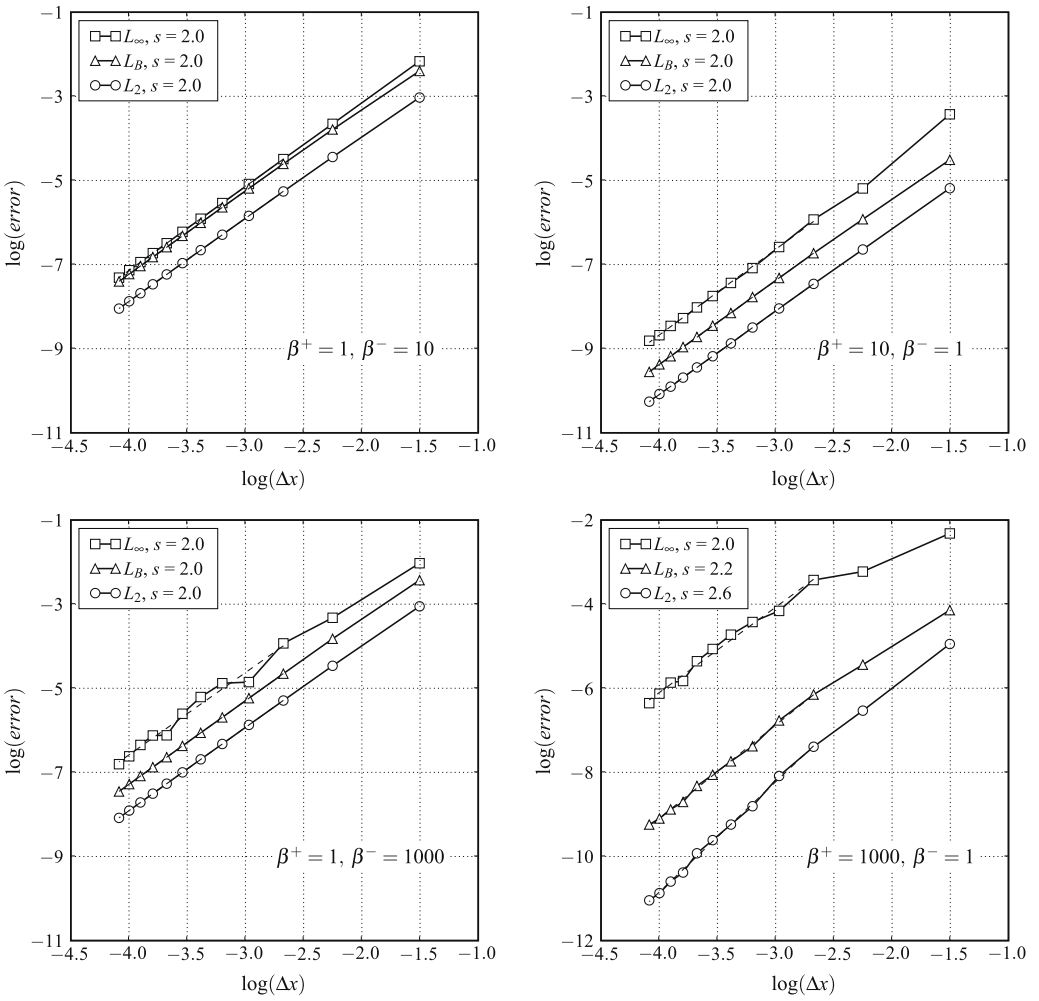

Fig. 12. Convergence results for example 2 with a spherical interface and different ratios of the coefficients β^+:β^-. The slope s of the linear least square fit is the order of convergence.

