Journal of Computational Physics 250 (2013) 77–105

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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

A second order virtual node algorithm for Stokes flow problems with interfacial forces, discontinuous material properties and irregular domains

Diego C. Assêncio [a,*], Joseph M. Teran [b]

[a] University of California, Los Angeles, Department of Physics and Astronomy, 430 Portola, Physics and Astronomy Building, Los Angeles, CA 90095, United States
[b] University of California, Los Angeles, Department of Mathematics, 520 Portola Plaza, Math Sciences Building 6363, Los Angeles, CA 90095, United States

ARTICLE INFO

Article history:
Received 25 December 2011
Received in revised form 13 April 2013
Accepted 24 April 2013
Available online 22 May 2013

Keywords:
Stokes flow
Virtual node algorithms
Interface problems
Finite element methods
Finite difference methods
Ghost fluid methods

ABSTRACT

We present numerical methods for the solution of the Stokes equations that handle interfacial discontinuities, discontinuous material properties and irregular domains. The discretization couples a Lagrangian representation of the material interface with Eulerian representations of the fluid velocity and pressure. The methods are efficient, easy to implement and yield discretely divergence-free velocities that are second order accurate in L^∞. For the special case of continuous fluid viscosity, we present a method that decouples the Stokes equations into three Poisson interface problems which we use the techniques in Bedrossian (2010) [1] to solve. We also solve a fourth Poisson equation to enforce a discrete divergence free condition in this case. We discretize all equations using an embedded approach on a uniform MAC grid employing virtual nodes and duplicated cells at the interfaces. These additional degrees of freedom allow for accurate resolution of discontinuities in the fluid stress at the material interface. In the case of discontinuous viscosity, we require a Lagrange multiplier term to enforce continuity of the fluid velocity. We provide a novel discretization of this term that accurately resolves constant pressure null modes. We show that the accurate resolution of these modes significantly improves performance. The discrete coupled equations for the velocity, pressure and Lagrange multipliers are in the form of a symmetric KKT system. However, if both fluids have the same viscosity then all four linear systems involved are symmetric positive definite with three of the four having the standard 5-point Laplace stencil everywhere. Numerical results indicate second order accuracy for the velocities and first order accuracy for the pressure in the general case. For the continuous viscosity case, numerical results indicate second order accuracy for both velocities and pressure.

1. Introduction

We consider the Stokes equations for two-phase, highly viscous incompressible flow in irregular domains:

$$\nabla \cdot \sigma = \mu \Delta \mathbf{u} - \nabla p = -\mathbf{f}, \quad \mathbf{x} \in \Omega \setminus \Gamma, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \quad \mathbf{x} \in \Omega \setminus \Gamma, \tag{2}$$

$$\mathbf{u}(\mathbf{x}) = \mathbf{a}(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega, \tag{3}$$

* Corresponding author. Tel.: +1 310 756 3794.
E-mail addresses: dassencio@physics.ucla.edu (D.C. Assêncio), jteran@math.ucla.edu (J.M. Teran).

0021-9991/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2013.04.041

$$[\mathbf{u}](\mathbf{x}) = \mathbf{0}, \quad \mathbf{x} \in \Gamma, \tag{4}$$

$$[\sigma \cdot \mathbf{n}](\mathbf{x}) = \mathbf{f}^i, \quad \mathbf{x} \in \Gamma. \tag{5}$$

Here p is the pressure, $\mathbf{u} = (u, v)$ is the fluid velocity, $\sigma = \mu\left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial \mathbf{u}}{\partial \mathbf{x}}^T\right) - p\mathbf{I}$ is the fluid stress, Γ is the interface between the two phases, $\mathbf{f}$ is the body force density and $\mathbf{f}^i$ is force supported on the interface between the two phases (e.g. surface tension). The interface Γ is generally a codimension one closed curve that divides the domain into an interior region Ω^- and exterior region Ω^+ such that $\Omega = \Omega^- \cup \Omega^+ \cup \Gamma$ (see Fig. 1(a)). We let $\mathbf{n}$ denote the outward unit normal to Ω^- at a point $\mathbf{x} \in \Gamma$ and define $[v](\mathbf{x}) := v^+(\mathbf{x}) - v^-(\mathbf{x}) = \lim_{\epsilon \to 0^+} v(\mathbf{x} + \epsilon\mathbf{n}) - \lim_{\epsilon \to 0^+} v(\mathbf{x} - \epsilon\mathbf{n})$ as the "jump" of the quantity v across the interface Γ. Unless otherwise stated, we assume the curve Γ is smooth.

Due to the irregular geometry of the domain boundaries and interfaces, a natural approach to the numerical approximation is the finite element method (FEM) with unstructured meshes that conform to the geometry of Γ. However, meshing complex interface geometries can prove difficult and time-consuming when the interface frequently changes shape. Also, many numerical methods, such as finite differences, do not naturally apply to unstructured meshes. These concerns motivated the development of embedded (or, immersed) methods that approximate solutions on Cartesian grids or structured meshes that do not conform to the interface. Despite advances in this direction, embedded methods that retain higher order accuracy in L^∞ typically introduce relatively difficult linear algebra problems and complex implementations that often require significant effort to adapt to general applications.

With these concerns in mind, we introduce second-order virtual node methods for approximating the two-phase Stokes equations with irregular embedded interfaces and boundaries on a uniform Cartesian Marker and Cell (MAC) grid [2]. We use regular grids because it simplifies the implementation, permits straightforward numerical linear algebra and achieves higher order accuracy in L^∞. Our approach uses duplicated Cartesian grid cells along the interface to introduce additional "virtual" nodes that accurately resolve discontinuous quantities. This also naturally allows for discontinuities in material properties such as viscosity and density. Interface cells are cut using a level set to allow for accurate evaluation of integrals needed for the numerical stencils. In the case of discontinuous material properties, the discretization of the viscous stress forces as well as the divergence-free and jump constraints are constructed from a variational formulation that yields symmetric numerical stencils. Lagrange multipliers enforce continuity of the fluid velocity across duplicated interface cells. Our cutting procedure for interface cells is designed to resolve the constant pressure null modes of the weak formulation. We show that the accurate resolution of these modes significantly accelerates the solution of the linear systems involved. In the general case, the discrete linear system is of KKT (Karush–Kuhn–Tucker) type. If both fluids have the same viscosity, we can formulate the Stokes problem as three Poisson equations with jump conditions to allow us to leverage our previous work in [1]. In this case, we also solve a final Poisson equation over the pressure grid to enforce a discrete divergence free condition yielding a total of four Poisson solves per Stokes solve. In all but the final Poisson equation, our approach yields the standard 5-point difference stencil away from embedded boundaries (notably, we have the standard 5-point stencil *across* the interface between the phases). Interfaces are represented by Lagrangian particles for straightforward advection (see Fig. 1(b)). We provide a robust method for transforming the Lagrangian representation into a level set defined over each pressure grid. Our representation of the interface ensures that the discrete interface conditions are enforced consistently for the staggered variables on the MAC grid. Numerical experiments indicate second order accuracy in L^∞ for the velocity and first order accuracy in L^∞ for pressure in general. In the case of continuous viscosity, numerical experiments indicate second order accuracy for both the velocities and the pressure.

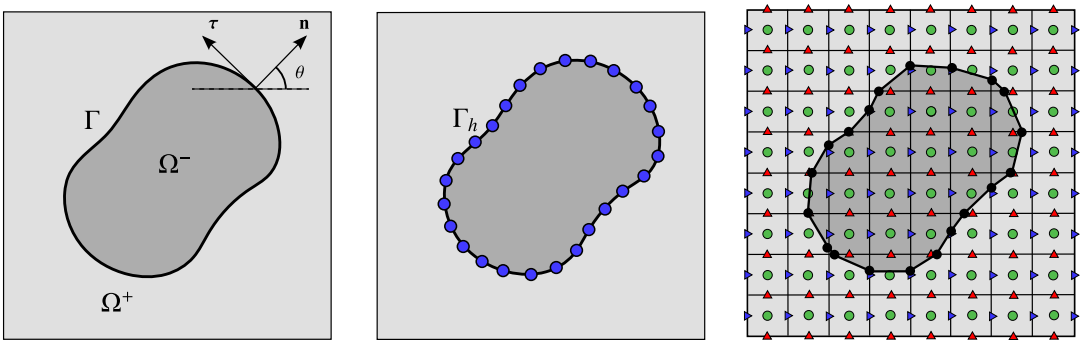

Fig. 1. (a) Interface Γ separates the fluid domain $\Omega = \Omega^+ \cup \Omega^- \cup \Gamma$ into its exterior Ω^+ and interior Ω^-. The figure shows the unit normal and tangent vectors $\mathbf{n}$ and $\boldsymbol{\tau}$. If θ is the angle between $\mathbf{n}$ and the horizontal x axis, then $\mathbf{n} = (\cos\theta, \sin\theta)$ and $\boldsymbol{\tau} = (-\sin\theta, \cos\theta)$. (b) Discrete Lagrangian interface Γ_h. (c) Fluid variables are discretized over an Eulerian MAC grid. A level set representation of the interface is defined over the nodes of the pressure sub grid. We approximate the zero isocontour of the level set with a piecewise linear curve $\hat{\Gamma}_h$. The black dots represent the intersection of the interface $\hat{\Gamma}_h$ and the edges of each pressure grid cells.

2. Existing methods

We build on the recent virtual node methods for elliptic interface problems developed in [1,3,4]. Our method is second order in L^∞ for the velocities and, in general, first order in L^∞ for the pressure (with second order in the case of continuous viscosity). This is achieved with modest linear algebra demands. Specifically, all systems are sparse and symmetric (symmetric positive definite in the case of continuous viscosity). Furthermore, our approach yields discretely divergence-free velocities. Although many researchers have developed embedded methods for the Stokes equations with interfacial discontinuities, our approach is the first to support this feature set. In our discussion of existing approaches, we will focus only on embedded (or immersed) methods that avoid unstructured meshing when addressing boundary and interface conditions at irregular geometric boundaries.

Embedded techniques use a computational domain that simply encompasses rather than geometrically adheres to the interface. A good review of embedded methods is given by Lew et al. in [5]. They point out that these techniques originated with the papers of Harlow and Welch [6] and Charles Peskin [7]. Peskin developed the immersed boundary method (IBM) to simulate blood flow in the heart [8–11], but it has also been applied to many other problems. A summary of the development of the immersed boundary method and its applications can be found in [12]. Despite its vast popularity and considerable ease of implementation, the IBM suffers from its use of regularized delta functions to represent singular forces acting on interfaces. This renders the method first order accurate and implies that the physical characteristics of the flow near the interfacial boundaries are not accurately resolved. Singular forces acting on the interface lead to discontinuities in the pressure and velocity derivatives that the IBM may fail to accurately resolve [13]. However, for sufficiently smooth problems in which the interfaces are thick instead of infinitesimally thin the IBM can achieve second order accuracy [14,15]. Adaptive versions of the IBM were developed in [16] to enhance convergence over coarse grids but the results were still only first order accurate. Another deficiency of the IBM is poor conservation of volume near the interface. The seriousness of this problem, especially for the simulation of blood flow in the heart, motivated the development of a better volume conserving version of the IBM in [17].

Many methods were designed to improve the order of accuracy of the original IBM. The Immersed Interface Method (IIM) is perhaps the most popular example of this. The IIM was first developed for elliptic equations with interfacial discontinuities [18] and later applied to Stokes flow [19]. The IIM achieves second (and higher) order accuracy by capturing interfacial discontinuities in the pressure, the velocities and their derivatives in a sharp manner. The IIM has been used in many fluid flow problems including interface and rigid boundary problems [13,20–22], Hele-Shaw flow [23] and also problems in which the viscosity is discontinuous across the interfaces [24–26]. Arbitrarily high orders of accuracy have been achieved [27,28]. The method is considerably more difficult to implement than the IBM and most applications are in two space dimensions as a result. However, researchers have applied the IIM to three dimensional flows [29]. A limitation of the IIM is the lack of symmetry in discretizations arising from problems with discontinuous coefficients. This imposes an obstacle on the overall speed of these methods since fast linear solvers such as conjugate gradients cannot be used. However, it should be noted that the IIM can yield symmetric matrices for continuous viscosity Stokes flow. In the case of discontinuous viscosity, the IIM is complicated by the lack of explicit knowledge of jump conditions on the fluid variables (and their derivatives) along the interface. As shown in [24], this can complicate the problem significantly.

One method that is capable of always guaranteeing a symmetric discretization is the Ghost Fluid Method (GFM). Initially applied to the Poisson equation with interfacial jumps and variable coefficients [30], the GFM was also used to simulate multiphase incompressible flow [31]. Unfortunately, the GFM is only capable of achieving first order results for interface problems. Also, in [31] the GFM treats viscous terms explicitly because they cannot be decoupled in the case of discontinuous viscosity.

Some of the first embedded methods were fictitious domain methods by Hyman [32] and Saul'ev [33]. The fictitious domain approach has been used with incompressible materials in a number of works [34–42]. These approaches embed the irregular geometry in a simpler domain for which fast solvers exist (e.g. Fast Fourier Transforms). The calculations include fictitious material in the complement of the domain of interest. A forcing term (often from a Lagrange multiplier) is used to maintain boundary conditions at the irregular geometry. Although these techniques naturally allow for efficient solution procedures, they depend on a smooth solution across the embedded domain geometry for optimal accuracy, which is not typically possible.

The extended finite element method (XFEM) and related approaches in the finite element literature also make use of geometry embedded in regular elements. Although originally developed for crack-based field discontinuities in elasticity problems, these techniques are also used with embedded problems in irregular domains. Daux et al. first showed that these techniques can naturally capture embedded Neumann boundary conditions [43,44]. Enforcement of Dirichlet constraints is more difficult with variational cut cell approaches [45,5] and typically involves a Lagrange multiplier or stabilization. Dolbow and Devan recently investigated the convergence of such approaches with incompressible materials and point out that much analysis in this context remains to be completed [46]. Despite the lack of thorough analysis, such XFEM approaches appear to be very accurate and have been used in many applications involving incompressible materials in irregular domains [47–51].

There are also many finite difference (FDM) and finite volume methods (FVM) that utilize cut uniform grid cells. Many of these methods have been developed in the context of incompressible flow. For example, Almgren et al. use cut uniform

bilinear cells to solve the Poisson equation for pressures in incompressible flow calculations [52]. Marelle et al. use collocated grids and define sub cell interface and boundary geometry in cut cells via level sets [53]. Ng et al. also use level set descriptions of the irregular domain and achieve second order accuracy in L^∞ for incompressible flows [54]. The approach of Batty and Bridson is similar, but not as accurate [55]. Cut cell FDM and FVM have also been developed for incompressible and nearly incompressible elastic materials. Bijelonja et al. use cut cell FVM to enforce incompressibility more accurately than is typically seen with FEM [56]. Beirão da Veiga et al. use polygonal FVM cells to avoid remeshing with irregular domains [57]. Barton et al. [58] and Hill et al. [59] use cut cells with Eulerian elastic/plastic flows.

3. Description of numerical methods

We couple a Lagangian representation of the interface (Γ_h) with an Eulerian representation of the fluid velocity and pressure (see Fig. 1). The Eulerian discretization is defined over a staggered MAC grid. In the general case, we use a weak form of the Stokes equations and simultaneously solve for the pressure, velocity and Lagrange multipliers. In this case, we use a piecewise-bilinear velocity and piecewise-constant pressure as in [60] (see Fig. 2). In the case of continuous viscosity, we reformulate the problem into three Poisson interface problems which we use our previous work in [1] to solve. This decoupling of the variables prevents the need for additional Lagrange multiplier unknowns and simplifies the numerical linear algebra. We use a level set representation of the Lagrangian interface curve Γ_h. This level set naturally defines a consistent representation of the interface on each of the staggered velocity and pressure grids. Interface cells in each grid can then be duplicated consistently. In the case of discontinuous viscosity, our interface cutting procedure is specifically designed to resolve the constant pressure null modes of the weak formulation. We show that this property of the discretization significantly improves the performance of our solver. Advection for the particles in the Lagrangian Γ_h is explicit and is done with the interpolated local fluid velocity. Interfacial forces are defined on the Lagrangian particles and then transferred to the cut cells to define stress jump conditions for the Eulerian variables. The complete procedure for advancing one time step is:

1. Compute interface forces (from surface tension or elasticity) at all particles in Γ_h; details in Sections 4.4 and 4.5.
2. Compute the level set from Γ_h and transfer stress jump conditions; details in Sections 3.1.
3. Construct discrete stencils that respect the cut grid cells; details for discontinuous viscosity in Section 3.2.1, for continuous viscosity in Section 3.2.2 and for irregular domain Dirichlet velocity in Section 3.2.3.
4. Solve the systems for the velocity, pressure and (in the case of discontinuous viscosity and/or irregular domains) Lagrange multipliers; details in Section 4.2.
5. Interpolate velocities from the MAC grid to the Lagrangian interface and update particle positions using forward Euler; details in Section 4.3.

We will now describe each of these steps in detail. We give only limited discussion of the computation of the Lagrangian interface forces $\mathbf{f}^i$ (Section 4.4) and the interpolation from the MAC grid to the Lagrangian interface (Section 4.3) as these aspects are relatively straightforward. Also, for clarity we note that we use two intermediate representations of the interface in the process of defining stress jump conditions: $\tilde{\Gamma}_h$ and $\hat{\Gamma}_h$. $\tilde{\Gamma}_h$ is simply a subdivided version of Γ_h needed so that the spac-

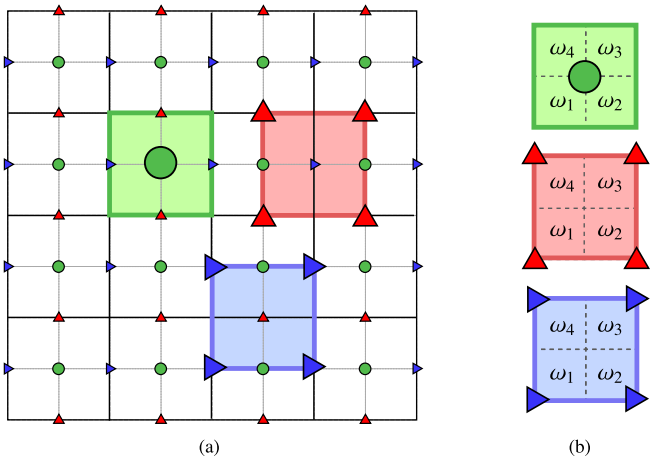

Fig. 2. (a) Cells on the u, v and p sub-grids of the MAC grid. Triangles and circles indicate where the velocity and pressure degrees of freedom are located. The blue cell with corner triangles pointing to the right is a u grid cell; the red cell with corner triangles pointing up is a v grid cell and the green cell with a single center circle is a p cell. Notice that while u and v cells have four degrees of freedom per cell, p cells have only a single degree of freedom. (b) Cells from Figure (a) with their four subcells indexed as ω_1, ω_2, ω_3, and ω_4. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ing of Lagrangian points are separated by a distance no bigger than the grid spacing. $\hat{\Gamma}_h$ is an approximation of the zero isocontour of the level set and is needed for grid duplication and quadrature.