6.3. Example 3

In this example we consider the case of a complex interface, varying and high β-ratios, and nonlinear solutions on both sides of the interface. The interface is given in parametric form as

$$r(\phi,\theta) = R + \Delta R_\phi \cos^3(\theta)\cos(\omega_\phi \phi) + \Delta R_\theta \cos(\omega_\theta \theta), \tag{59}$$

with $0 \leqslant \phi \leqslant 2\pi, 0 \leqslant \theta \leqslant \pi, R = 0.65, \Delta R_\phi = \Delta R_\theta = 0.15, \omega_\phi = 6$, and $\omega_\theta = 4$, see Fig. 14 for an illustration of the interface. We use the same nonlinear solutions u^+ and u^- as in example 1 with the following different combinations of β values:

$$\text{I}: \beta^+ = \sin(x+y+z)+2, \beta^- = \cos(x+y+z)+2,$$
$$\text{II}: \beta^+ = \sin(x+y+z)+2, \beta^- = 250\cos(x+y+z)+500,$$
$$\text{III}: \beta^+ = 250\cos(x+y+z)+500, \beta^- = \sin(x+y+z)+2.$$

Cases II and III lead to a maximum ratio of the coefficients $\beta^+ : \beta^- = 1 : 750$ and $\beta^+ : \beta^- = 750 : 1$, respectively.

Fig. 15 summarises the convergence results for the different test cases. For all combinations of $\beta^+ : \beta^-$ values we get second order of accuracy in the L_∞, L_B, and L_2 norm. As in the examples before, the difference between the error in the L_∞ and L_B-norm increases with increasing ratio of the coefficients β^+ and β^-. From the results in Fig. 15 and the results of the preceding examples we conclude that the convergence ratio seems not to decrease with (a) increasing β-ratio and (b) complexity of the interface. However, the absolute value of the error is strongly affected by the β-ratio.

CPU times for the solution of the linear system of equations are shown in Fig. 16 for the different cases. We see again an almost linear scaling of the CPU time with the number of grid points. The numbers are in the same order of magnitude as in Example 2 indicating a weak influence of the interface complexity on solution times of the linear equation system solver.

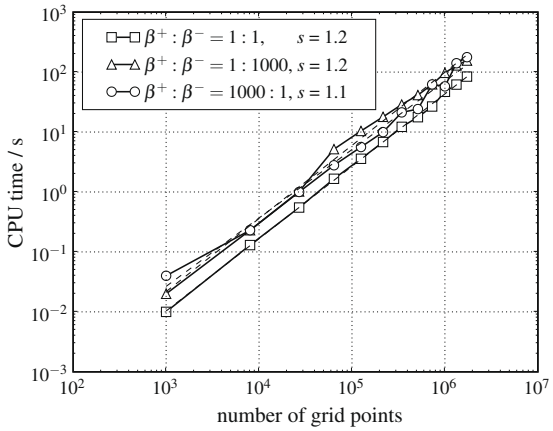

Fig. 13. CPU times for solving the resulting set of linear equations of example 2 as a function of the total number of grid points and the ratio of the coefficient β^+:β^-. The scaling factors s has been evaluated using a linear least square fit.

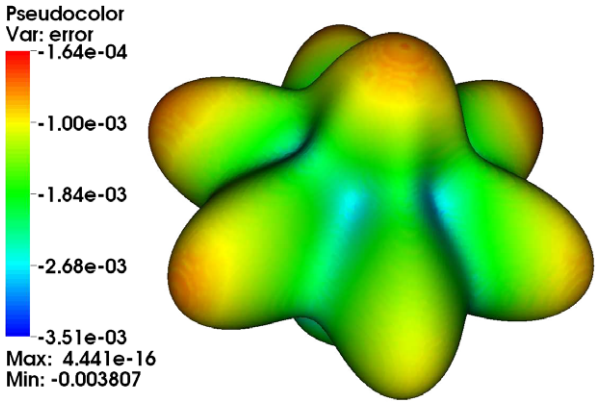

Fig. 14. Surface error plot of the star like interface of example 3.

6.4. Example 4 – oak acorn

This example has been taken from [42]. The solutions and coefficients on both sides of the interface are given as

$$\begin{aligned} &u^+ = 10(x+y+z)+1, \quad u^- = 10\cos(kx)\cos(ky)\cos(kz), \\ &\beta^+ = 80, \quad \beta^- = 1, \end{aligned} \tag{60}$$

with $k = 3$. The interface is an oak acorn defined by

$$\begin{aligned} &\left(\tfrac{x}{d}\right)^2 + \left(\tfrac{y}{d}\right)^2 = (z-q)^2, \quad \text{if } z > 0, \\ &x^2 + y^2 + (z-g)^2 = R^2, \quad \text{if } z \leqslant 0, \end{aligned}$$

with $q = -6/7$, $g = 1/2$, $R = 15/7$, and $d = \sqrt{(R^2 - g^2)/q^2}$. The interface is depicted as a plot of the error at the interface in Fig. 17. The computational domain is set to $-5 \leqslant x, y, z \leqslant 5$. The oak acorn is an example of a non-smooth interface featuring a tip and an edge. The surface error plot in Fig. 17 shows a maximum at the tip and some minima at and around the edge of the oak acorn.

The convergence plots in Fig. 17 for different ratios of the coefficients $\beta + /\beta^-$ demonstrate second order of accuracy in the L_∞, L_B, and L_2 norm for a moderate ratio $\beta^+/\beta^- = 1/80$. For large ratios of the coefficients $\beta^+/\beta^- = 1/1000$ and $\beta^+/\beta^- = 1000/1$ we see decreased convergence rates of 1.3 and 1.1, respectively, in the L_∞ norm, whereas the convergence rates in the L_B, and L_2 norm still show second order of accuracy. Furthermore, for $\beta^+/\beta^- = 1/1000$ we also observe a non-smooth convergence under grid refinement. However, it is known for interface problems that the error does not necessarily behave monotonically under grid refinement [28]. Some preliminary investigations suggest a possibility that this behaviour is due to non-smooth intersections of the trilinear ansatz on the side-faces of neighbouring cells. Unfortunately, the paper of Yu and Wei [42] does not provide comparable results for non-smooth interfaces with a high ratio of the coefficients β^+/β^-.