3.1. Computation of the level set from Γ_h

A Lagrangian representation of the interface is convenient for computing interfacial forces and also for explicit Euler update of the interface geometry. We therefore represent the discrete interface Γ_h with a sequence of Lagrangian points $\mathbf{x}_i$ where $i \in \{0, 1, 2, \ldots M-1\}$. The points are connected by segments and form a closed curve as shown in Fig. 1(b) (we also support multiple closed curves). However, we also require an Eulerian representation of the interface for the discretization of the fluid and pressure variables. We duplicate cells in the MAC grid that are crossed by the interface to introduce additional degrees of freedom that allow us to accurately capture discontinuities present at the interface. In order to complete this duplication procedure, we must know which cells in the MAC grid intersect the interface and we must know which velocity and pressure grid nodes are inside or outside the interface. We determine this with a level set approximation to Γ_h. For the special case of continuous viscosity, we define the level set over a "doubly fine" grid that contains each of the sub-grids in the MAC grid. We omit discussion of this case until Section 3.2.2. In the case of discontinuous viscosity, we define the level set over the pressure grid (see Figs. 1(c) and 3). We define the pressure grid to consist of cells centered around pressure degrees of freedom (see Figs. 2 and 3). These choices are motivated by the constraint that the normal to the interface must be approximated as constant over each cell on the pressure grid. The necessity of this constraint is outlined in Section 3.2.1.4. Level set values are defined at the corner nodes of the pressure grid cells (see Fig. 3). This level set can naturally be interpreted as having a constant normal over pressure cells. The discrete Eulerian interface is perturbed to prevent its isocontour from intersecting any of the velocity nodes on the MAC grid. This correction is necessary for our discretization to prevent cut cells on the u and v grids from having very small material regions as in [1].

3.1.1. Level set definition over the p grid

We define u, v and p sub-grids of the MAC grid. These are shown in Fig. 2. The u and v grids are node-centric in that the degrees of freedom are located at grid nodes. The p grid however is cell-centric in that p degrees of freedom are located at cell centers. We use h to denote the grid cell size (which we assume is the same in both x and y directions). We define the level set representation of the grid over the p grid. However, unlike the pressure degrees of freedom, we store the signed distance values at the nodes of the grid (see Fig. 3(b)). Specifically, we first compute the exact signed distance to Γ_h on the corner nodes of the pressure cells intersecting Γ_h. Any node whose four surrounding cells do not intersect Γ_h is set to positive or negative h depending on whether the node is inside Ω_h^+ or Ω_h^- respectively (see Fig. 4). To simplify the process of determining which interface segments in Γ_h intersect which cells on the pressure grid, we first ensure that each segment on the interface is smaller than the cell width h of the grid. This is done by adding nodes $\bar{\mathbf{x}}_i$ to subdivide the segments in Γ_h (see Fig. 5). Note that this subdivision is only done temporarily for the computation of nodal level set values; the computation of the interfacial forces uses the original set of segments Γ_h. We will denote this subdivided discrete interface as $\tilde{\Gamma}_h$. Next, we perturb the nodes $\bar{\mathbf{x}}_i$ of $\tilde{\Gamma}_h$ to prevent them from falling directly on the edges of any cell (see Fig. 6). Specifically, we first determine which pressure grid cell contains each $\bar{\mathbf{x}}_i$. We then clamp this node toward the cell center in a dimension by dimension fashion until it is at least as far as tolerance αh away from edges of the cell (typically we use $\alpha = 10^{-6}$). Once the nodes are perturbed, we can unambiguously determine which segments in $\tilde{\Gamma}_h$ intersect which edges in the pressure grid cell. Cell edges cut an even number of times are considered to be uncut. All possible cases are illustrated in Fig. 7(a). In order to avoid ambiguities, cases (4) and (7) require special treatment. We treat case (4) as a sub-case of case (3) and also case (7) as a sub-case

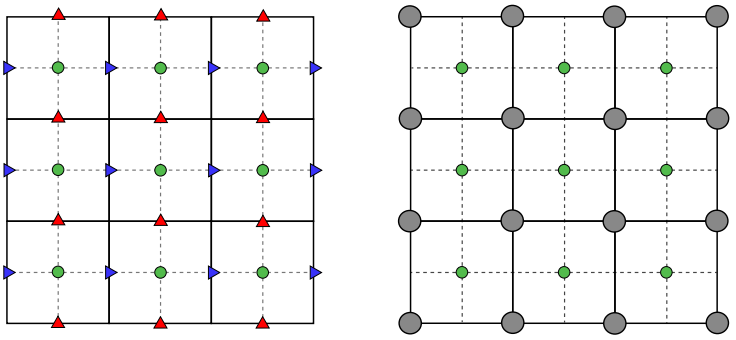

Fig. 3. Figure (a) shows the original MAC grid with the pressure cells drawn. The edges of the subcells are drawn with dashed lines. Figure (b) shows the nodes where the level set is computed drawn as large circles. These nodes lie on the corners of each pressure grid cell (note that pressure degrees of freedom are not stored there, but instead at the small green nodes at the center of each p cell); u and v grid nodes are omitted for clarity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

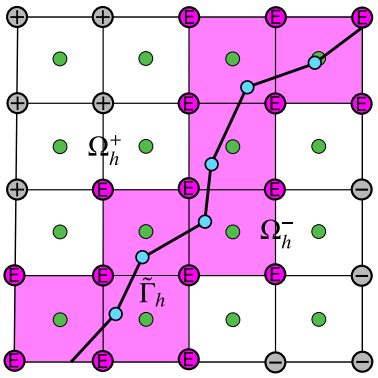

Fig. 4. The level set values are computed exactly on the corner nodes of pressure cells which are crossed by the interface $\tilde{\Gamma}_h$. These cut cells are shaded in pink, their nodes are labeled with an "E". The level set values are set to h on the nodes labeled with a "+" and to $-h$ on the nodes labeled with a "−". The "+" and "−" nodes lie on the side of Ω_h^+ and Ω_h^- respectively.

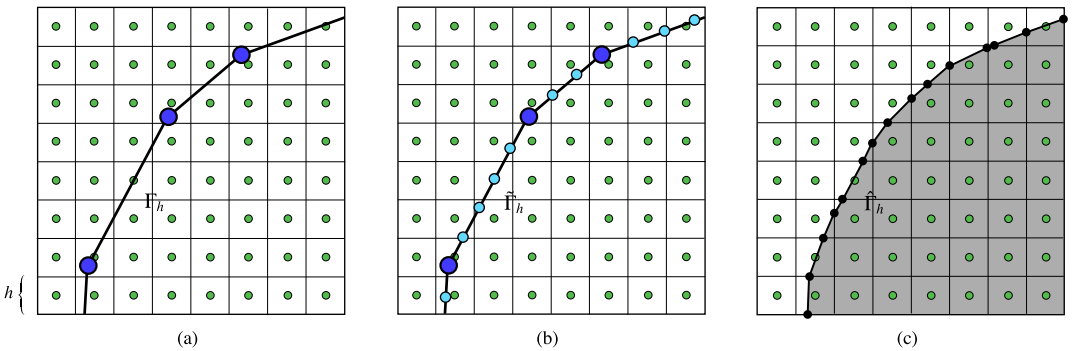

Fig. 5. Figure (a) shows the original discrete Lagrangian interface Γ_h with nodes $\mathbf{x}_i$ (large blue circles). Note that we exaggerate the separation between particles in Γ_h for illustration purposes. Figure (b) shows the subdivided $\tilde{\Gamma}_h$. The added nodes ($\tilde{\mathbf{x}}_i$) are indicated with smaller light blue circles. Nodes are added until all segments of $\tilde{\Gamma}_h$ are shorter than the cell width h of the grid. Figure (c) shows the Lagrangian approximation $\hat{\Gamma}_h$ of the discrete interface Γ_h which is generated by the level set computed from Γ_h on the corners of the pressure grid cells. The black circles denote the positions at which $\hat{\Gamma}_h$ crosses the edges of each pressure cut cell. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

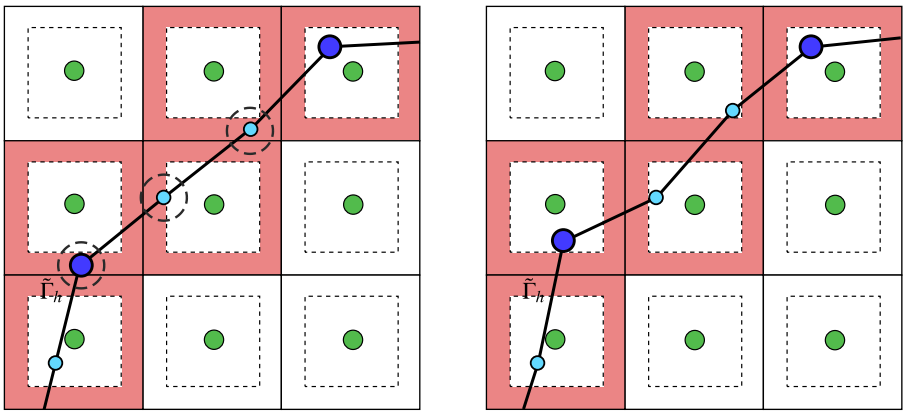

Fig. 6. The figure on the left shows a piece of the interface $\tilde{\Gamma}_h$ with three (circled) nodes lying too close to the edges of the pressure grid cells they fall into. The figure on the right shows the perturbation applied to these nodes. The shaded regions depict the thresholds used for determining if a node is too close to a grid cell edge. Note that the size of these regions is exaggerated for ease of visualization. In practice they are set to have a width proportionate to h.

of case (6). For example, in case (4) the node A is within a small tolerance of the segment PQ, P is in the lower-left cell and Q is in the upper-right cell (see Fig. 7(b)). Here, we automatically consider edges AC and AB as intersecting edge PQ. Otherwise if node A is sufficiently far from edge PQ we explicitly compute intersections between edge PQ and edges AC, AB, AE and AD to determine if we are in case (5) or case (3).

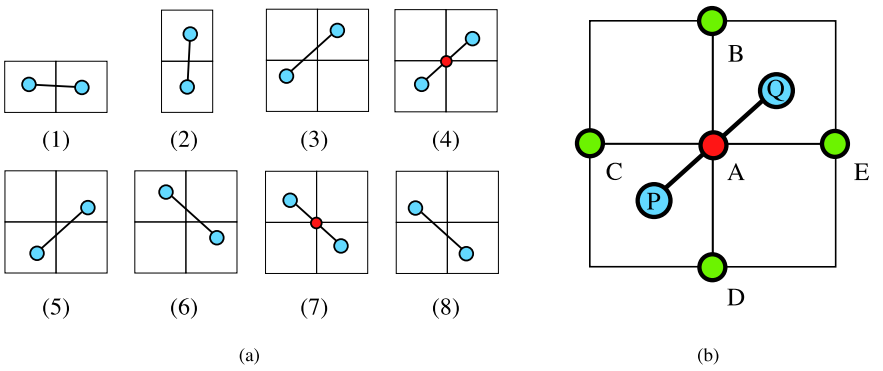

Fig. 7. Segment-cell collision detection. Figure (a) shows the eight cases to be considered when determining which cell edges are cut by a given segment of the discrete interface $\tilde{\Gamma}_h$. Cases (4) and (7) happen when a segment intersects four cells simultaneously by crossing a corner node (shown as small red circles in the figure). Figure (b) shows a case which requires special attention. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Once we have determined which subdivided interface segments in $\tilde{\Gamma}_h$ intersect which pressure grid cell edges, we compute the exact distance to $\tilde{\Gamma}_h$ for all corner nodes incident on a pressure grid cell. That is, for all pressure grid cell corner nodes with at least one of its four incident pressure cells cut by a segment in $\tilde{\Gamma}_h$, we compute the analytic distance from that node to each of the segments that intersect any of its incident cells and the minimum is defined as the distance from the node to $\tilde{\Gamma}_h$. We set the distance to be h for all pressure cell nodes not incident on a cell that intersects $\tilde{\Gamma}_h$. Finally, we determine the sign of the level set values depending on whether or not the nodes they correspond to are inside or outside $\tilde{\Gamma}_h$ (see Fig. 4). This is done with a flood-fill approach. We assume the bottom left corner node of the bottom left pressure grid cell is in Ω_h^+, we then sweep through the nodes of pressure cells in a dimension-by-dimension fashion. The sign of the next node in the sweep is the same as the previous node if the edge connecting them is crossed an even number of times, otherwise it is given the opposite sign.

3.1.2. Definition of u and v interface cells

The level set on the pressure grid naturally defines interface pressure cells as those with any vertices having signed distance values with opposite signs. However, we also need to know which u and v cells cross the interface. The velocity subgrids of the MAC grid are defined node-wise (see Fig. 2). That is, velocity grid cells are defined to have velocity degrees of freedom on their vertices, as opposed to on their centers as with the pressure grid. We will later duplicate all interface pressure and velocity cells to introduce virtual degrees of freedom, however first we must define the interface velocity cells in a manner consistent with the definition of interface pressure cells. This is done by first creating a single segment approximation to the zero isocontour of the level set over each pressure grid cell. Using linear interpolation, we determine the approximate intersection of the level set isocontour at each edge of each interface pressure cell. If this intersection is too close to a u

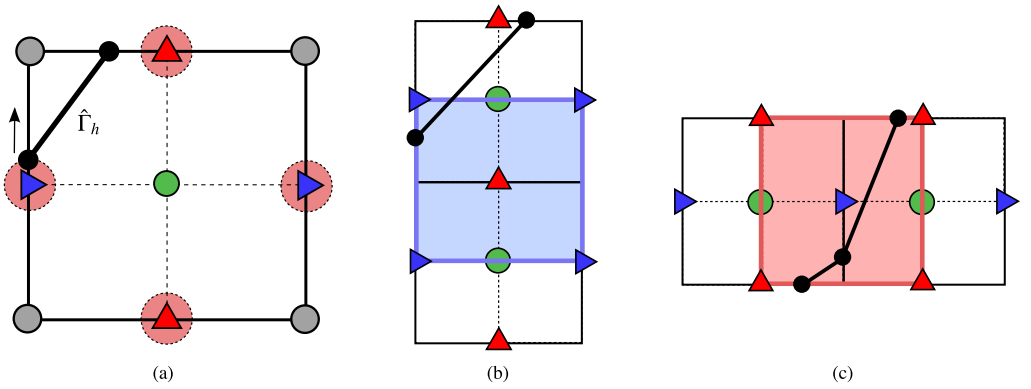

Fig. 8. Determination of cut u and v cells from $\hat{\Gamma}_h$: velocity cells are defined to be interface cells if either of the two incident pressure cells have a segment in $\hat{\Gamma}_h$ that intersects the cell. Figure (a) shows a typical case of interface perturbation to prevent u and v nodes from intersecting the segments. The dashed red circles indicate the threshold distance from the interface to a given u or v node. The upward arrow indicates the perturbation of the intersection point away from that u node. Figure (b) shows a u cell and the two p cells it intersects. This u cell is considered cut since it is intersected by the interface segment at one of these pressure cells. Figure (c) shows a similar example for a cut cell on the v grid. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

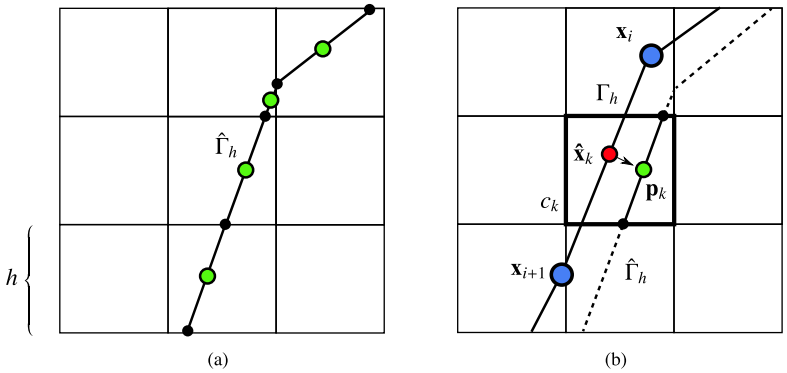

Fig. 9. Setting the jump conditions for the stress tensor components. (a) The interfacial forces (Eq. (5)) must be defined at the centers of the segments on the pressure-cell-wise Lagrangian approximation to the zero isocontour of the level set $\hat{\Gamma}_h$. These cell-wise segment centers are shown in light-green large circles above over each pressure cell. The black smaller circles represent points were the level set interpolates to zero for the pressure cut cells. (b) Jump conditions are naturally computed at the nodes $\mathbf{x}_i$ of the discrete interface Γ_h and then transferred to the points $\mathbf{p}_k$ through the points $\hat{\mathbf{x}}_k$. Each point $\mathbf{p}_k$ lies at the center of the $\hat{\Gamma}_h$ segment on the pressure cell c_k. The force at $\hat{\mathbf{x}}_k$ is computed by linearly interpolating the force values at the nodes $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. Note that the above separation between the level set generated interface $\hat{\Gamma}_h$ and the discrete interface Γ_h is exaggerated for illustrative purposes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

or v node, we perturb it slightly towards the end node of that cell edge lying closer to the intersection point (see Fig. 8). This modification is enough to move the intersection away from the relevant u or v node. Typically we perturb the position of the intersection point by αh with $\alpha = 10^{-6}$. This perturbation is performed because our discretization of the u and v fluid equations will require that this interface does not cross the velocity grid nodes (see Section 3.2.1.2). Connecting the segments in each pressure cell together forms a closed curve that we denote as $\hat{\Gamma}_h$ (see Fig. 9). We use this curve to define which cells in the u and v grids intersect the interface. Each velocity grid cell intersects two pressure cells. A u or v velocity cell is defined to intersect the interface whenever either of its two incident pressure cells has an interface segment that intersects the velocity cell (see Fig. 8).

3.1.3. Transfer of the Lagrangian interfacial forces to the Eulerian fluid grid

We require a description of the interface forces $\mathbf{f}^i$ along the embedded interface $\hat{\Gamma}_h$ when discretizing the Eulerian fluid variables (see Section 3.2.1.2). It suffices to define values for $\mathbf{f}^i$ at the center of each segment of $\hat{\Gamma}_h$ (see Fig. 9(a)). Unfortunately, the interface forces $\mathbf{f}^i$ are naturally defined at the nodes of the original Lagrangian interface Γ_h. We provide a procedure to transfer these conditions from the Lagrangian nodes of Γ_h to the centers of the segments on $\hat{\Gamma}_h$ (see Fig. 9).

Recall that we denote the points on the curve Γ_h as $\mathbf{x}_i$ with $i = 0, 1, \ldots, M-1$. We denote the number of pressure cells that intersect the interface as n_q. Let $\mathcal{P} = \left\{\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{n_q-1}\right\}$ denote the set of segment centers on $\hat{\Gamma}_h$ (see Fig. 9(b)). Assume we have computed the forces $\mathbf{f}^i$ at the points $\mathbf{x}_i$ of Γ_h. For each $k = 0, 1, \ldots, n_q - 1$, let $\hat{\mathbf{x}}_k$ be the point on Γ_h closest to $\mathbf{p}_k$ (see Fig. 9). We linearly interpolate $\mathbf{f}^i$ at $\hat{\mathbf{x}}_k$ from the values at the ends of the segment to which it belongs. This interpolated value is then defined to be the interfacial force at $\mathbf{p}_k$.

3.2. Eulerian discretization details

We will now describe the discretization of the Eulerian fluid variables over the MAC grid. We will first discuss the general case of discontinuous viscosity. We then give details for the special case of continuous viscosity. Finally, we describe the case of embedded Dirichlet velocity boundary conditions on irregular domains.

3.2.1. Variational formulation: discontinuous viscosity

In the case of discontinuous viscosity, our discrete stencils for the velocity and fluid variables are defined from the weak formulation of the Stokes equations. Our method is somewhat similar to X-FEM in this regard. We discuss such similarities explicitly in [1]. We will first derive this weak form including the effects of the discontinuity in the fluid stresses arising from the interfacial forces. We use a cell duplication procedure to introduce virtual nodes that accurately capture the discontinuities in the stress. However, this process requires a Lagrange multiplier term to guarantee continuity in the fluid velocity across the interface. Our discretization procedure is designed to exactly capture the null modes in the variational formulation. In the case of periodic Ω or Dirichlet velocity on $\partial\Omega$, the pressure can only be determined up to a constant, and the Lagrange multipliers that guarantee continuity can only be determined up to the same constant times the x and y components of the interface normal. We show that failure to discretely capture these modes exactly leads to significantly inferior performance. Furthermore, we show that our definition of the Eulerian interface $\hat{\Gamma}_h$ is designed to facilitate resolution of these modes. For simplicity of exposition, we assume that the domain Ω is rectangular and periodic in this section (we con-

sider the case of general irregular domains in Section 3.2.3). Consider the integral of the dot product of the force balance portion of the Stokes interface problem (Eq. (1)) with a function $\mathbf{w}(\mathbf{x}) \in \mathbf{H}^1(\Omega \setminus \Gamma) \times \mathbf{H}^1(\Omega \setminus \Gamma)$:

$$-\int_{\Omega\setminus\Gamma} \mathbf{w}\cdot(\nabla\cdot\sigma)\,dA = -\int_{\Omega\setminus\Gamma} \nabla\cdot(\mathbf{w}\cdot\sigma) - \nabla\mathbf{w}:\sigma\,dA = \int_{\Gamma}[\mathbf{w}\cdot\sigma]\cdot\mathbf{n}\,dl + \int_{\Omega\setminus\Gamma}\nabla\mathbf{w}:\sigma\,dA = \int_{\Omega\setminus\Gamma}\mathbf{w}\cdot\mathbf{f}\,dA, \tag{6}$$

where the divergence theorem was used with $\mathbf{n}$ being the outward unit normal to Ω^- at a point $\mathbf{x} \in \Gamma$ as in Section 1. Using the equalities

$$[ab] = a^+b^+ - a^-b^- = [a]\overline{b} + \overline{a}[b], \quad \overline{a} = \frac{a^+ + a^-}{2} \quad \text{and} \quad \overline{b} = \frac{b^+ + b^-}{2} \tag{7}$$

and the symmetry of σ, we can rewrite these equations as:

$$\begin{aligned}
-\int_{\Omega\setminus\Gamma} \mathbf{w}\cdot(\nabla\cdot\sigma)\,dA &= \int_{\Gamma}[\mathbf{w}]\cdot\overline{\sigma}\cdot\mathbf{n}\,dl + \int_{\Gamma}\overline{\mathbf{w}}\cdot[\sigma]\cdot\mathbf{n}\,dl + \int_{\Omega\setminus\Gamma}\nabla\mathbf{w}:\sigma\,dA \\
&= \int_{\Omega\setminus\Gamma}\nabla\mathbf{w}:\left(\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^T) - p\mathbf{I}\right)dA + \int_{\Gamma}[\mathbf{w}]\cdot\overline{\sigma}\cdot\mathbf{n}\,dl + \int_{\Gamma}\overline{\mathbf{w}}\cdot[\sigma]\cdot\mathbf{n}\,dl \\
&= \int_{\Omega\setminus\Gamma}\frac{\mu}{2}(\nabla\mathbf{w}+\nabla\mathbf{w}^T):(\nabla\mathbf{u}+\nabla\mathbf{u}^T)\,dA - \int_{\Omega\setminus\Gamma}p(\nabla\cdot\mathbf{w})\,dA + \int_{\Gamma}[\mathbf{w}]\cdot\mathbf{q}\,dl + \int_{\Gamma}\overline{\mathbf{w}}\cdot\mathbf{f}^i\,dl \\
&= \int_{\Omega\setminus\Gamma}\mathbf{w}\cdot\mathbf{f}\,dA,
\end{aligned} \tag{8}$$

where $\mathbf{q}(\mathbf{x}) = (q^1, q^2)^T = \overline{\sigma}\cdot\mathbf{n}$ is defined only along Γ. Therefore, we can express the weak form of the Stokes interface problem as:

$$\begin{aligned}
&\text{Find } (\mathbf{u}, p, \mathbf{q}) \in \mathbf{H}^1(\Omega\setminus\Gamma)\times\mathbf{H}^1(\Omega\setminus\Gamma)\times\mathbf{L}^2(\Omega\setminus\Gamma)\times\mathbf{L}^2(\partial\Omega)\times\mathbf{L}^2(\partial\Omega) \quad \text{such that} \quad \int_{\Omega\setminus\Gamma}\frac{\mu}{2}(\nabla\mathbf{w}+\nabla\mathbf{w}^T) \\
&\quad :(\nabla\mathbf{u}+\nabla\mathbf{u}^T)\,dA - \int_{\Omega\setminus\Gamma}p(\nabla\cdot\mathbf{w})\,dA + \int_{\Gamma}[\mathbf{w}]\cdot\mathbf{q}\,dl = -\int_{\Gamma}\overline{\mathbf{w}}\cdot\mathbf{f}^i\,dl + \int_{\Omega\setminus\Gamma}\mathbf{w}\cdot\mathbf{f}\,dA, \quad \forall\mathbf{w} \\
&\quad \in \mathbf{H}^1(\Omega\setminus\Gamma)\times\mathbf{H}^1(\Omega\setminus\Gamma),
\end{aligned} \tag{9}$$

$$\int_{\Omega\setminus\Gamma}\lambda\nabla\cdot\mathbf{u}\,dA = 0, \quad \forall\lambda \in \mathbf{L}^2(\Omega\setminus\Gamma), \tag{10}$$

$$\int_{\Gamma}\boldsymbol{\psi}(\mathbf{x})\cdot[\mathbf{u}(\mathbf{x})]\,dl = \mathbf{0}, \quad \forall\boldsymbol{\psi} \in \mathbf{L}^2(\Gamma)\times\mathbf{L}^2(\Gamma). \tag{11}$$

Note that we must treat $\mathbf{q}$ as an additional unknown since although we could compute $\overline{\sigma}\cdot\mathbf{n}$ directly from $\mathbf{u}$, it will not generally be known and must be solved for simultaneously with $\mathbf{u}$. This can be viewed as the Lagrange multiplier associated with the constraint that the velocity must be continuous (Eq. (11)).

3.2.1.1. Null modes. The null modes of the weak formulation play an important role in our discretization. Specifically, we found that failure to exactly resolve discrete counterparts to the continuous null modes associated with constant pressure leads to drastically inferior performance. This is not uncommon for simple discrete systems like the Poisson equation with periodic or Neumann boundary conditions in which the rows of the discrete system must sum to zero. In the case of the weak form of the periodic interfacial Stokes problem, there are three null modes. The first two modes are simply constant x or constant y velocities (with pressures p and Lagrange multipliers $\mathbf{q}$ equal to zero). The third mode is a constant pressure mode with Lagrange multipliers equal to the constant pressure times the x or y component of normal at the interfacial discontinuity (and with zero velocities). This is also a null mode in the case of an interface problem with Dirichlet velocity boundary conditions rather than periodic conditions since this mode has no velocity component. We can derive these modes by first noting that all integral operators on the left of Eqs. (9)–(11) are bilinear in $\mathbf{u}$, p, $\mathbf{q}$ and $\mathbf{w}$, λ, $\boldsymbol{\psi}$. Hence, with some abuse of notation, we can write the left hand side of these equations as:

$$(\mathbf{w},\ \lambda,\ \boldsymbol{\psi})\mathcal{K}\begin{pmatrix}\mathbf{u}\\ p\\ \mathbf{q}\end{pmatrix} = (\mathbf{w},\ \lambda,\ \boldsymbol{\psi})\begin{pmatrix}\mathcal{A} & -\mathcal{G} & \mathcal{C}\\ -\mathcal{D} & 0 & 0\\ \mathcal{B} & 0 & 0\end{pmatrix}\begin{pmatrix}\mathbf{u}\\ p\\ \mathbf{q}\end{pmatrix}, \tag{12}$$

where:

$$\mathbf{w}\mathcal{A}\mathbf{u} = \int_{\Omega\setminus\Gamma}\frac{\mu}{2}(\nabla\mathbf{w}+\nabla\mathbf{w}^T):(\nabla\mathbf{u}+\nabla\mathbf{u}^T)\,dA, \quad \mathbf{w}\mathcal{G}p = \int_{\Omega\setminus\Gamma}p(\nabla\cdot\mathbf{w})\,dA, \quad \mathbf{w}\mathcal{C}\mathbf{q} = \int_{\Gamma}[\mathbf{w}]\cdot\mathbf{q}\,dl,$$

$$\lambda\mathcal{D}\mathbf{u} = \int_{\Omega\setminus\Gamma}\lambda\nabla\cdot\mathbf{u}\,dA = 0, \quad \boldsymbol{\psi}\mathcal{B}\mathbf{u} = \int_{\Gamma}\boldsymbol{\psi}(\mathbf{x})\cdot[\mathbf{u}(\mathbf{x})]\,dl = \mathbf{0}.$$

Setting $\mathbf{u} = (c^1, c^2)^T$ with $p = 0$ and $\mathbf{q} = \mathbf{0}$ for arbitrary constants c^1 and c^2, it is trivial to see that $\mathbf{w}\mathcal{A}\mathbf{u} = 0$ since $\nabla(c^1, c^2)^T = \mathbf{0}$, $\lambda\mathcal{D}\mathbf{u} = 0$ since $\nabla \cdot (c^1, c^2)^T = 0$ and $\psi\mathcal{B}\mathbf{u} = 0$ since $\mathbf{u} = (c^1, c^2)^T$ is continuous across the interface. Therefore, $\mathbf{u} = (c^1, c^2)^T$ with $p = 0$ and $\mathbf{q} = \mathbf{0}$ are clearly two of the null modes. We can derive the third mode by setting $\mathbf{u} = \mathbf{0}$ with $p = c$ and applying the divergence theorem:

$$\int_{\Omega\setminus\Gamma} p(\nabla \cdot \mathbf{w})\, dA = c \int_{\Omega\setminus\Gamma} (\nabla \cdot \mathbf{w})\, dA = -c \int_{\Gamma} [\mathbf{w}] \cdot \mathbf{n}\, dl = \int_{\Gamma} [\mathbf{w}] \cdot \mathbf{q}\, dl. \tag{13}$$

Therefore in order for $-\mathcal{G}p + C\mathbf{q} = \mathbf{0}$, we must have $\mathbf{q} = -c\mathbf{n}$. We will discuss the discrete versions of these kernel modes in Section 3.2.1.4.

3.2.1.2. Discretization of fluid variables. We duplicate all u, v and p interface grid cells to introduce virtual nodes that capture the discontinuities in the fluid stress. That is, for each interface cell, we create a positive and negative version of the cell associated with Ω_h^+ and Ω_h^- respectively. These new cells introduce four new degrees of freedom for the u and v grids and one new degree of freedom for the p grid. We refer to these newly introduced degrees of freedom as "virtual". For the u and v grids, the virtual degrees of freedom are on the vertices of the newly created Ω_h^+ and Ω_h^- cells that lie outside the respective sub-

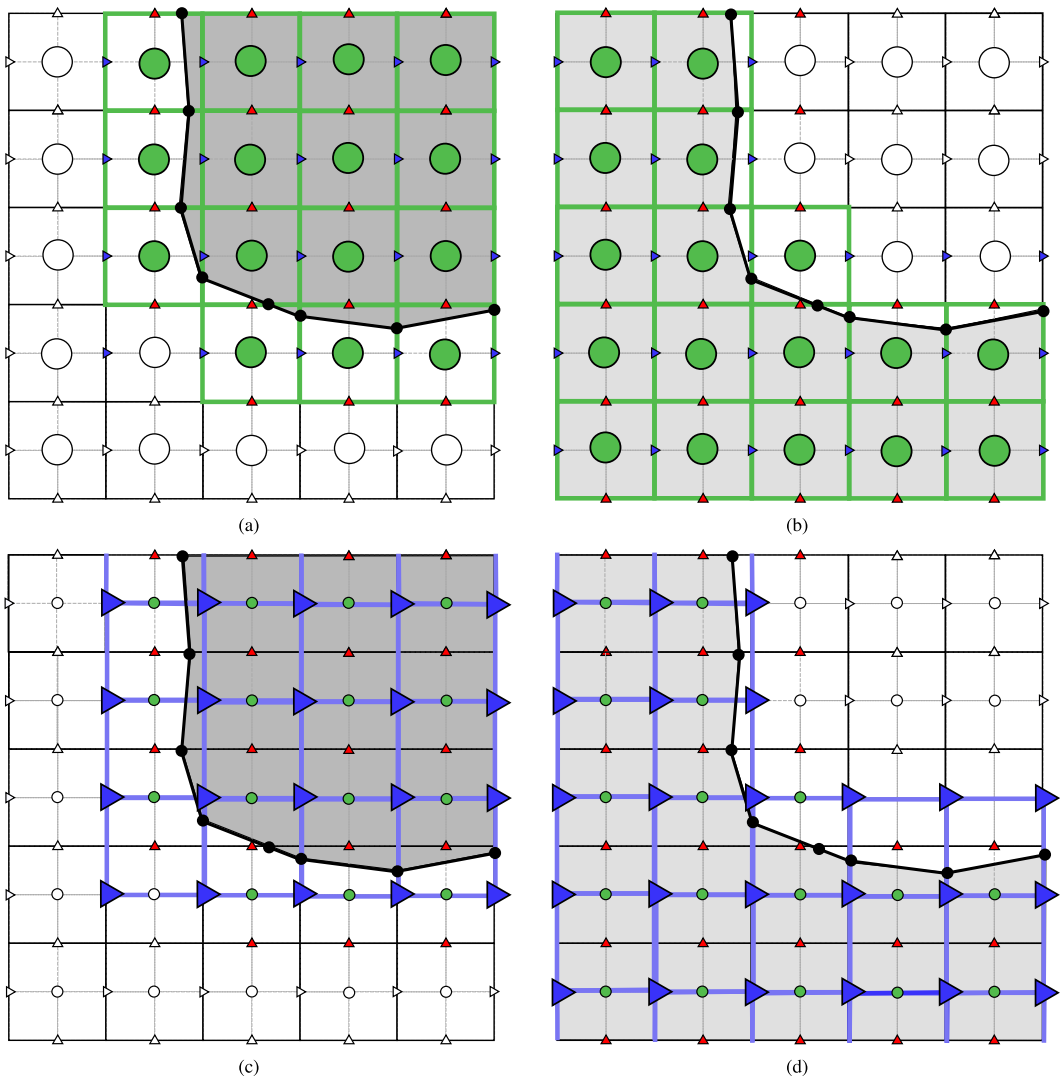

Fig. 10. All interface velocity and pressure cells are duplicated to introduce virtual degrees of freedom for capturing discontinuities. Figure (a) shows the material and virtual nodes on the pressure grid associated with Ω_h^- while Figure (b) shows the material and virtual nodes of that grid associated with Ω_h^+. The virtual nodes associated with Ω_h^- are the duplicated versions of the material nodes associated with Ω_h^+ which lie at the same positions. Figures (c) and (d) show the nodes of the u grid associated with Ω_h^- and Ω_h^+ respectively.

domains. There is only one virtual pressure degree of freedom since the p degrees of freedom are on cell centers. This process is illustrated in Fig. 10.