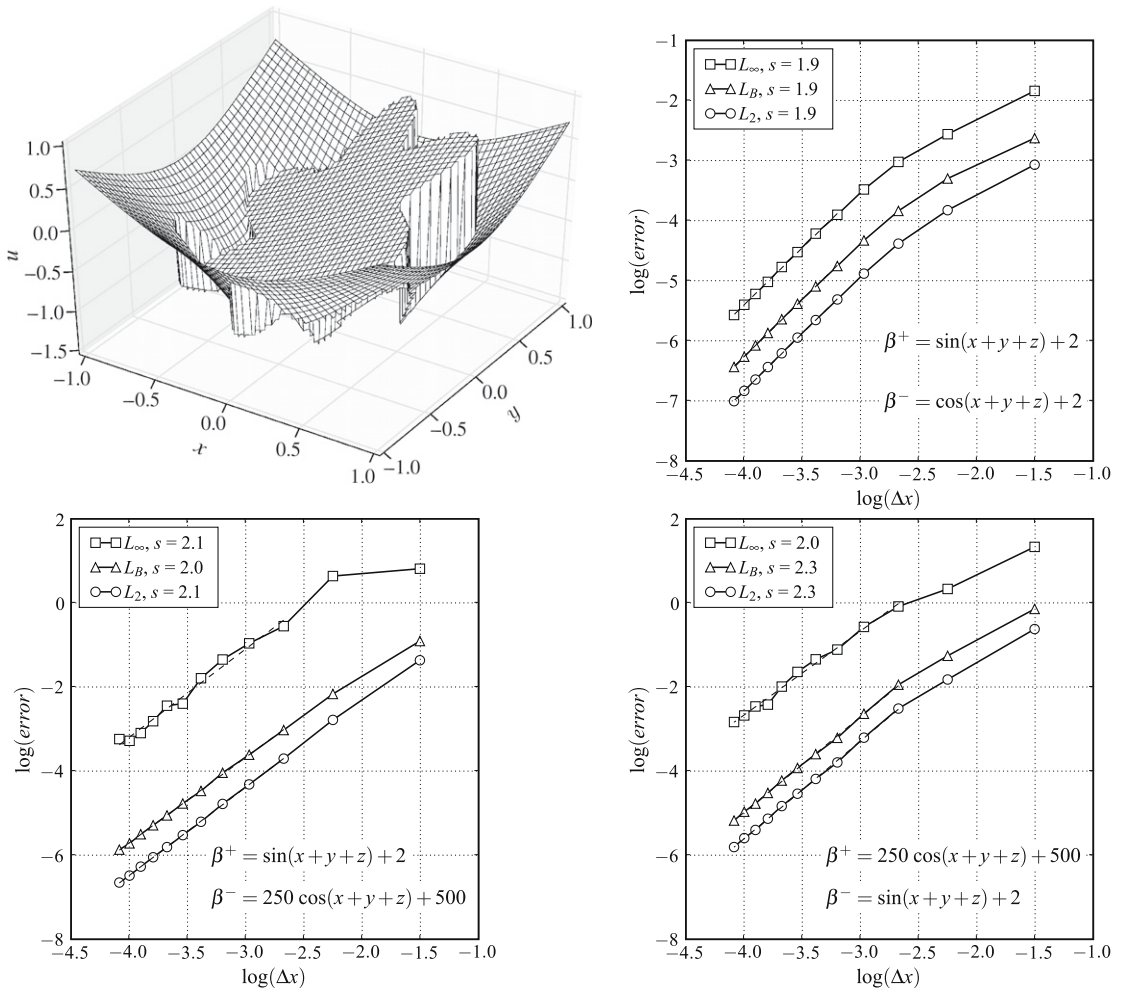

Fig. 15. Solution for example 3 at $z = 0.2$ and convergence results for different β-ratios for the star-example. The convergence rates s have been evaluated using a linear least square fit.

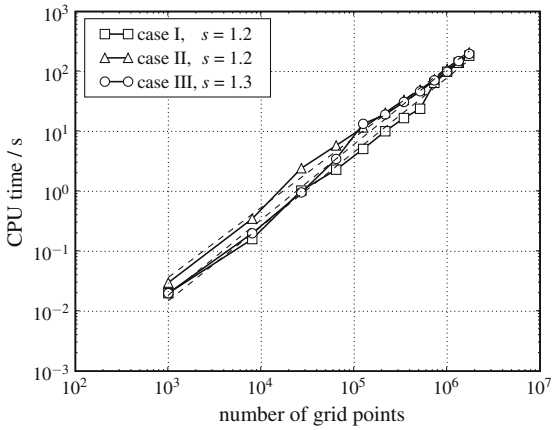

Fig. 16. CPU times for solving the resulting set of linear equations as a function of the total number of grid points and different ratios of the coefficient β. The scaling factors s has been evaluated using a linear least square fit.