This duplication process provides a classification of all grid cells as either interior (Ω_h^-) or exterior (Ω_h^+). Furthermore, all degrees of freedom can also be classified as either interior or exterior. We use this classification to define interior and exterior piecewise bilinear approximations to u and v and interior and exterior piecewise constant approximations to p. Specifically, we assume u and v are bilinear over interior and exterior cells, and we assume p is constant over each exterior and interior cell. Note that our duplication of grid cells at the interface means that these fields are discontinuous at the interface. We can describe these piecewise bilinear fields as

$$u_h(\mathbf{x}) = \sum_{i=1}^{n_u} u_i N_i^u(\mathbf{x}) \quad \text{and} \quad v_h(\mathbf{x}) = \sum_{j=1}^{n_v} v_j N_j^v(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \Omega_h, \tag{14}$$

where n_u and n_v are the number of x and y nodes, respectively, and N_i^u and N_j^v are the standard piecewise bilinear interpolating functions associated with x node i and y node j, respectively. We will henceforth use $\mathbf{U} \in \mathbb{R}^{n_u+n_v}$ to denote the vector of velocity degrees of freedom. Similarly, the pressure field is described as

$$p_h(\mathbf{x}) = \sum_{k=1}^{n_p} p_k \phi_k^p(\mathbf{x}), \quad \text{for all } \mathbf{x} \in \Omega_h \setminus \hat{\Gamma}_h, \tag{15}$$

where n_p is the number of cells in the duplicated p grid and ϕ_k^p is the characteristic function of the kth pressure cell (equal to one for $\mathbf{x}$ in the cell and zero otherwise). We will henceforth use $\mathbf{P} \in \mathbb{R}^{n_p}$ to denote the vector of pressure degrees of freedom. Note that each of the functions $N_i^u(\mathbf{x}), N_i^v(\mathbf{x})$ and $\phi_i^p(\mathbf{x})$ are defined on one side of the interface. Hence, if $\mathbf{x} \in \Omega_h^+$, then $N_i^u(\mathbf{x}) = 0$ for all the nodes i on the u grid associated with Ω_h^- and vice versa. The same applies to the basis functions on the v and p grids.

Since $\mathbf{q}$ is defined only along Γ, we build its discrete counterpart only along $\hat{\Gamma}_h$. Specifically, we assume that $\mathbf{q}(\mathbf{x})$ is piecewise constant over each segment in $\hat{\Gamma}_h$. It is therefore piecewise constant over each (unduplicated) pressure cell on the interface. If there are n_q pressure cells cut by $\hat{\Gamma}_h$, then:

$$\mathbf{q}_h(\mathbf{x}) = \sum_{l=1}^{n_q} \mathbf{q}_l \phi_l^q(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \hat{\Gamma}_h. \tag{16}$$

$\phi_l^q(\mathbf{x})$ is the characteristic function of the lth pressure cell that is cut by the interface. We will henceforth use $\mathbf{Q} = \begin{pmatrix} \mathbf{Q}^1 \\ \mathbf{Q}^2 \end{pmatrix} \in \mathbb{R}^{2n_q}$ to denote the vector of Largange multiplier degrees of freedom ($\mathbf{Q}^1$ and $\mathbf{Q}^2$ contain all x and y degrees of freedom of $\mathbf{q}_h$ respectively).

The body force density is also discretized by assuming it is piecewise constant over each pressure cell. Hence:

$$\mathbf{f}_h = \sum_{k=1}^{n_p} \bar{\mathbf{f}}_k \phi_k^p(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \Omega_h \setminus \hat{\Gamma}_h, \tag{17}$$

where $\bar{\mathbf{f}}_k$ is the average body force density on the kth pressure cell. In all examples considered, the body forces are spatially constant over each fluid domain and hence the cell averages are constants over Ω_h^+ and Ω_h^-. Similarly, as discussed in Section 3.1.3, we define the jump conditions in the stress to be piecewise constant over each segment in $\hat{\Gamma}_h$, so we can represent the discrete Eulerian stress jump $\mathbf{f}^i$ as

$$\mathbf{f}_h^i(\mathbf{x}) = \sum_{i=1}^{n_q} \mathbf{f}^i(\mathbf{p}_i) \phi_i^q(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \hat{\Gamma}_h. \tag{18}$$

Here, $\mathbf{f}^i(\mathbf{p}_i)$ are the values of the interfacial forces transferred from the discrete Lagrangian interface Γ_h to the segment centers $\mathbf{p}_i$ in the discrete Eulerian interface $\hat{\Gamma}_h$ as described on Section 3.1.3.

We now define our discretization of the Stokes interface problem (9)–(11) using the Eulerian approximations $u_h,\ v_h,\ p_h,\ \mathbf{q}_h,\ \mathbf{f}_h^i$ and $\mathbf{f}_h$ just discussed. That is, we approximate the space $\mathbf{H}^1(\Omega \setminus \Gamma) \times \mathbf{H}^1(\Omega \setminus \Gamma)$ by the space $V_h^u(\Omega_h) \times V_h^v(\Omega_h)$ of piecewise bilinear functions over the u and v grids, the space $L^2(\Omega_h)$ by the space $V_h^p(\Omega_h)$ of piecewise constant functions over each pressure grid cell and the space $L^2(\Gamma)$ by the space $V_h^q(\hat{\Gamma}_h)$ of piecewise constant functions over the intersection of $\hat{\Gamma}_h$ with each pressure cell. We obtain a linear system by assuming that $\mathbf{w}_h$ is approximated as with $\mathbf{u}_h$, that λ_h is approximated as with p_h and that $\boldsymbol{\psi}_h$ is approximated as with $\mathbf{q}_h$. Specifically, the discrete variational problem can be written as:

Find $\mathbf{U} \in \mathbb{R}^{n_u+n_v},\ \mathbf{P} \in \mathbb{R}^{n_p},\ \mathbf{Q} \in \mathbb{R}^{2n_q}$ such that

$$(\mathbf{W}^T, \boldsymbol{\Lambda}^T, \boldsymbol{\Psi}^T)\mathbf{K}\begin{pmatrix} \mathbf{U} \\ \mathbf{P} \\ \mathbf{Q} \end{pmatrix} = (\mathbf{W}^T, \boldsymbol{\Lambda}^T, \boldsymbol{\Psi}^T)\begin{pmatrix} \mathbf{F} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \quad \forall \begin{pmatrix} \mathbf{W} \\ \boldsymbol{\Lambda} \\ \boldsymbol{\Psi} \end{pmatrix} \in \mathbb{R}^{n_u+n_v+n_p+2n_q}, \tag{19}$$

where

$$\mathbf{K} = \begin{pmatrix} \mathbf{A}^{uu} & \mathbf{A}^{uv} & -\mathbf{G}^{u} & \mathbf{C}^{u} & \mathbf{0} \\ \mathbf{A}^{vu} & \mathbf{A}^{vv} & -\mathbf{G}^{v} & \mathbf{0} & \mathbf{C}^{v} \\ -\mathbf{D}^{u} & -\mathbf{D}^{v} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}^{u} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^{v} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad \text{and} \quad \mathbf{F} = \begin{pmatrix} \mathbf{F^1} \\ \mathbf{F}^2 \end{pmatrix} = \begin{pmatrix} \widehat{\mathbf{F}}^{\mathbf{1}} \\ \widehat{\mathbf{F}}^{\mathbf{2}} \end{pmatrix} + \begin{pmatrix} \hat{\mathbf{f}}^{\mathbf{1}} \\ \hat{\mathbf{f}}^{\mathbf{2}} \end{pmatrix}. \tag{20}$$

Here, $\mathbf{W} \in \mathbb{R}^{n_u+n_v}$, $\mathbf{\Lambda} \in \mathbb{R}^{n_p}$, and $\mathbf{\Psi} \in \mathbb{R}^{2n_q}$ are the discrete degrees of freedom representing $\mathbf{w}_h$, λ_h and $\boldsymbol{\psi}_h$, respectively, and $\mathbf{U} \in \mathbb{R}^{n_u+n_v}$, $\mathbf{P} \in \mathbb{R}^{n_p}$, and $\mathbf{Q} \in \mathbb{R}^{2n_q}$ are the discrete degrees of freedom representing $\mathbf{u}_h, p_h$ and $\mathbf{q}_h$, respectively. $\widehat{\mathbf{F}}^1 \in \mathbb{R}^{n_u}$ and $\widehat{\mathbf{F}}^2 \in \mathbb{R}^{n_v}$ are associated with the interfacial forces along the x and y directions respectively while $\hat{\mathbf{f}}^1 \in \mathbb{R}^{n_u}$ and $\hat{\mathbf{f}}^2 \in \mathbb{R}^{n_v}$ are associated with the body forces along the x and y directions respectively. We can derive the entries in the matrix $\mathbf{K}$ from Eq. (9) by assuming that only one entry in any of $\mathbf{W} \in \mathbb{R}^{n_u+n_v}$, $\mathbf{\Lambda} \in \mathbb{R}^{n_p}$, $\mathbf{\Psi} \in \mathbb{R}^{2n_q}$ is equal to one with all other entries zero.

The careful derivation of these equations is somewhat lengthy so we cover this in the Appendix A.

3.2.1.3. Computation of the matrix $\mathbf{K}$ *elements.* There are many area and line integrals involved in the definition of the discrete system. We will show here that they can all be computed with modest implementation complexity and computational cost. We perform all area integrals by dividing them into sums of integrals over cells in the pressure grid. For example, each area integral is represented as a sum of integrals over the material region of the spatially disjoint pressure cells (whose union is the entire fluid domain). This pressure-cell-wise view of the integration allows us to naturally evaluate the integrals needed for the velocity and pressure matrices near the geometrically elaborate interface. The equations associated with the continuous velocity constraint only involve line integrals and we will discuss them later. We use c_i^p to denote the ith pressure cell for $i = 1, 2, \ldots, n_p$. To further facilitate the computation of the integrals, we divide each pressure cell c_i^p into four subcells ω_1, ω_2, ω_3, and ω_4, as shown in Fig. 11(a). Also, for reasons which will be outlined below, we construct positively oriented boundaries for each of these subcells intersecting the material region as in Fig. 12. With this convention, the area integrals are expressed as

$$\int_{\Omega_h \setminus \hat{\Gamma}_h} f(x,y)\, dA = \int_{\Omega_h^+} f(x,y)\, dA + \int_{\Omega_h^-} f(x,y)\, dA = \sum_{i=1}^{n_p} \sum_{j=1}^{4} \int_{w_j \cap \Omega_h^+} f(x,y)\, dA + \sum_{i=1}^{n_p} \sum_{j=1}^{4} \int_{w_j \cap \Omega_h^-} f(x,y)\, dA. \tag{21}$$

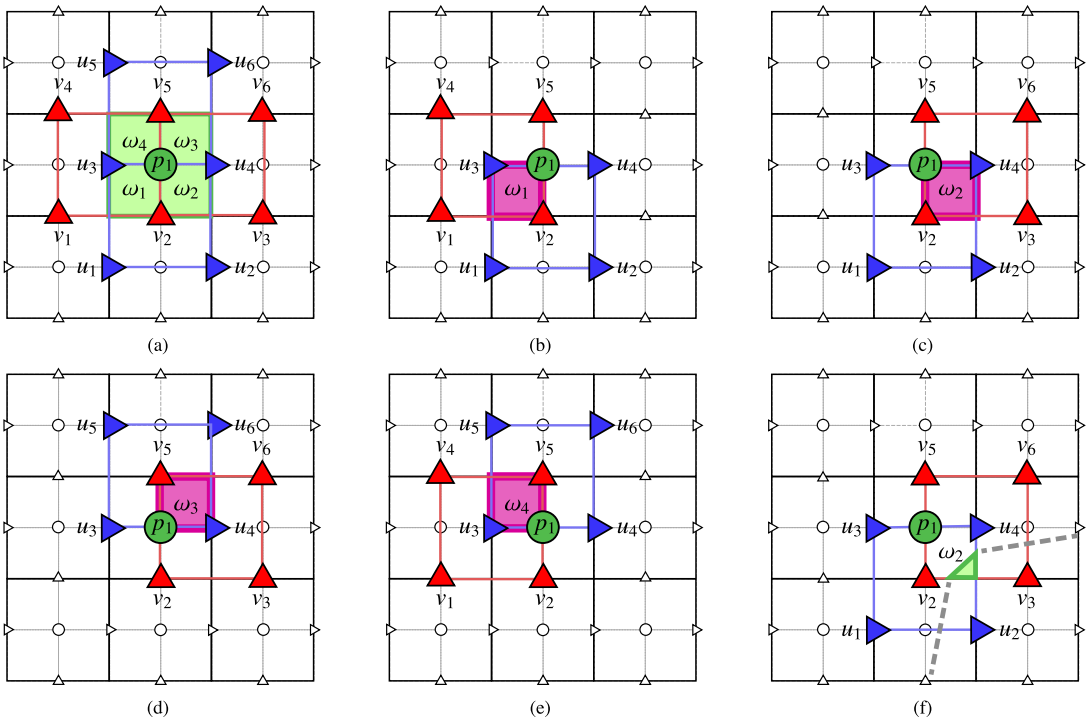

Fig. 11. Nodes involved in the contribution to the elements of $\mathbf{K}$ from a given (shaded) pressure cell. Figure (a) shows all contributing nodes on a given material pressure cell, while figures (b)–(e) show the contributing nodes for each subcell (marked in pink) lying on the interior of the pressure cell from Figure (a). Figure (f) shows an example in which the pressure cell is cut; in the example shown fewer u and v nodes are needed to compute the matrix elements associated with that cell in comparison to the cell of Figure (a) (in this case, only subcell ω_2 has a nonvanishing material region). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

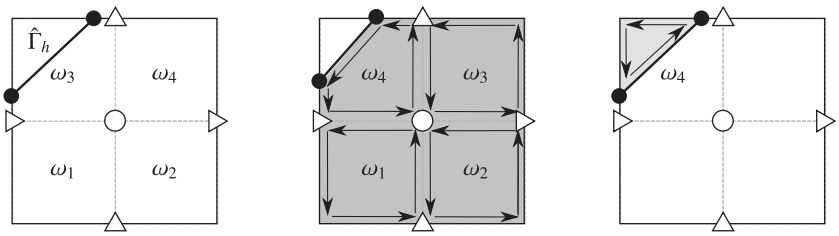

Fig. 12. Construction of the positively oriented boundaries of each subcell in a given pressure cell. The figure on the left shows the pressure cell and its four subcells ω_1, ω_2, ω_3 and ω_4. The subcells are duplicated since the original pressure cell is cut. The figure in the middle shows the orientation of the boundaries of the subcells associated with the copy of the pressure cell on Ω_h^-, while the figure on the right shows the same for the single subcell (ω_4) associated with the copy of the pressure cell on Ω_h^+.

Note that the second sum in each term of the right hand side is over the subcells ω_j of the pressure cell c_i^p and that some of these subregions may be empty in the case of cut cells, as outlined in Fig. 12.

We use our pressure-cell-wise integration convention to express the discrete system matrix $\mathbf{K}$ as the sum of the pressure-cell element matrices $\mathbf{K}^{c_i^p} \in \mathbb{R}^{n_i^p \times n_i^p}$ for cells c_i^p. Here we slightly abuse the term "sum" since n_i^p will be between 7 and 13. We simply mean that the entries in $\mathbf{K}$ can be written as sums of entries in these cell-wise matrices. The size of each matrix $\mathbf{K}^{c_i^p}$ is determined by the number of interpolating functions supported over pressure cell c_i^p. The staggering of variables leads to 13 interpolating functions supported over a given interior/uncut pressure cell (6 x-components, 6 y-components and one pressure). There may be fewer than 12 velocity nodes involved in the case of cut cells (see Fig. 11). The number is determined by the subregions ω_1–ω_4 that intersect the domain. This is illustrated in Fig. 11(c).

As mentioned above, we further divide the cell-wise $\mathbf{K}^{c_i^p}$ as a sum of matrices defined over the subregions $\omega_1 - \omega_4$ as $\mathbf{K}^{c^p} = \mathbf{K}^{c_i^p}_{\omega_1} + \mathbf{K}^{c_i^p}_{\omega_2} + \mathbf{K}^{c_i^p}_{\omega_3} + \mathbf{K}^{c_i^p}_{\omega_4}$. This is done because the integrands are all smooth over the ω_1–ω_4 subregions (with kinks at the boundaries of these regions). Specifically, the integrands are at most quadratic over the subregions and we preform these integrations analytically following the ideas developed in [1]. For example, the entries in $\mathbf{K}^{c_i^p}_{\omega_1}$ involve u_1, u_2, u_3, u_4, v_1, v_2, v_4 and v_5 as demonstrated in Fig. 11(b); therefore, it only has non-zero values on rows and columns involving these degrees of freedom. The resulting equations based on those degrees of freedom are shown in Fig. 13. If we order the 13 nodes with indices shown in Fig. 11(a), then on the interior of the domain, where $c_i^p \cap \Omega_h = c_i^p$, the sum of these four subintegrals is always the same:

$$\mathbf{K}^{c_i^p}_{\omega_1} = \frac{1}{64}\begin{pmatrix}
16\mu & 0 & 0 & -16\mu & 0 & 0 & 9\mu & -6\mu & -3\mu & 3\mu & -2\mu & -\mu & 8h \\
0 & 16\mu & -16\mu & 0 & 0 & 0 & 3\mu & 6\mu & -9\mu & \mu & 2\mu & -3\mu & -8h \\
0 & -16\mu & 96\mu & -\mu & 0 & -16\mu & -6\mu & 4\mu & 2\mu & 6\mu & -4\mu & -2\mu & 48h \\
-16\mu & 0 & -\mu & 96\mu & -16\mu & 0 & -2\mu & -4\mu & 6\mu & 2\mu & 4\mu & -6\mu & -48h \\
0 & 0 & 0 & -16\mu & 16\mu & 0 & -3\mu & 2\mu & \mu & -9\mu & 6\mu & 3\mu & 8h \\
0 & 0 & -16\mu & 0 & 0 & 16\mu & -\mu & -2\mu & 3\mu & -3\mu & -6\mu & 9\mu & -8h \\
9\mu & 3\mu & -6\mu & -2\mu & -3\mu & -\mu & 16\mu & 0 & 0 & 0 & -16\mu & 0 & 8h \\
-6\mu & 6\mu & 4\mu & -4\mu & 2\mu & -2\mu & 0 & 96\mu & 0 & -16\mu & -\mu & -16\mu & 48h \\
-3\mu & -9\mu & 2\mu & 6\mu & \mu & 3\mu & 0 & 0 & 16\mu & 0 & -16\mu & 0 & 8h \\
3\mu & \mu & 6\mu & 2\mu & -9\mu & -3\mu & 0 & -16\mu & 0 & 16\mu & 0 & 0 & -8h \\
-2\mu & 2\mu & -4\mu & 4\mu & 6\mu & -6\mu & -16\mu & -\mu & -16\mu & 0 & 96\mu & 0 & -48h \\
-\mu & -3\mu & -2\mu & -6\mu & 3\mu & 9\mu & 0 & -16\mu & 0 & 0 & 0 & 16\mu & -8h \\
8h & -8h & 48h & -48h & 8h & -8h & 8h & 48h & 8h & -8h & -48h & -8h & 0
\end{pmatrix}. \tag{22}$$

We illustrate the stencil sparsity in Fig. 16 for equations sufficiently far from the interface.