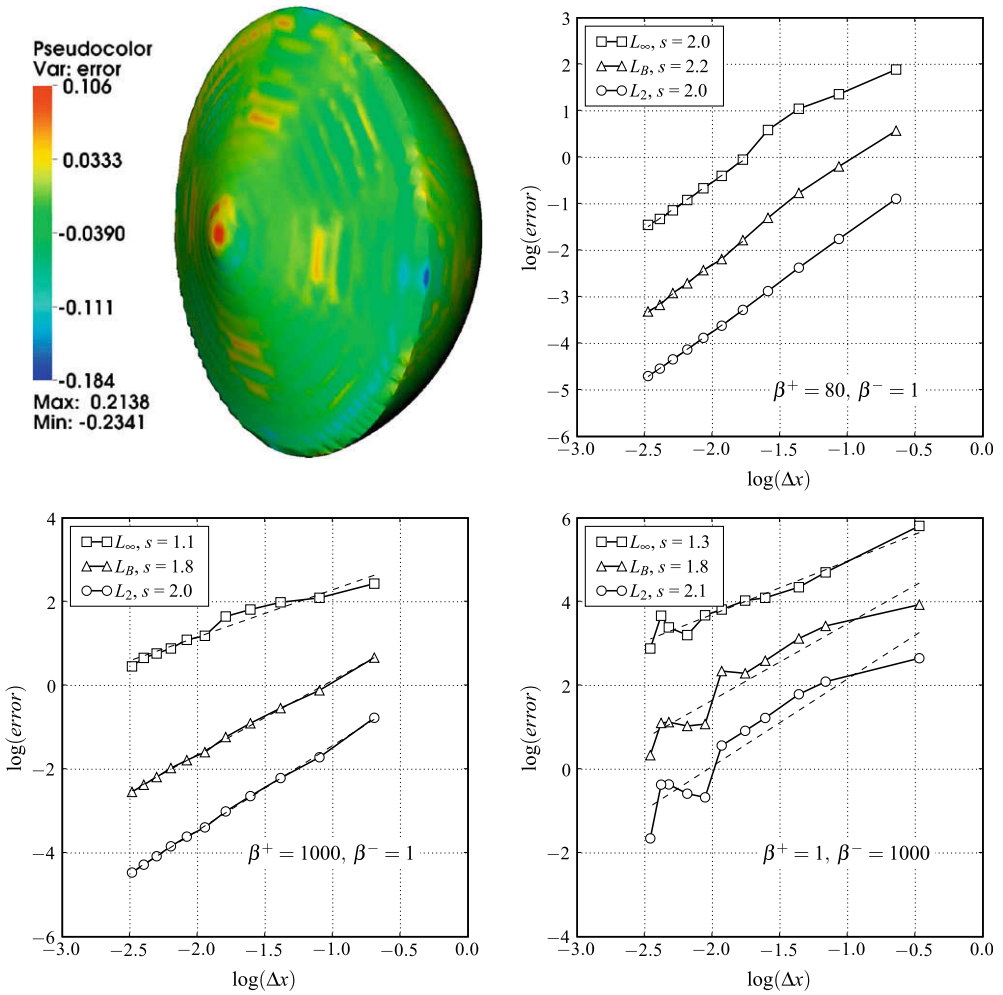

Fig. 17. Error at the interface and convergence results for the oak acorn (example 4). The surface error plot has been obtained on a grid with $100 \times 100 \times 100$ points. The convergence rates s have been evaluated using a linear least square fit.

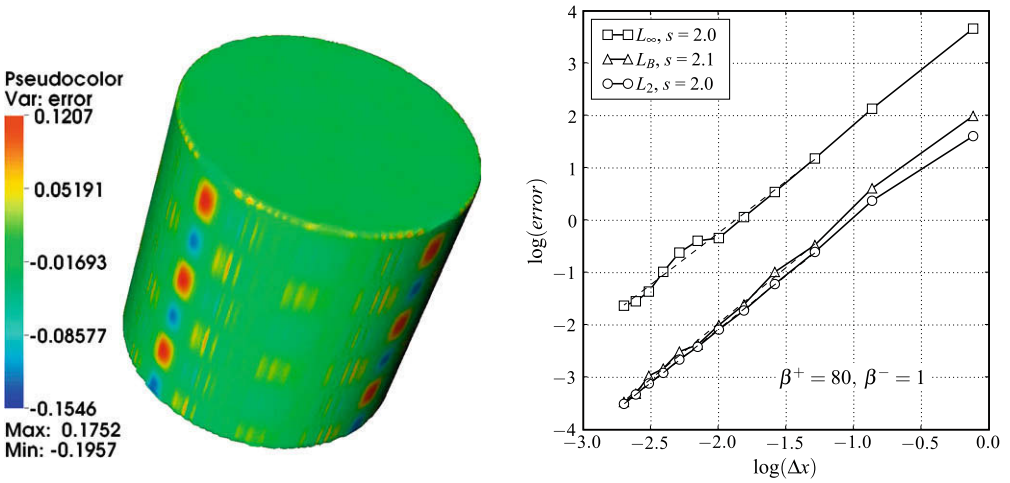

Fig. 18. Surface error plot and convergence results for a cylinder interface. The convergence rates s have been evaluated using a linear least square fit.

6.5. Example 5 – cylinder

The last example has also been taken from [42]. The interface is a cylinder with diameter π and height 2π, see Fig. 18. The computational domain has been set in accordance with [42] to $-4 \leqslant x, y \leqslant 4$ and $-2 \leqslant z \leqslant 8.4$. The analytic solutions and the coefficients on the two sides of the interface are given by (60). The cylinder is another example of a non-smooth interface featuring two edges.

Fig. 18 shows on the left hand side a surface error plot on a grid with a resolution of $120 \times 120 \times 120$ points. Although the analytic solution is identical to the example of the oak acorn, we observe the extreme values of the error not at the edges but on the sides of the cylinder correlating with the large amplitudes of the harmonic solution u^-, Eq. (60). Yu and Wei observed a similar behaviour of their method in [42].

On the right hand side of Fig. 18 we see second order of convergence in the L_∞, L_B, and L_2 norms. In contrast to the oak acorn example the errors in the L_B and L_2 norms show the same order of magnitude. The order of magnitude in the L_∞-norm is comparable to the example of the oak acorn.

7. Conclusion

We have developed a second order sharp interface finite volume method on Cartesian grids for the solution of elliptic equations in 3D with variable coefficients and discontinuities across an embedded interface.

In contrast to most sharp interface methods in the literature we use a finite volume approach on Cartesian grids using ideas from finite element methods in reconstructing the solution within grid cells. On cells which are intersected by the interface we apply a dual piecewise trilinear solution ansatz leading to a finite element for irregular cut-cells taking into account known jump conditions of the solution and the normal gradient across the interface. The work extends our 2D method presented in [28] to 3D and introduces new concepts for the construction of the double trilinear solution ansatz via a minimization approach. Singularities arising from vanishing partial volumes within a cell are removed in analogy to [28] by a two-term asymptotic approach. This asymptotic approach allows partial volumes down to machine accuracy without affecting the condition number of the local minimization problem.

In the 2D method [28] we were always able to impose exactly as many jump conditions as unknowns at the cost of another singularity besides vanishing partial volumes. However, the analog of the 2D method in 3D would have lead to additional singularities of the type mentioned in Remark 5.1. The present minimization approach avoids additional special asymptotics for this type of degenerated interface topologies and, therefore, simplifies the treatment of the different cut cell types.

Our discretisation scheme always leads to a compact 27-point stencil for the discrete Laplacian, with appropriately adjusted weights near the interface. The resulting set of linear equations is usually non-symmetric and reduces to a symmetric and diagonally dominant system in cases of equal coefficients $\beta^+ = \beta^-$. We used the BiCGSTAB solver preconditioned with the algebraic multigrid BOOMER of the public domain package `hypre` [6] for the solution of the system of linear equations. Near optimal scaling for all test cases including complex interfaces and large ratios of the coefficients β^+ and β^- was observed.

Except for the case of a non-smooth interfaces at high ratio of the coefficients all examples investigated in this work demonstrate locally second order of accuracy of the method.

Although the subject of this work is the numerical solution of elliptic equations we note that our trilinear finite element might be equally useful for the reconstruction of any other discontinuous function on Cartesian grids (e.g.: the velocity field in premixed combustion) and, of course, for finite element methods itself.

Problems of the type considered here become notoriously difficult to solve numerically for limitingly large ratios of the coefficients, say $\beta^+ : \beta^- \to \infty$. We will address this issue systematically, again using asymptotic methods, in a forthcoming publication.