For boundary cells where $c^p \cap \Omega_h \neq c^p$, we have to perform the integrations involved in each of $\mathbf{K}^{c_i^p}_{\omega_j}$ carefully, taking into account the boundary geometry. These integrals can be computed analytically in a straightforward manner following the approach of [1]. Let $\mathcal{P}$ be the polygonal cut-cell geometry consisting of d boundary segments s_i. All pressure subcells cells fit into this category since their boundaries are a set of segments which we orient as in Fig. 12. Let (x_i, y_i) and $(x_i + \Delta x_i, y_i + \Delta y_i) = (x_{i+1}, y_{i+1})$ be the coordinates of the end points of s_i. If $p_2(x,y) = a + bx + cy + dx^2 + exy + fy^2$ is a quadratic function of x and y, then it can be shown that:

i/j	$u_1 \quad u_2 \quad u_3 \quad u_4$	$v_1 \quad v_2 \quad v_3 \quad v_4$	p_1
u_1 u_2 u_3 u_4	$\mu \int_{\omega_1} 2N^u_{i,x}N^u_{j,x} + N^u_{i,y}N^u_{j,y}dA$	$\mu \int_{\omega_1} N^u_{i,y}N^v_{j,x}dA$	$-\int_{\omega_1} N^u_{i,x}dA$
v_1 v_2 v_3 v_4	$\mu \int_{\omega_1} N^v_{i,x}N^u_{j,y}dA$	$\mu \int_{\omega_1} 2N^v_{i,y}N^v_{j,y} + N^v_{i,x}N^v_{j,x}dA$	$-\int_{\omega_1} N^v_{i,y}dA$
p_1	$-\int_{\omega_1} N^u_{j,x}dA$	$-\int_{\omega_1} N^v_{j,y}dA$	0

Fig. 13. Symbolic formulas for the entries in the pressure-cell-wise discretization: $K^{c^p}_{w_1}$.

$$\begin{aligned}\int_{\mathcal{P}} p_2(x,y)\,dA = \sum_{i=0}^{d-1}\Delta y_i\Bigg[a\left(\frac{\Delta x_i}{2}+x_i\right)+b\left(\frac{(\Delta x_i)^2}{6}+\frac{(\Delta x_i)x_i}{2}+\frac{x_i^2}{2}\right)+c\left(\frac{(\Delta x_i)(\Delta y_i)}{3}+\frac{(\Delta x_i)y_i+(\Delta y_i)x_i}{2}+x_iy_i\right)\\ +d\left(\frac{(\Delta x_i)^3}{12}+\frac{x_i(\Delta x_i)^2}{3}+\frac{x_i^2(\Delta x_i)}{2}+\frac{x_i^3}{3}\right)+e\left(\frac{(\Delta x_i)^2(\Delta y_i)}{8}+\frac{(\Delta x_i)(\Delta y_i)x_i}{3}+\frac{x_i^2(\Delta y_i)}{4}+\frac{(\Delta x_i)^2y_i}{6}+\frac{(\Delta x_i)x_iy_i}{2}+\frac{x_i^2y_i}{2}\right)\\ +f\left(\frac{(\Delta x_i)(\Delta y_i)^2}{4}+\frac{2(\Delta x_i)(\Delta y_i)y_i}{3}+\frac{(\Delta x_i)y_i^2}{2}+\frac{x_i(\Delta y_i)^2}{3}+(\Delta y_i)y_ix_i+x_iy_i^2\right)\Bigg].\end{aligned} \tag{23}$$

All functions that need to be integrated when computing the elements of **K** are at most quadratic over the pressure subcells. We use the procedure above to compute these integrals. Eq. (23) is derived as in [1] by applying the divergence theorem:

$$\begin{aligned}\int_{\mathcal{P}} p_2(x,y)\,dA &= \int_{\mathcal{P}}\left(a+bx+cy+dx^2+exy+fy^2\right)dA = \int_{\mathcal{P}}\nabla\cdot\left(ax+\frac{bx^2}{2}+cxy+\frac{dx^3}{3}+\frac{ex^2y}{2}+fxy^2,0\right)dA\\ &= \sum_{i=1}^{d}\int_{s_i}\left(ax+\frac{bx^2}{2}+cxy+\frac{dx^3}{3}+\frac{ex^2y}{2}+fxy^2,0\right)\cdot\tilde{\mathbf{n}}_i dl\\ &= \sum_{i=1}^{d}\int_{s_i}\left(ax+\frac{bx^2}{2}+cxy+\frac{dx^3}{3}+\frac{ex^2y}{2}+fxy^2\right)\tilde{n}_i^1 dl,\end{aligned} \tag{24}$$

where $\tilde{n}_i^1$ is the x component of the normal $\tilde{\mathbf{n}}_i$ to the segment s_i. The computation of the line integrals is simplified if we parameterize each segment s_i as $\mathbf{p}(s) = (x_i + s\Delta x_i, y_i + s\Delta y_i)$ for $s \in [0,1]$.

Line integrals of quadratic polynomials $p_2(x,y)$ over the oriented polygonal boundaries are also needed for the right-hand-side terms and for the entries in the jump constraint matrix. These can also be computed analytically. For a polygonal curve $\mathcal{S}$ with d segments, we get:

$$\begin{aligned}\int_{\mathcal{S}} p_2(x,y)dl = \sum_{i=1}^{d} l_i\Bigg[a+b\left(x_i+\frac{\Delta x_i}{2}\right)+c\left(y_i+\frac{\Delta y_i}{2}\right)+d\left(x_i^2+x_i(\Delta x_i)+\frac{(\Delta x_i)^2}{3}\right)+e\left(x_iy_i+\frac{x_i(\Delta y_i)}{2}+\frac{y_i(\Delta x_i)}{2}+\frac{(\Delta x_i)(\Delta y_i)}{3}\right)\\ +f\left(y_i^2+y_i(\Delta y_i)+\frac{(\Delta y_i)^2}{3}\right)\Bigg],\end{aligned} \tag{25}$$

where $l_i = \sqrt{\Delta x_i^2+\Delta y_i^2}$ is the length of the ith segment of $\mathcal{S}$.

3.2.1.4. Kernel modes of the matrix **K**. Here we show that our discretization choices allow us to capture exactly the constant pressure null modes in our formulation. Failure to discretely resolve these modes resulted in significantly inferior performance of the method. Specifically, failure to discretize the interface geometry in a manner consistent with the discrete interface jump conditions leads to a matrix **K** that does not capture the constant pressure mode. The smallest mode of **K** is then numerically similar to the constant pressure mode, but with non-zero eigenvalue. This numerical error in the rank of the

matrix causes considerable degradation in the performance of the iterative solver for the discrete systems. We demonstrate this with explicit examples in Section 4.6.

First, in the case of periodicity, there are two modes that correspond to constant x and y velocities. These modes will always be captured when the rows of $\mathbf{K}$ associated with x and y velocity equations sum to zero. This constraint is satisfied for nearly all choices of interface geometry and jump condition discretization. As discussed in Section 3.2.1.1, the third mode (which is also present for Dirichlet velocity conditions) has zero velocity, constant pressure and Lagrange multiplier $\mathbf{q}$ equal to the constant pressure times the outward normal to the interface. Our choice of interface geometry discretization yields a constant normal to the interface on each pressure cell. We also set the Lagrange multiplier space to be piecewise constant wherever the pressure and the normals are piecewise constant. This combination of choices allows us to have a discrete mode that corresponds exactly to the continuous mode over a piecewise linear interface. Using the notation for our discrete variables, this mode has zero velocities $\mathbf{U}$ with any scalar multiple of $\mathbf{P} = (1, \ldots, 1)^T$ and the same scalar multiple of $q_i^l = -n_i^l$ for $l = 1, 2$ for each entry in $\mathbf{Q}$. Here, $\mathbf{n}_i = (n_i^1, n_i^2)$ is the ith outward normal of the discrete interface $\hat{\Gamma}_h$. As discussed in Section 3.2.1.2, we have a different value of the index i for each cut pressure cell. To verify that this is a kernel mode of $\mathbf{K}$, note that:

$$\mathbf{K}\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{P} \\ \mathbf{Q}^1 \\ \mathbf{Q}^2 \end{pmatrix} = \begin{pmatrix} -(\mathbf{D}^u)^T\mathbf{P} + (\mathbf{B}^u)^T\mathbf{Q}^1 \\ -(\mathbf{D}^v)^T\mathbf{P} + (\mathbf{B}^u)^T\mathbf{Q}^2 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}. \tag{26}$$

Since the velocities are zero, all $\mathbf{q}$ and p equations will be zero. Now, consider the ith x velocity equation. This is the equation associated with x velocity node $i = 1, 2, \ldots, n_u$. Assume that the node is associated with Ω_h^+. Then, if $\mathbf{P} = (1, \ldots, 1)^T$,

$$-\left((\mathbf{D}^u)^T\mathbf{P}\right)_i = -\sum_{j=1}^{n_p} D_{ji}^u p_j = -\sum_{j=1}^{n_p} \int_{\Omega_h \setminus \hat{\Gamma}_h} \phi_j^p N_{i,x}^u \, dA = -\int_{\Omega_h^+} \nabla \cdot (N_i^u, 0) \, dA = \int_{\hat{\Gamma}_h} N_i^u n^1 \, dl = \sum_{k=1}^{n_q} n_k^1 \int_{\hat{\Gamma}_h \cap c_{j(k)}^p} N_i^u \, dl. \tag{27}$$

Here we use the fact that the interpolating function N_i^u is only supported on Ω_h^+ since i is assumed to be a positive node. Also, we use the fact that n_k^1 is constant over $\hat{\Gamma}_h \cap c_{j(k)}^p$ where $c_{j(k)}^p$ is the pressure cell that intersects the kth segment in $\hat{\Gamma}_h$.

Now, if we set $q_k^1 = -n_k^1$ for $k = 1, 2, \ldots, n_q$ then,

$$\left((\mathbf{B}^u)^T\mathbf{Q}^1\right)_i = \sum_{k=1}^{n_q} B_{ki}^u q_k^1 = -\sum_{k=1}^{n_q} \Theta_i^u \int_{\hat{\Gamma}_h} N_i^u \phi_k^q n_k^1 \, dl = -\sum_{k=1}^{n_q} n_k^1 \int_{\hat{\Gamma}_h \cap c_{j(k)}^p} N_i^u \, dl \tag{28}$$

since i is a positive node implies that $\Theta_i^u = 1$. Therefore $-\left((\mathbf{D}^u)^T\mathbf{P}\right)_i + \left((\mathbf{B}^u)^T\mathbf{Q}^1\right)_i = 0$ for all i positive x node equations. The argument is simular for negative nodes and for the y equations. We reiterate that our choice of discretization for the Lagrangian multiplier $\mathbf{q}$, which is associated with the jump conditions for u and v, is crucial for the incorporation of the constant pressure mode into the kernel of the matrix $\mathbf{K}$. Generally speaking, the discrete $\mathbf{q}$ must be piecewise constant exactly where the discrete interface normals are piecewise constant. Other choices of discretization may prevent this mode from being captured exactly. An example of a discretization which fails to capture this mode is presented in Section 4.6. It is shown there that the simulation times are significantly larger when compared to our choice of discretization for $\mathbf{q}$.

Since the matrix $\mathbf{K}$ is symmetric, for any kernel mode $\mathbf{z}$ we have that $\mathbf{z}^T\mathbf{K} = (\mathbf{K}^T\mathbf{z})^T = \mathbf{0}$ because $\mathbf{K} = \mathbf{K}^T$. Hence, from Eq. (20), we see that the right hand side force terms must be orthogonal to the kernel of $\mathbf{K}$. The constant u, constant v and constant p modes described above provide an orthogonal basis of the kernel of $\mathbf{K}$ and hence we must always project out components on the right hand side forces which are parallel to the space spanned by these three vectors. Notice that projecting out the constant u and v vectors from the right hand side implies that the total force acting on the fluid in both the horizontal and vertical directions must have zero sum. The right hand side forces on Eq. (20) are naturally orthogonal to the constant pressure mode.

3.2.2. Continuous fluid viscosity

When the fluid viscosities are continuous ($\mu^+ = \mu^- = \mu$), we reduce the linear algebra demands by decoupling the velocity and pressure unknowns in the system into three Poisson interface problems (see [13,21,61]). We then use the methods in [1] to solve the Poisson interface problems. We also solve a fourth Poisson equation to enforce a discrete divergence-free condition on the velocity. This reformulation of the Stokes problem is only possible in the case of continuous viscosity. When the viscosity is discontinuous, there is no simple, decoupled formula for the individual velocity and pressure jump conditions in terms of the interfacial forces. Without knowledge of these jumps, it is not possible to decouple the system.

We derive the pressure equations by taking the divergence of both sides of Eq. (1) and noting that $\nabla \cdot \mathbf{u} = 0$ from incompressibility. We then get the following Poisson equation:

$$\nabla \cdot (\mu \Delta \mathbf{u} - \nabla p + \mathbf{f}) = 0 \;\Rightarrow\; \Delta p = \nabla \cdot \mathbf{f}, \quad \mathbf{x} \in \Omega \setminus \Gamma. \tag{29}$$

This divergence is rigorously defined for all points $\mathbf{x} \in \Omega$ on either side of the interface ($\mathbf{x} \in \Omega \setminus \Gamma = \Omega^- \cup \Omega^+$) but not for points $\mathbf{x}$ on the interface. After solving this Poisson equation for the pressure p, we can solve another two Poisson equations for the velocity components u and v using the solution for p:

$$\Delta p = \nabla \cdot \mathbf{f}, \quad \mathbf{x} \in \Omega \setminus \Gamma, \tag{30}$$

$$\mu \Delta u = p_x - f^1, \quad \mathbf{x} \in \Omega \setminus \Gamma, \tag{31}$$

$$\mu \Delta v = p_y - f^2, \quad \mathbf{x} \in \Omega \setminus \Gamma. \tag{32}$$

Here, f^1 and f^2 are the horizontal and vertical components of the body force density $\mathbf{f}$. These equations hold on the interior of the domain and away from the interface. We must therefore derive interface conditions for each Poisson interface problem in terms of the body forces $\mathbf{f}$ and the interfacial forces $\mathbf{f}^i$ to solve these equations in practice.

3.2.2.1. Interface conditions. Let $\mathbf{x}(s,t)$ be an arbitrary parametrization of the curve Γ. We can express the jump conditions in u, v and p in terms of the force per unit length of the parametrization parameter s. We will use $\mathbf{F}(s,t)$ to denote this parameterization dependent force density. The normal and tangential components of the force density $\mathbf{F}(\lambda,t)$ are then:

$$\begin{aligned} F_n(s,t) &:= \mathbf{F}(s,t) \cdot \mathbf{n}(s,t) \\ F_\tau(s,t) &:= \mathbf{F}(s,t) \cdot \boldsymbol{\tau}(s,t). \end{aligned} \tag{33}$$

Following the work of [13,21,61], we can use these conventions to express the jump conditions as:

$$[p](\mathbf{x}(s,t)) = F_n(s,t) \left\| \frac{\partial \mathbf{x}}{\partial s} \right\|^{-1}, \tag{34}$$

$$[u](\mathbf{x}(s,t)) = 0, \tag{35}$$

$$[v](\mathbf{x}(s,t)) = 0, \tag{36}$$

$$\left[\frac{\partial p}{\partial n} \right](\mathbf{x}(s,t)) = \frac{\partial}{\partial s} \left(F_\tau(s,t) \left\| \frac{\partial \mathbf{x}}{\partial s} \right\|^{-1} \right) \left\| \frac{\partial \mathbf{x}}{\partial s} \right\|^{-1} + [\mathbf{f}] \cdot \mathbf{n}, \tag{37}$$

$$\mu \left[\frac{\partial u}{\partial n} \right](\mathbf{x}(s,t)) = F_\tau(s,t) \sin\theta \left\| \frac{\partial \mathbf{x}}{\partial s} \right\|^{-1}, \tag{38}$$

$$\mu \left[\frac{\partial v}{\partial n} \right](\mathbf{x}(s,t)) = -F_\tau(s,t) \cos\theta \left\| \frac{\partial \mathbf{x}}{\partial s} \right\|^{-1}, \tag{39}$$

Despite the explicit appearance of the parametrization function $\mathbf{x}(s,t)$, the jumps defined above are independent of the chosen parametrization in the sense that, if $\tilde{\mathbf{x}}(\tilde{s},t)$ also parameterizes Γ, then all the jumps above are the same at each point $\tilde{\mathbf{x}}(\tilde{s},t) = \mathbf{x}(s,t)$.

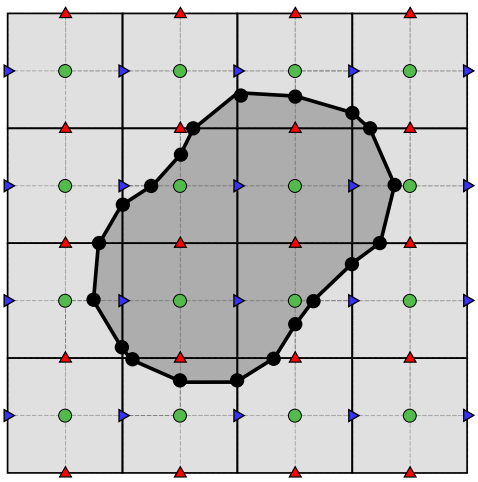

Fig. 14. The level set is defined on a doubly-fine grid with $2N$ cells per direction. The duplication of nodes and cells required by the method of [1] is done for the u, v and p grids separately.

3.2.2.2. Construction of ∇p. The method in [1] also uses duplicated grid cells along the interface to introduce additional nodal degrees of freedom when solving interface problems and this process must again be coordinated over the respective u, v and p subgrids. However, unlike the discontinuous viscosity case, we define pressure cells to contain four nodes at their corners in the same manner as u and v cells. This is because we must solve a Poisson interface problem for each of u, v and p and [1] requires the grids be node-centric. This effects the u and v solves since the p degrees of freedom interact through the right hand side terms in Poisson equations for u and v (where p_x and p_y respectively appear). Therefore, the duplication process must admit a procedure for the computation of the respective components of ∇p on each of the duplicated grids for u and v. This is complicated by the use of virtual p degrees of freedom in the ∇p stencils on cut cells.