Acknowledgment

R.K. thanks Deutsche Forschungsgemeinschaft, who have partially supported this work through Grants KL 611/14, SPP 1276 and SFB 557/B8.

References

[1] T. Belytschko, N. Möes, S. Usui, C. Parimi, Arbitrary discontinuities in finite elements, International Journal for Numerical Methods in Engineering 50 (2000) 993–1013.
[2] Z. Chen, J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numerische Mathematik 79 (1998) 175–202.
[3] R. Cortez, M. Minion, The blob projection method for immersed boundary problems, Journal of Computational Physics 138 (2000) 428–453.
[4] S. Deng, K. Ito, Z. Li, Three-dimensional elliptic solvers for interface problems and applications, Journal of Computational Physics 184 (2003) 215–243.
[5] B. Engquist, A.-K. Tornberg, R. Tsai, Discretization of Dirac delta functions in level set methods, Journal of Computational Physics 207 (2005) 28–51.
[6] R.D. Falgout, U.M. Yang, hypre: a library of high performance preconditioners, in: P.M.A. Sloot, C.J.K. Tan, J.J. Dongarra, A.G. Hoekstra (Eds.), Computational Science – ICCS 2002 Part III, Lecture Notes in Computer Science, vol. 23, Springer-Verlag, Berlin, 2002, pp. 632–641.

[7] F. Gibou, R.P. Fedkiw, A fourth order accurate discretization of the Laplace and heat equations on arbitrary domains with applications to the Stephan problem, Journal of Computational Physics 202 (2005) 577–601.
[8] S. Groß, A. Reusken, An extended pressure finite element space for two-phase incompressible flows with surface tension, Journal of Computational Physics 224 (2007) 40–58.
[9] A. Hansbo, P. Hansbo, An unfitted finite element method, based on Nitsche's method, for elliptic interface problems, Computer Methods in Applied Mechanics and Engineering 191 (47–48) (2002) 5537–5552.
[10] P. Hansbo, C. Lovadina, I. Perugia, G. Sangalli, A Lagrange multiplier method for the finite element solution of elliptic interface problems using non-matching meshes, Numerische Mathematik 100 (1) (2005) 91–115.
[11] S. Hou, X. Liu, A numerical method for solving variable coefficient elliptic equation with interfaces, Journal of Computational Physics 202 (2005) 411–445.
[12] H. Johansen, P. Colella, A cartesian grid embedded boundary method for Poisson's equation on irregular domains, Journal of Computational Physics 147 (1998) 60–85.
[13] S.Y. Kadioglu, M.L. Minion, R. Klein, A fourth-order auxiliary variable projection method for zero-mach number gas dynamics, Journal of Computational Physics 227 (2008) 2012–2043.
[14] R. Klein, Numerics in combustion, in: L. Vervisch, D. Veynante (Eds.), Introduction to Turbulent Combustion, Brussels, Belgium, January 6–9, 1999, von Karman Institute for Fluid Dynamics.
[15] R. Klein, N. Botta, L. Hofmann, A. Meister, C.. Munz, S. Roller, T. Sonar, Asymptotic adaptive methods for multiscale problems in fluid mechanics, Journal of Engineering Mathematics 39 (2001) 261–343.
[16] R.J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM Journal on Numerical Analysis 31 (4) (1994) 1019–1044.
[17] R.J. LeVeque, Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, SIAM Journal on Scientific Computing 18 (1997) 709–735.
[18] Z. Li, A fast iterative algorithm for elliptic interface problems, SIAM Journal of Numerical Analysis 35 (1) (1998) 230–254.
[19] Z. Li, K. Ito, Maximum principle preserving schemes for interface problems with discontinuous coefficients, SIAM Journal of Scientific Computing 23 (1) (2001) 339–361.
[20] Z. Li, K. Ito, The immersed interface method – numerical solutions of PDEs involving interfaces and irregular domains, SIAM Frontiers in Applied Mathematics 33 (2006).
[21] Z. Li, T. Lin, X. Wu, New cartesian grid methods for interface problems using finite element formulation, Numerische Mathematik 96 (2003) 61–98.
[22] X. Liu, R.P. Fedkiw, M. Kang, A boundary condition capturing method for poisson's equation on irregular domains, Journal of Computational Physics 160 (1) (2000) 151–178.
[23] X.-D. Liu, T.C. Sideris, Convergence of the ghost fluid method for elliptic equations with interfaces, Mathematics of Computation 72 (244) (2003) 1731–1746.
[24] W.E. Lorensen, H.E. Cline, Marching cubes: a high resolution 3D surface construction algorithm, Computer Graphics 21 (4) (1987) 163–169.
[25] A. Mayo, The fast solution of Poisson's and the biharmonic equation on irregular regions, SIAM Journal on Numerical Analysis 21 (2) (1984) 285–299.
[26] A. Mayo, Fast high order accurate solutions of Laplace's equation on irregular domains, SIAM Journal on Science and Statistical Computing 6 (1) (1985) 144–157.
[27] N. Möes, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, International Journal for Numerical Methods in Engineering 46 (1999) 131–150.
[28] M. Oevermann, R. Klein, A cartesian grid finite volume method for elliptic equations with variable coefficients and embedded interfaces, Journal of Computational Physics 219 (2006) 749–769.
[29] M. Oevermann, R. Klein, M. Berger, J. Goodman, A projection method for incompressible two-phase flow with surface tension, Technical Report 00-17, Konrad-Zuse-Zentrum, Berlin, 2000.
[30] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer-Verlag, New York, 2003.
[31] C.S. Peskin, Numerical analysis of blood flow in the heart, Journal of Computational Physics 25 (1977) 220–252.
[32] T. Schneider, N. Botta, K.J. Geratz, R. Klein, Extension of finite volume compressible flow solvers to multi-dimensional variable density zero Mach number flows, Journal of Computational Physics 155 (1999) 248–286.
[33] T. Schneider, R. Klein, Overcoming mass losses in level-set-based interface tracking schemes, in: Second International Symposium on Finite Volumes for Complex Applications – Problems and Perspectives, Berlin, Germany, July 19–22, 1999.
[34] Th. Schneider, Verfolgung von Flammenfronten und Phasengrenzen in schwach kompressiblen Strömungen, Dissertation, RWTH Aachen, 2000.
[35] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, New York, 2006.
[36] E. Süli, Convergence of finite volume schemes for poisson's equation on nouniform meshes, SIAM Journal on Numerical Analysis 28 (5) (1991) 1419–1430.
[37] P. Terhoeven, Ein numerisches Verfahren zur Berechnung von Flammenfronten bei kleiner Mach-Zahl, Dissertation, RWTHAachen, 1998.
[38] A.-K. Tornberg, B. Engquist, Regularization techniques for numerical approximation of PDES with singularities, Journal of Scientific Computing 19 (2003) 527–552.
[39] A.-K. Tornberg, B. Engquist, Numerical approximations of singular source terms in differential equations, Journal of Computational Physics 200 (2004) 462–488.
[40] S. Vater, A new projection method for the zero Froude number shallow water equations, Technical Report 97, Potsdam Institute for Climate Impact Research, 2005.
[41] S. Vater, R. Klein, Stability of Cartesian grid projection methods for zero Froude number shallow water flows, Numerische Mathematik, in revision.
[42] S. Yu, G.W. Wei, Three-dimensional matched interface and boundary (MIB) method for treating geometric singularities, Journal of Computational Physics 227 (2007) 602–632.
[43] Y.C. Zhou, G.W. Wei, On the fictitious-domain and interpolation formulations of the matched interface and boundary (MIB) method, Journal of Computational Physics 219 (2006) 228–246.
[44] Y.C. Zhou, S. Zhao, M. Feig, G.W. Wei, High order matched interface and boundary (MIB) schemes for elliptic equations with discontinuous coefficients and singular sources, Journal of Computational Physics 213 (2006) 1–30.