Fortunately, we will show here that the definition of the level set over a "doubly-fine" grid admits a consistent duplication and subsequent ∇p transfer procedure. By "doubly-fine", we mean a grid with $2N$ cells per direction that contains all of the u, v and p nodes (see Fig. 14). We define the level set over this doubly-fine grid from the Lagrangian Γ_h in the same manner as described in Section 3.1. Necessary cell duplication is then performed by inheriting the doubly-fine level set to each of the u, v and p subgrids. After computing the doubly-fine level set, we define the Lagrangian zero isocontour on the doubly-fine grid ($\tilde{\Gamma}_h$). This gives a piecewise linear approximation to the zero isocontour over each cut cell in the doubly-fine grid. Then, for each subgrid we define the Lagranagian cell-wise approximation to the zero isocontour as the union of the piecewise linear approximations over the four grid cells in the doubly-fine grid that are contained in the subgrid. This can be seen in Figs. 14 and 15. Also, a u, v or p cell is considered cut (and then subsequently duplicated) if any of its four doubly-fine grid cells are cut.

The staggering of the variables on the MAC grid naturally facilitates a second order centered finite difference stencil for p_x at each u degree of freedom and for p_y at each v degree of freedom. Not coincidentally, these are precisely the source terms

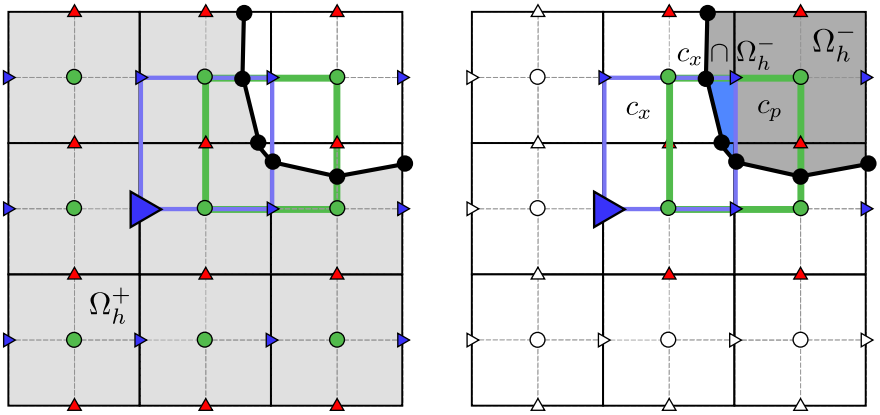

Fig. 15. A typical case where the computation of ∇p is not trivial. The colored nodes in figure (a) are nodes of the u, v and p grids associated with Ω_h^-, while the colored nodes on figure (b) are the analogous ones for Ω_h^+. The large blue triangle corresponds to a virtual node on the Ω_h^- u grid. The original node is shown enlarged on the left in Ω_h^+. Notice that simple methods such as centered differences cannot be used to compute p_x numerically on the node on the Ω_h^- side since there is no p node of Ω_h^- on its left. We highlight the cells c_x and c_p as well as the region $c_x \cap \Omega_h^-$ in the image at the right to aid in the discussion from Section 3.2.2.2. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

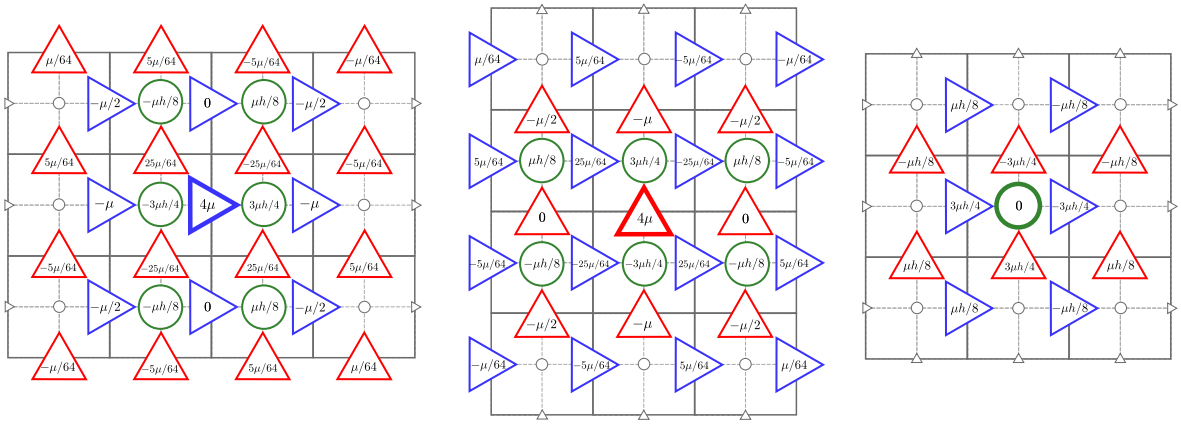

Fig. 16. Visualization of discrete stencils away from the interface in the matrix **K**. u node stencils are shown at the left, v node stencils are in the middle and p node stencils are shown at the right.

needed in the Poisson equations for u and v respectively. Unfortunately, these stencils are not always well defined for interfacial u and v degrees of freedom. For example consider the left subfigure in Fig. 15. Here, a centered difference computation of p_x at the virtual u node (drawn as a large blue triangle pointing to the right) is not possible because there is no p degree of freedom to the left.

Fortunately, [1] requires only cell-wise averages of the right hand side terms of the Poisson equations (see Section 4 in [1]). Therefore we will never explicitly require components of ∇p at virtual u and v nodes like the one node highlighted in the left subfigure in Fig. 15. We compute the required cell-wise averages as:

$$\overline{p_x} = \frac{\int_{c_x \cap \Omega_h^{\pm}} p_x dA}{\int_{c_x \cap \Omega_h^{\pm}} dA} \quad \text{and} \quad \overline{p_y} = \frac{\int_{c_y \cap \Omega_h^{\pm}} p_y dA}{\int_{c_y \cap \Omega_h^{\pm}} dA}. \tag{40}$$

Here, c_x and c_y are cells in the duplicated u and v grids respectively. $c_x \cap \Omega_h^{\pm}$ is the intersection of the fluid domain with c_x (similar for $c_y \cap \Omega_h^{\pm}$). Note that these regions are non-trivial for cells in the duplicated grids that intersect the interface. That is, these regions correspond to cut cells. An example of such a cut cell is shown incident to the enlarged u node in left subfigure of Fig. 15. While we do not have a p degree of freedom to the left of this node, we do have enough information to compute $\overline{p_x}$ over the cut cell. Specifically, we have enough information to determine a piecewise bilinear approximation to p over $c_x \cap \Omega_h^-$. Note that there is a p cell (c_p) containing region $c_x \cap \Omega_h^-$ in Fig. 15. This implies that we can approximate p as bilinear over the required region. In general, there will be at most two p cells overlapping any $c_x \cap \Omega_h^{\pm}$ or $c_y \cap \Omega_h^{\pm}$ region and we can always use this to generate a piecewise bilinear approximation to p wherever needed.

Using the bilinear approximation over the cell, we compute the necessary derivatives as:

$$p_x = \sum_{k=1}^{n_p} p_k N_{k,x}^p \quad \text{and} \quad p_y = \sum_{k=1}^{n_p} p_k N_{k,y}^p, \tag{41}$$

where N^p are the piecewise bilinear basis functions for the (now node-centric) pressure grid. The cell-wise averages ($\overline{p_x}$ and $\overline{p_y}$) of p_x and p_y can therefore be computed exactly as in [1]. This is easily done because the integrands in

$$\overline{p_x} = \frac{\sum_{k=1}^{n_p} p_k \int_{c_x \cap \Omega_h^{\pm}} N_{k,x}^p dA}{\int_{c_x \cap \Omega_h^{\pm}} dA} \quad \text{and} \quad \overline{p_y} = \frac{\sum_{k=1}^{n_p} p_k \int_{c_y \cap \Omega_h^{\pm}} N_{k,y}^p dA}{\int_{c_y \cap \Omega_h^{\pm}} dA} \tag{42}$$

are piecewise bilinear. That is, we need only integrate a low order polynomial in x and y over the polygonal cut cell regions $c_x \cap \Omega_h^{\pm}$ and $c_y \cap \Omega_h^{\pm}$. We compute these integrals in the same way as described in Section 3.2.1.3.

3.2.2.3. Projection onto divergence-free space. When we use the Poisson formulation (Eqs. (31) and (32)) for the discretization of u and v, we get second order (in L^∞) accurate velocities. However, they do not in general satisfy a discrete divergence free condition as was the case in our proposed method for the discontinuous viscosity case. While any consistent approximation of the divergence will converge to zero under refinement with the u and v we generate from the Poisson equations, it is often advantageous to enforce a discrete divergence free condition. We satisfy this divergence free condition via projection and our numerical experiments suggest that this process does not degrade the L^∞ accuracy.

The divergence free condition is defined over each cell in a p-node centered grid in the same way as for the discontinuous viscosity case (see Eq. (10)). For every p-node (including virtual p-nodes created in the duplication procedure), we define the p-centered grid cell to consist of the four surrounding sub cells in the doubly-fine grid. These cells are also duplicated whenever the p node at its center is virtual. We then discretize Eq. (10) in the same way as we did for the discontinous viscosity case. However, with the doubly-fine grid, normals are no longer constant over pressure cells so the constraint is slightly different as a consequence. If we again define $\mathbf{U}$ to be the vector of all discrete degrees of freedom of $\mathbf{u}_h$, then enforcing incompressibility requires that we modify the solution $\mathbf{u}_h$ so that $\mathbf{DU} = \mathbf{0}$, where $\mathbf{D} = \begin{pmatrix} \mathbf{D}^u & \mathbf{D}^v \end{pmatrix}$. The matrices $\mathbf{D}^u$ and $\mathbf{D}^v$ are defined exactly as in the discontinuous viscosity case (see Eqs. (76) and (77)). The projection of $\mathbf{U}$ obtained by solving the discrete Eqs. (31) and (32) can then be performed as:

1. Solve $\mathbf{D}\mathbf{D}^T\hat{\mathbf{P}} = \mathbf{D}\mathbf{U}$, where $\hat{\mathbf{P}} \in \mathbb{R}^{n_p}$
2. $\mathbf{U} \leftarrow \mathbf{U} - \mathbf{D}^T\hat{\mathbf{P}}$.

Notably, the linear system $\mathbf{D}\mathbf{D}^T\hat{\mathbf{P}} = \mathbf{D}\mathbf{U}$ is symmetric positive definite.

3.2.3. Irregular fluid domains

We also consider a single fluid with viscosity μ in an irregular domain Ω. As in Section 3.2.1, we define our discretization via the variational formulation:

$$\begin{aligned} &\text{Find } (\mathbf{u}, p, \mathbf{q}) \in \mathbf{H}^1(\Omega) \times \mathbf{H}^1(\Omega) \times \mathbf{L}^2(\Omega) \times \mathbf{L}^2(\partial\Omega) \times \mathbf{L}^2(\partial\Omega) \text{ such that } \int_\Omega \frac{\mu}{2} (\nabla \mathbf{w} + \nabla \mathbf{w}^T) \\ &\quad : (\nabla \mathbf{u} + \nabla \mathbf{u}^T)\, dA - \int_\Omega p(\nabla \cdot \mathbf{w})\, dA + \int_{\partial\Omega} \mathbf{w} \cdot \mathbf{q}\, dl = \int_\Omega \mathbf{w} \cdot \mathbf{f}\, dA, \quad \forall \mathbf{w} \in \mathbf{H}^1(\Omega) \times \mathbf{H}^1(\Omega), \end{aligned} \tag{43}$$

$$\int_{\Omega} \lambda \nabla \cdot \mathbf{u}\, dA = 0, \quad \forall \lambda \in \mathbf{L}^2(\Omega), \tag{44}$$

$$\int_{\partial\Omega} \boldsymbol{\psi}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x})\, dl = \int_{\partial\Omega} \boldsymbol{\psi}(\mathbf{x}) \cdot \mathbf{a}(\mathbf{x}), \quad \forall \boldsymbol{\psi} \in \mathbf{L}^2(\partial\Omega) \times \mathbf{L}^2(\partial\Omega), \tag{45}$$

where $\mathbf{q}(\mathbf{x}) = -(q^1, q^2)^T = -\sigma \cdot \mathbf{n}$ is defined only along $\partial\Omega$, with $\mathbf{n}$ being the outward unit normal at $\mathbf{x} \in \partial\Omega$. Note that we must again treat $\mathbf{q}$ as an additional unknown since $\sigma \cdot \mathbf{n}$ will not generally be known. This can now be viewed as the Lagrange multiplier associated with the Dirichlet boundary condition for the velocities (Eq. (3)). The discretization of the equations above is directly analogous to Section 3.2.1. $\mathbf{u}_h$, p_h and $\mathbf{q}_h$ are discretized in the same way as in Eqs. (14)–(16) respectively. The boundary geometry can be naturally be defined by a level set defined over the nodes of the pressure grid. This leads to a discrete version of $\partial\Omega$ which, just as in the interface case, has a single segment per pressure cell. This approximation to the level set isocontour is denoted $\partial\hat{\Omega}_h$. The interior of $\partial\hat{\Omega}_h$ is then the discrete irregular domain Ω_h. As in the case of interface problems with discontinuous viscosity, it is necessary to treat the boundary geometry in this manner to guarantee that we capture a discrete version of the null modes in the variational problem. Note that for irregular domain problems no duplication of nodes is necessary but virtual nodes are still used on cells which are cut by the discrete domain boundary $\partial\hat{\Omega}_h$.

4. Numerical examples

We present a number of numerical tests to demonstrate that we achieve second order accuracy in L^∞ for u and v and first order in L^∞ for p. We first define the notion of order of convergence that we use and then demonstrate the order with a number of examples using grid refinement studies. We also describe the preconditioned MINRES method we use for solving the symmetric indefinite systems of equations for discontinuous viscosity problems and we discuss the discretization of the Lagrangian interface forces and velocity interpolation. Finally, we also demonstrate the performance gains provided by our null-space capturing discretization.

4.1. Convergence measure

We examine the convergence behavior as a function of the grid spacing. We compare by running simulations at varying grid resolutions to a given time. We then estimate the error at this point in time by comparing the solutions at the different grid resolutions. We use N to denote the number of MAC grid cells per direction. We define our discrete approximation to a field g at grid point (x_i, y_j) as $g^N_{i,j}$. Let $E^N_{i,j} = g^N_{i,j} - g(x_i, y_j)$. We examine convergence in the point-wise infinity norm which we define as:

$$e^N = \left\| E^N \right\|_\infty = \max_{i,j} \left| E^N_{i,j} \right|. \tag{46}$$

We say that the method is rth order accurate if:

$$e^N \leqslant Ch^r \tag{47}$$

for some constant C with $h \propto 1/N$ being the cell width. Hence:

$$\left\| g^{2N} - g^N \right\|_\infty \leqslant \left\| g^{2N} - g \right\|_\infty + \left\| g^N - g \right\|_\infty \leqslant C(h^r_{2N} + h^r_N) = C' \left(\frac{1}{(2N)^r} + \frac{1}{N^r} \right) = C' \frac{1}{N^r} \left(\frac{1}{2^r} + 1 \right) = C'' \frac{1}{N^r}, \tag{48}$$

where all constants were incorporated into C''. Taking the $\log_{10}$ on both sides of the equation above and defining $a = \log_{10} C''$, we get:

$$\log_{10} \left\| g^{2N} - g^N \right\|_\infty \leqslant a - r \log_{10} N. \tag{49}$$

In other words, the negative of the slope of the plot of $\log_{10} \left\| g^{2N} - g^N \right\|_\infty$ versus $\log_{10} N$ is the order of convergence of the method in the L^∞ norm. We use this procedure to compute the order of convergence in the results that follow. Note that in order to define the quantity $g^{2N} - g^N$, they must be defined at common grid points (i,j). We do this by taking the difference at all grid points (i,j) in the grid associated with g^N as g^{2N} will also be defined at those points.

The discontinuities in the velocities and the pressure at the interface $\hat{\Gamma}_h$ require special treatment in our grid refinement studies. The comparison of g^{2N} and g^N must only be done at points on the same side of the interface. That is, a given grid point (i,j) may be on the interior of the interface for g^{2N} and on the exterior of the interface for g^N (or vice versa). This will happen because the geometry of the interface will change with N since the level set is also defined on a grid with resolution set by N. It would artificially degrade our error convergence estimates if we compare points on opposite sides of the interface. Therefore, we define the infinity norm of the difference between g^{2N} and g^N to only consider points that are on the same side of the interface on both grids. Note that this also precludes comparing the values at virtual degrees of freedom.

4.2. Jacobi preconditioned MINRES

Our discretization requires the solution of a symmetric linear system for the velocities, the pressure and the Lagrange multipliers. In all examples described below, we use MINRES to solve the linear systems involved. As noted in [1], the cut cells can lead to rows in the matrix with very small magnitude. This can significantly degrade the condition number of the system. We alleviate this with a simple Jacobi preconditioning strategy. This is slightly different that in [1] where the matrices were all symmetric positive definite. Our system is symmetric but with a zero lower right block. We therefore use a block Jacobi approach to alleviate rows with small magnitude.

We first construct a Jacobi preconditioner for the **A** matrix portion of **K**. We define this block of the preconditioner $\mathbf{J}^{(1)}$ as:

$$J_{ii}^{(1)} = \begin{cases} \frac{1}{\sqrt{K_{ii}}} & \text{if } i < N_u + N_v, \\ 1 & \text{if } i \geqslant N_u + N_v. \end{cases} \tag{50}$$

Let $\mathbf{K}^{(1)} = \mathbf{J}^{(1)}\mathbf{K}\mathbf{J}^{(1)}$. The diagonal entries in the first $N_u + N_v$ rows of $\mathbf{K}^{(1)}$ are 1 and its off-diagonal entries are smaller than 1. In the second step, we construct a preconditioning matrix that normalizes each row $i > N_u + N_v$ of $\mathbf{K}^{(1)}$, leaving its leading $(N_u + N_v) \times (N_u + N_v)$ block unchanged. Letting $\mathbf{k}_i$ be the ith row of $\mathbf{K}^{(1)}$, we construct the second preconditioning matrix $\mathbf{J}^{(2)}$ as:

$$J_{ii}^{(2)} = \begin{cases} 1 & \text{if} i < N_u + N_v, \\ \frac{1}{\|\mathbf{k}_i\|_2} & \text{if} i \geqslant N_u + N_v. \end{cases} \tag{51}$$

Letting $\mathbf{J} = \mathbf{J}^{(2)}\mathbf{J}^{(1)}$ and writing the solution vector in Eq. (19) as **x** and the right hand side as **b**, we then solve the following linear system:

$$\mathbf{J}\mathbf{K}\mathbf{J}\mathbf{J}^{-1}\mathbf{x} = (\mathbf{J}\mathbf{K}\mathbf{J})\mathbf{y} = \mathbf{J}\mathbf{b}, \tag{52}$$

where $\mathbf{y} = \mathbf{J}^{-1}\mathbf{x}$.

4.3. Interface advection

At the end of each time step, we advect the nodes of the interface Γ_h using forward Euler. To accomplish this we must interpolate u and v values from the MAC grid to each node $\mathbf{x}_i$ of Γ_h. This must be computed using virtual and material degrees of freedom associated with either Ω_h^+ or Ω_h^- but not both as in [1]. For that, we determine the original pressure cell at which $\mathbf{x}_i$ lies and use bilinear interpolation to compute the velocity values at $\mathbf{x}_i$ using only degrees of freedom from Ω_h^+ or Ω_h^-. The choice of region is in principle arbitrary since Γ_h and $\hat{\Gamma}_h$ approximate each other. In practice, however, we verify explicitly if $\mathbf{x}_i$ falls into Ω_h^+ or Ω_h^- by directly verifying whether or not it is inside the oriented polygonal surface of the portion of the original p cell inside of Ω_h^+. Using degrees of freedom of u and v associated with both sides of $\hat{\Gamma}_h$ would decrease the order of accuracy of our results since these quantities are not continuously differentiable across $\hat{\Gamma}_h$.

4.4. Elastic interface discretization

4.4.1. Discontinuous fluid viscosity

Our discretization of elastic interfacial forces is very standard, but we briefly cover it here for completeness. Let $\mathbf{x}(s,t)$ be a parametrization of the interface Γ. We assume that the range of s is $[0, L_0]$, where L_0 is the equilibrium length of the elastic interface. This equilibrium length L_0 is defined such that all elastic forces vanish at all points of Γ if its configuration is a circle of radius $R_0 = 2\pi/L_0$. The elastic force density (per unit length of the parameter s) at a given point $\mathbf{x}(s,t) \in \Gamma$ is given by the equation:

$$\mathbf{F}(s,t) = \frac{\partial}{\partial s}(T(s,t)\boldsymbol{\tau}(s,t)) \tag{53}$$

Table 1
Orders of convergence for the elastic interface problem of Section 4.4.1.

Quantity	$t = 0.1$	$t = 0.2$	$t = 0.3$	$t = 0.4$
u_h	1.76	1.80	2.40	1.85
v_h	1.60	1.85	2.03	2.03
p_h	0.63	0.85	0.82	0.69
Γ_h	2.55	2.47	2.48	2.47
$R_{\max}$	2.03	2.05	2.09	2.10
$R_{\min}$	2.02	2.08	2.08	2.10
ΔV_h	2.23	2.16	2.21	2.24

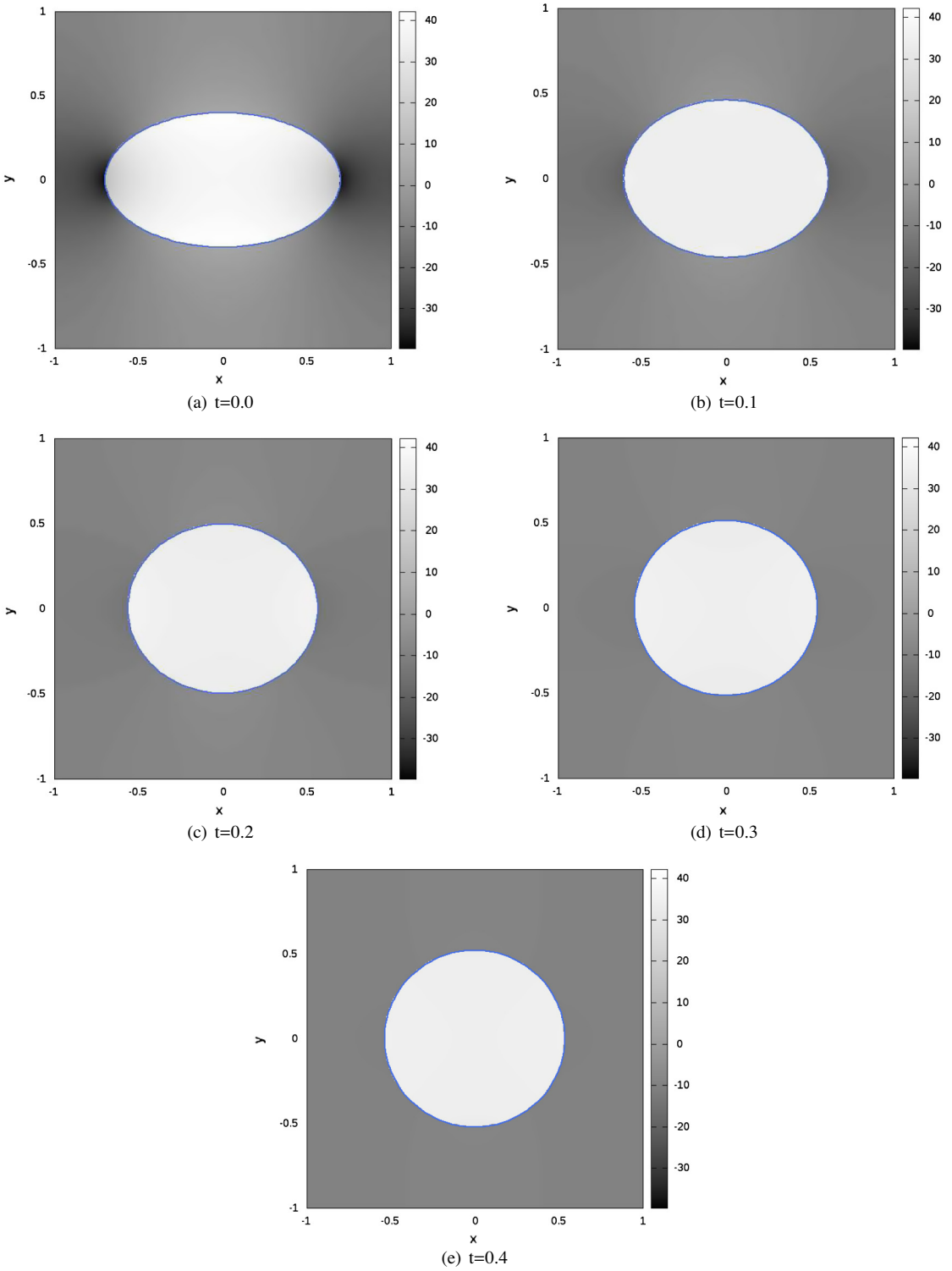

Fig. 17. Configuration of Γ_h at different times for the elastic interface problem of Section 4.4. Pressure is shown in gray. Dark regions have lower pressure; brighter regions have higher pressure.

where $\boldsymbol{\tau}(s,t)$ is the unit vector tangential to the interface at the point $\mathbf{x}(s,t)$:

$$\boldsymbol{\tau}(s,t) = \frac{\partial \mathbf{x}}{\partial s} \Big/ \left\| \frac{\partial \mathbf{x}}{\partial s} \right\| \tag{54}$$

and $T(s,t)$ is the tension at that point:

$$T(s,t) = \kappa(s)\left(\left\|\frac{\partial \mathbf{x}}{\partial s}\right\| - 1\right). \tag{55}$$

The function $\kappa(s)$ determines the elastic properties at each point of the interface. Here we will take $\kappa(s)$ to be a constant. The interface is discretely represented with M nodes and we compute the force density $\mathbf{F}_i$ on each node $\mathbf{x}_i$ of Γ_h as:

$$\boldsymbol{\tau}_{i+1/2} = \left(\frac{\partial \mathbf{x}}{\partial s}\right)_{i+1/2} \Bigg/ \left\|\frac{\partial \mathbf{x}}{\partial s}\right\|_{i+1/2} = \frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|}, \tag{56}$$

$$T_{i+1/2} = \kappa\left(\frac{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|}{\Delta s} - 1\right) \tag{57}$$

and

$$\mathbf{F}_i = \frac{T_{i+1/2}\boldsymbol{\tau}_{i+1/2} - T_{i-1/2}\boldsymbol{\tau}_{i-1/2}}{\Delta s} \tag{58}$$

with $\Delta s = L_0/M$. Here the subscript $i+1/2$ refers to the point at the center of the segment which connects the nodes $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. We divide the force density (58) by $\left\|\frac{\partial \mathbf{x}}{\partial s}\right\|_i = \frac{\|\mathbf{x}_i - \mathbf{x}_{i-1}\| + \|\mathbf{x}_{i+1} - \mathbf{x}_i\|}{2\Delta s}$ to define the jump conditions:

$$\mathbf{f}_i^i = \frac{2\Delta s}{\|\mathbf{x}_i - \mathbf{x}_{i-1}\| + \|\mathbf{x}_{i+1} - \mathbf{x}_i\|}\mathbf{F}_i. \tag{59}$$

For all elastic interface tests we use a rectangular periodic domain $[-1,1]\times[-1,1]$ with an initial elliptical interface of semi-major radius $a = 0.7$ and semi-minor radius $b = 0.4$ centered at the domain origin at $t = 0$. The ellipse is given uniform elasticity constant $\kappa = 10$. The viscosity of the fluid inside the interface is set to $\mu^- = 3$ and the viscosity of the fluid outside the interface is set to $\mu^+ = 1$. For a MAC grid with N cells per direction, the interface is represented with $M = N/2$ segments. The nodes of the interface are positioned initially according to the expression:

$$\mathbf{x}_i = (a\cos\theta_i, b\sin\theta_i), \tag{60}$$

where $\theta_i = 2\pi i/N_b = 4\pi i/N$ for $i = 1,2,\ldots,N/2$. This choice of the parametrization of the interface ensures each of its segments has length between $2.5h$ and $4.5h$ at all times of the simulation. The time step was $\Delta t = 5h^2 = 20/N^2$. We verified the order of convergence for the pressure, for the velocities and for the interface Γ_h according to the L^∞ norm at the times $t_1 = 0.1,\ t_2 = 0.2,\ t_3 = 0.3$ and $t_4 = 0.4$. Also, we checked how the total interface volume change $\Delta V_h =$ (interface volume at time t) − (interface volume at time $t = 0$) converged to zero as the grid was refined and the order of convergence of the maximum and minimum distances between the interface Γ_h and the origin of the rectangular domain (these distances are denoted below as $R_{\max}$ and $R_{\min}$ respectively). The numbers of grid cells per direction used on the convergence test were $N = 32, 64, 128, 256$. As can be seen from Table 1, all the results obtained are second order accurate for all these quantities except for the pressure, which is first order accurate. Fig. 17 shows the time evolution of the interface Γ_h.

Table 2
Orders of convergence for the elastic interface problem of Section 4.4.2.

Quantity	$t = 0.1$	$t = 0.2$	$t = 0.3$	$t = 0.4$
u_h	2.44	2.30	2.22	2.24
v_h	2.31	2.20	2.26	2.15
p_h	2.74	2.01	2.23	2.23
Γ_h	2.63	2.24	2.11	2.10
$R_{\max}$	2.10	2.05	2.02	2.02
$R_{\min}$	1.76	1.95	1.98	1.99
ΔV_h	2.05	2.01	2.01	2.01

Table 3
Orders of convergence for the surface tension problem of Section 4.5.1.

Quantity	$t = 0.1$	$t = 0.2$	$t = 0.3$	$t = 0.4$
u_h	2.05	1.80	2.25	1.88
v_h	1.89	2.16	1.85	1.87
p_h	0.84	0.97	1.06	0.86
Γ_h	1.82	2.01	2.21	2.34
$R_{\max}$	2.22	2.28	2.28	2.26
$R_{\min}$	1.95	1.96	1.98	1.98
ΔV_h	2.12	2.15	2.16	2.19

4.4.2. Continuous fluid viscosity

Using the method described on Section 3.2.2, we ran the elastic interface example of the previous section with $\mu^+ = \mu^- = 1$ and $N = 64, 128, 256$. The initial interface configuration and the elastic interface parameters are the same. As

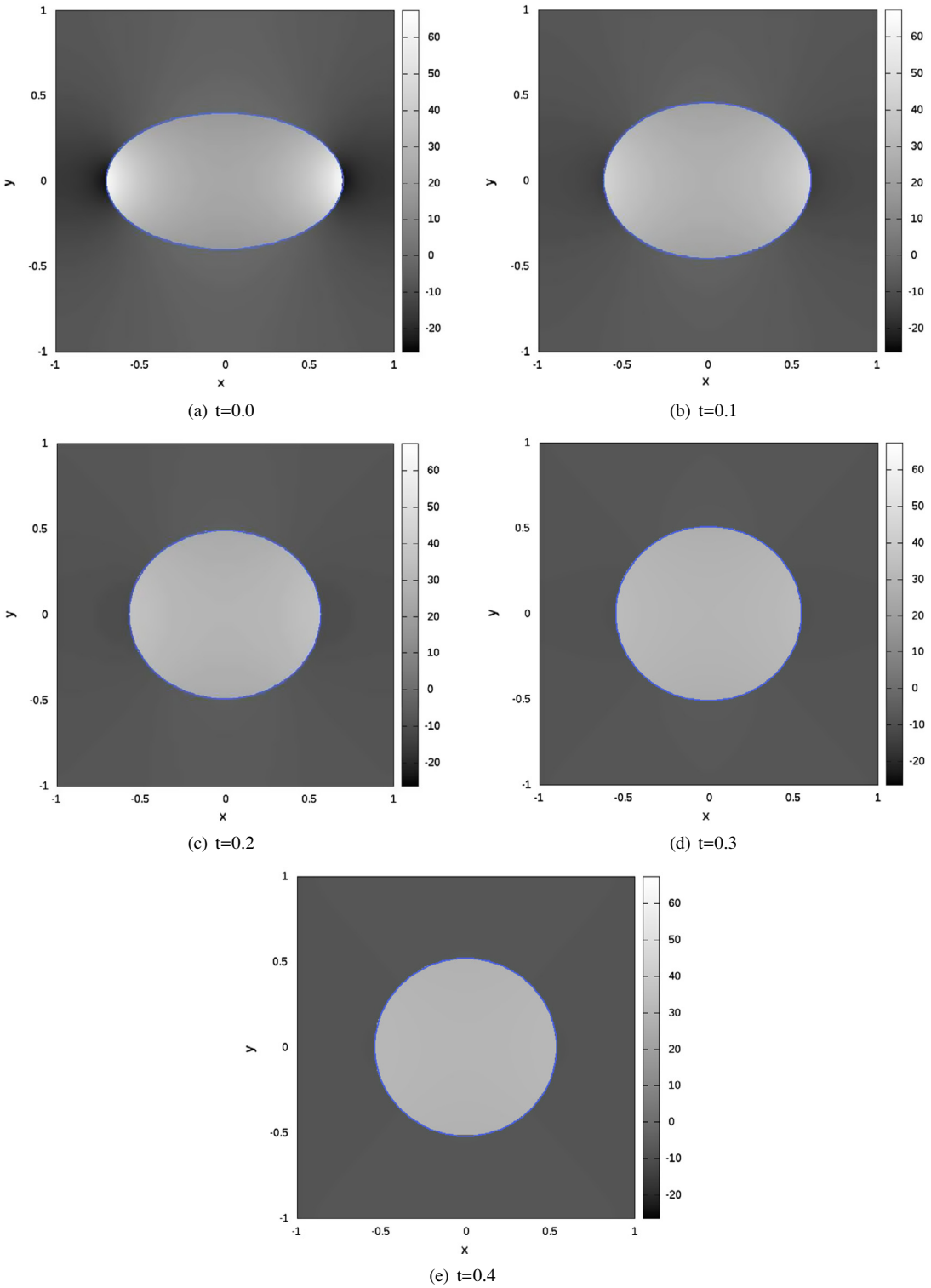

Fig. 18. Configuration Γ_h at different times for the surface tension problem of Section 4.5.1. Pressure values are shown in gray. Dark regions have lower pressure; brighter regions have higher pressure. Compare these results with the ones of Fig. 17.

can be seen from Table 2, all the results obtained are second order accurate for all quantities including the pressure. Snapshots of the interface evolution were omitted here for brevity.

4.5. Surface tension

4.5.1. Discontinuous Fluid Viscosity

Let $\mathbf{x}(s,t)$ be the parameterization of the interface with respect to arclength. The surface tension force density (per unit interface length) at a given point $\mathbf{x}(s,t) \in \Gamma$ is:

$$\mathbf{f}^i(s,t) = 2\sigma \frac{\partial \boldsymbol{\tau}(s,t)}{\partial s}, \tag{61}$$

where again $\boldsymbol{\tau}(s,t)$ is the unit vector tangential to the interface at the point $\mathbf{x}(s,t)$ and σ is the surface tension constant of the interface Γ. Since we are using the interface arclength as our parameterization, we have:

$$\left\| \frac{\partial \mathbf{x}}{\partial s} \right\| = 1. \tag{62}$$

The surface tension force density is then in this case:

$$\mathbf{f}_i^i = 2\sigma \frac{\tau_{i+1/2} - \tau_{i-1/2}}{\overline{\Delta s}}, \quad \overline{\Delta s} = \frac{\|\mathbf{x}_{i+1} - \mathbf{x}_i\| + \|\mathbf{x}_i - \mathbf{x}_{i-1}\|}{2}. \tag{63}$$

Here, $\tau_{i+1/2}$ is the same as in Eq. (56). Note that the surface tension force density (61) is always normal to the interface.

For all surface tension tests we use a rectangular periodic domain $[-1,1] \times [-1,1]$ with an elliptical interface of semi-major radius $a = 0.7$ and semi-minor radius $b = 0.4$ centered at the domain origin at $t = 0$. The fluid viscosities are again set to $\mu^+ = 1$ and $\mu^- = 3$, the surface tension constant is $\sigma = 10$. We use the same number of interface nodes as in the elastic interface examples: $M = N/2$ segments. The nodes of the interface are positioned initially as in the elastic interface case. Also the time step is again $\Delta t = 5h^2 = 20/N^2$.

We verified the order of convergence for the pressure, for the velocities and for the interface Γ_h according to the L^∞ norm at the times $t_1 = 0.1,\ t_2 = 0.2,\ t_3 = 0.3$ and $t_4 = 0.4$. Also, we again checked the order of convergence of the total interface volume change ΔV_h towards zero as the grid was refined and the order of convergences of $R_{\max}$ and $R_{\min}$. The numbers of MAC grid cells per direction used on the convergence tests were $N = 32, 64, 128, 256$. As can be seen from Table 3 all the results obtained are second order accurate for all these quantities except for the pressure, which is first order accurate. Fig. 18 shows the time evolution of the interface Γ_h.

Table 4
Orders of convergence for the surface tension problem of Section 4.5.2.

Quantity	$t = 0.1$	$t = 0.2$	$t = 0.3$	$t = 0.4$
u_h	2.25	1.90	1.99	2.05
v_h	2.05	1.84	2.06	2.12
p_h	1.75	1.79	1.92	2.05
Γ_h	2.06	1.95	2.37	2.10
$R_{\max}$	2.24	1.95	1.93	1.95
$R_{\min}$	1.96	1.96	1.97	1.97
ΔV_h	1.83	1.90	1.91	1.92

Table 5
Average number of MINRES iterations per time step for different grid resolutions.

N	Our method	Doubly-fine level set
32	978	1897
64	2500	5402
128	6635	16095

Table 6
Average run time per time step (in seconds) for different grid resolutions.

N	Our method	Doubly-fine level set
32	1.40	1.91
64	4.63	8.80
128	46.3	86.0

4.5.2. Continuous fluid viscosity

Using the method described on Section 3.2.2, we ran the surface tension example of the previous section with $\mu^+ = \mu^- = 1$ and $N = 64, 128, 256$. The initial interface configuration and the elastic interface parameters are the same. As can be seen from Table 4, all the results obtained are second order accurate for all quantities including the pressure. We have again omitted the snapshots of the interface evolution for brevity.

4.6. Pressure null mode test

Here we demonstrate the importance of capturing the constant pressure null mode. Our initial investigations used a doubly-fine level set to form the approximated Lagrangian interface $\hat{\Gamma}_h$ following the approach of Section 3.2.2. That is, we first

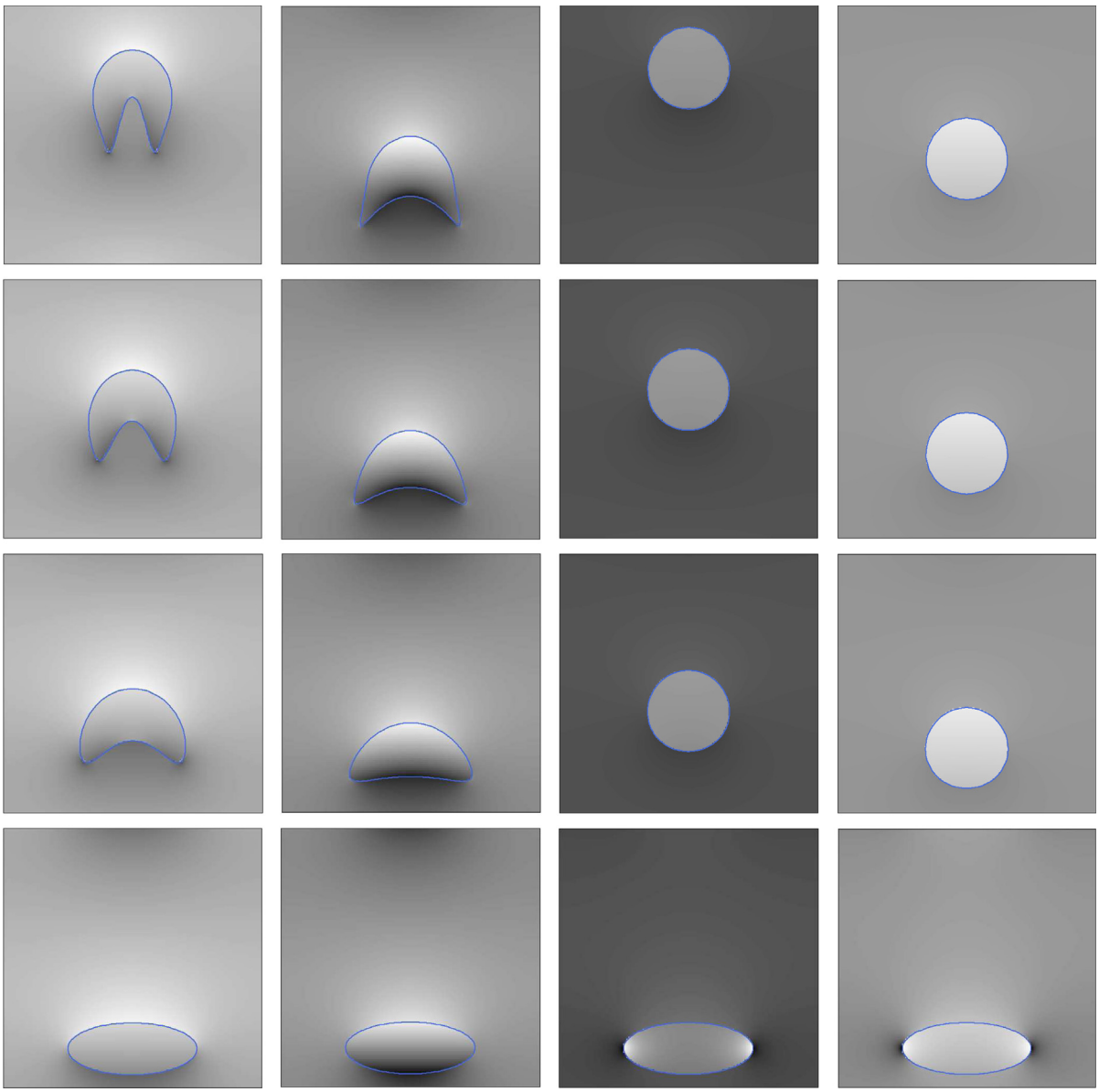

Fig. 19. Rising drop. Configurations of the interface Γ_h are shown at different times and for different combinations of σ, μ^+ and μ^-. Pressure is shown in gray. Dark regions have lower pressure; brighter regions have higher pressure. Each column represents a different test case; from bottom to top, the figures show the interface configurations at $t = 0.0$, $t = 3.0$, $t = 5.0$ and $t = 7.0$. From left to right, each column represents a test with each of these parameter combinations: $\mu^- = 3$, $\mu^+ = 1$, $\sigma = 0$ at column (a), $\mu^- = 1$, $\mu^+ = 3$, $\sigma = 0$ at column (b), $\mu^- = 3$, $\mu^+ = 1$, $\sigma = 1$ at column (c) and $\mu^- = 1$, $\mu^+ = 3$, $\sigma = 1$ at column (d).

Table 7
L^∞ norms of the errors for the discrete velocities and pressure at various grid resolutions and orders of convergence for the irregular domain problem of Section 4.8.

Quantity	e^{64}	e^{128}	e^{256}	e^{512}	OOC
u_h	2.15×10^{-3}	5.56×10^{-4}	1.45×10^{-4}	4.59×10^{-5}	1.86
v_h	1.73×10^{-3}	$6.48 \times 10E^{-4}$	1.62×10^{-4}	5.54×10^{-5}	1.69
p_h	7.31×10^{-1}	5.54×10^{-1}	3.03×10^{-1}	2.17×10^{-1}	0.61

used a level set defined over a grid with twice as many cells per dimension as the pressure grid. The nodes of the pressure grid are also nodes of this doubly-fine grid. Although this procedure is still adequate for consistently computing the integrals which define the system matrix **K** and the right hand side components, the resulting matrix **K** does not have the constant pressure mode in its kernel. This mode is then only captured approximately as $N \to \infty$. The effective conditioning of the system was significantly degraded by this. To illustrate, we ran the tests in the elasticity example section for $N = 32, 64, 128$ up to $t = 0.1$ and computed the average number of MINRES iterations for convergence and also the average amount of execution time per time step. Our residual tolerance was $r_{TOL} = 10^{-7}$. As can be seen from Tables 5 and 6, performance was significantly improved by capturing these modes.

4.7. Rising drop

We also demonstrate the effect of discontinuous material properties by simulating fluids with different densities and viscosities under with a gravitational body force. We use a uniform gravity $g = 10$, with fluid densities $\rho^- = 1$ inside the interface and the $\rho^+ = 2$ outside the interface. The viscosities are set to either $\mu^+ = 1$ and $\mu^- = 3$ or to $\mu^+ = 1$ and $\mu^- = 3$. We use a rectangular periodic domain $[-1, 1] \times [-1, 1]$ with an initial elliptical interface of semi-major radius $a = 0.5$ and semi-minor radius $b = 0.2$ centered at the position $(0, -0.7)$. We simulate interfaces with no surface tension and with surface tension constant $\sigma = 1$ for both these viscosity combinations. The interface is represented with $M = N$ segments for the cases in which the surface tension constant is set to zero and with $M = N/2$ segments otherwise. The nodes of the interface are positioned initially according to the expression:

$$\mathbf{x}_i = (a\cos\theta_i, b\sin\theta_i - 0.7), \tag{64}$$

where $\theta_i = 2\pi i/N_b = 4\pi i/N$ for $i = 1, 2, \ldots, N/2$. The time step is $\Delta t = 50h^2 = 200/N^2$. The results are shown on Fig. 19.

4.8. Irregular domains

As a final example, consider a"flower-shapped" domain Ω whose surface $\partial\Omega$ can be parametrized as:

$$r(\theta) = a + (b - a)\cos(5\theta/2)^2, \tag{65}$$

where θ ranges from 0 to 2π. Setting Dirichlet boundary conditions for the velocities, we probe the order of accuracy for the velocities and the pressure obtained with the method of Section 3.2.3. We analyze the convergence of the numerical solution towards the following analytical solution of the incompressible Stokes equations:

$$u = -\cos(\pi x)\sin(\pi y) \tag{66}$$

$$v = \sin(\pi x)\cos(\pi y) \tag{67}$$

$$p = \sin(\pi x)\sin(\pi y) \tag{68}$$

We embed the flower-shapped figure on Cartesian grids with $N = 64, 128, 256, 512$ cells per direction. The Cartesian grids encompass the domain $[-1, 1] \times [-1, 1]$. For efficiency purposes, cells which do not intersect the discrete fluid domain are not considered when solving the linear systems. As can be seen from Table 7 velocities are computed to second order accuracy while the pressure is computed to first order accuracy.

5. Conclusion

We presented a second order accurate method for the Stokes equations with immersed interfaces, discontinuous fluid properties and irregular domains. We also presented an optimized method in the case of continuous fluid viscosity. Our method is capable of resolving discrete counterparts of the continuous null modes for these interface problems and we showed that this is necessary for efficient performance. Also, the method is easy to implement and yields a symmetric linear system of equations. In the case of continuous viscosity, our systems are symmetric positive definite. Examples with elastic interfaces and surface tension were presented, as well as examples with drops which rise due to the presence of gravitational forces coupled with discontinuous fluid densities.

Acknowledgements

Authors were partially supported by NSF (DMS-0502315, DMS- 0652427, CCF-0830554, DOE (09-LR-04-116741-BERA), ONR (N000140310071, N000141010730, N000141210834) and an Intel STCVisual Computing Grant (20112360).

Appendix A. Equation derivation

We will derive the system of Eq. (20) in terms of x equations, y equations, p equations and $\mathbf{q}$ equations. Note that below we use Θ variables to help delineate quantities on different sides of the interface. Specifically, $\Theta_j^u = 1$ if the node j on the u grid is associated with a cut cell of Ω_h^+, $\Theta_j^u = -1$ if it is associated with a cut cell of Ω_h^- and $\Theta_j^u = 0$ for all other nodes; Θ_j^v is defined analogously for the nodes of the v grid. With this convention, our equations can be derived as:

x equations ($i = 1, \ldots, n_u$):

$$\sum_{j=1}^{n_u}\left(\mu\int_{\Omega_h\setminus\hat{\Gamma}_h} 2N_{i,x}^u N_{j,x}^u + N_{i,y}^u N_{j,y}^u\,dA\right)u_j + \sum_{j=1}^{n_v}\left(\mu\int_{\Omega_h\setminus\hat{\Gamma}_h} N_{i,y}^u N_{j,x}^v\,dA\right)v_j - \sum_{j=1}^{n_p}\left(\int_{\Omega_h\setminus\hat{\Gamma}_h} N_{i,x}^u\phi_j^p\,dA\right)p_j + \sum_{j=1}^{n_q}\left(\Theta_i^u\int_{\hat{\Gamma}_h} N_i^u\phi_j^q\,dl\right)q_j^1$$
$$= -\sum_{j=1}^{n_q}\left(\int_{\hat{\Gamma}_h}\frac{1}{2}N_i^u\phi_j^q\,dl\right)f^{i1}(\mathbf{p}_j) + \sum_{j=1}^{n_p}\left(\int_{\Omega_h\setminus\hat{\Gamma}_h} N_i^u\phi_j^p\,dA\right)\bar{f}_j^1 \quad \text{for all } i = 1,\ldots,n_u, \tag{69}$$

y equations ($i = 1, \ldots, n_v$):

$$\sum_{j=1}^{n_u}\left(\mu\int_{\Omega_h\setminus\hat{\Gamma}_h} N_{i,x}^v N_{j,y}^u\,dA\right)u_j + \sum_{j=1}^{n_v}\left(\mu\int_{\Omega_h\setminus\hat{\Gamma}_h} 2N_{i,y}^v N_{j,y}^v + N_{i,x}^v N_{j,x}^v\,dA\right)v_j - \sum_{j=1}^{n_p}\left(\int_{\Omega_h\setminus\hat{\Gamma}_h} N_{i,y}^v\phi_j^p\,dA\right)p_j + \sum_{j=1}^{n_q}\left(\Theta_i^v\int_{\hat{\Gamma}_h} N_i^v\phi_j^q\,dl\right)q_j^2$$
$$= -\sum_{j=1}^{n_q}\left(\int_{\hat{\Gamma}_h}\frac{1}{2}N_i^v\phi_j^q\,dl\right)f^{i2}(\mathbf{p}_j) + \sum_{j=1}^{n_p}\left(\int_{\Omega_h\setminus\hat{\Gamma}_h} N_i^v\phi_j^p\,dA\right)\bar{f}_j^2 \quad \text{for all } i = 1,\ldots,n_v, \tag{70}$$

p equations ($i = 1, \ldots, n_p$):

$$-\sum_{j=1}^{n_u}\left(\int_{\Omega_h\setminus\hat{\Gamma}_h}\phi_i^p N_{j,x}^u\,dA\right)u_j - \sum_{j=1}^{n_v}\left(\int_{\Omega_h\setminus\hat{\Gamma}_h}\phi_i^p N_{j,y}^v\,dA\right)v_j = 0, \quad \text{for all } i = 1,\ldots,n_p, \tag{71}$$

q_x equations ($i = 1, \ldots, n_q$):

$$\sum_{j=1}^{n_u}\left(\Theta_j^u\int_{\hat{\Gamma}_h}\phi_i^q N_j^u\,dl\right)u_j = 0, \quad \text{for all } i = 1,\ldots,n_q, \tag{72}$$

q_y equations ($i = 1, \ldots, n_q$):

$$\sum_{j=1}^{n_v}\left(\Theta_j^v\int_{\hat{\Gamma}_h}\phi_i^q N_j^v\,dl\right)v_j = 0, \quad \text{for all } i = 1,\ldots,n_q. \tag{73}$$

Note that in the x and y equations, μ takes the value μ^+ if the node i is associated with Ω_h^+ and μ^- if it is associated with Ω_h^-. We can rewrite the equations above using the notation introduced in Eq. (20) as:

$$A_{ij}^{uu} = \mu\int_{\Omega_h\setminus\hat{\Gamma}_h} 2N_{i,x}^u N_{j,x}^u + N_{i,y}^u N_{j,y}^u\,dA \quad G_{ij}^u = \int_{\Omega_h\setminus\hat{\Gamma}_h} N_{i,x}^u\phi_j^p\,dA \quad C_{ij}^u = \Theta_i^u\int_{\hat{\Gamma}_h} N_i^u\phi_j^q\,dl, \tag{74}$$

$$A_{ij}^{uv} = \mu\int_{\Omega_h\setminus\hat{\Gamma}_h} N_{i,y}^u N_{j,x}^v\,dA \quad G_{ij}^v = \int_{\Omega_h\setminus\hat{\Gamma}_h} N_{i,y}^v\phi_j^p\,dA \quad C_{ij}^v = \Theta_i^v\int_{\hat{\Gamma}_h} N_i^v\phi_j^q\,dl, \tag{75}$$

$$A_{ij}^{vu} = \mu\int_{\Omega_h\setminus\hat{\Gamma}_h} N_{i,x}^v N_{j,y}^u\,dA \quad D_{ij}^u = \int_{\Omega_h\setminus\hat{\Gamma}_h}\phi_i^p N_{j,x}^u\,dA \quad B_{ij}^u = \Theta_j^u\int_{\hat{\Gamma}_h}\phi_i^q N_j^u\,dl, \tag{76}$$

$$A_{ij}^{vv} = \mu\int_{\Omega_h\setminus\hat{\Gamma}_h} 2N_{i,y}^v N_{j,y}^v + N_{i,x}^v N_{j,x}^v\,dA \quad D_{ij}^v = \int_{\Omega_h\setminus\hat{\Gamma}_h}\phi_i^p N_{j,y}^v\,dA \quad B_{ij}^v = \Theta_j^v\int_{\hat{\Gamma}_h}\phi_i^q N_j^v\,dl, \tag{77}$$

$$\widehat{F}_i^1 = -\sum_{j=1}^{n_q}\left(\int_{\hat{\Gamma}_h}\frac{1}{2}N_i^u\phi_j^q\,dl\right)f^{i1}(\mathbf{p}_j) \quad \hat{f}_i^1 = \sum_{j=1}^{n_p}\left(\int_{\Omega_h\setminus\hat{\Gamma}_h} N_i^u\phi_j^p\,dA\right)\bar{f}_j^1, \tag{78}$$

$$\widehat{F}_i^2 = -\sum_{j=1}^{n_q}\left(\int_{\hat{\Gamma}_h}\frac{1}{2}N_i^v\phi_j^q\,dl\right)f^{i2}(\mathbf{p}_j) \quad \hat{f}_i^2 = \sum_{j=1}^{n_p}\left(\int_{\Omega_h\setminus\hat{\Gamma}_h}N_i^v\phi_j^p\,dA\right)\bar{f}_j^2. \tag{79}$$

Note that $\mathbf{A}^{uu}$ and $\mathbf{A}^{vv}$ are symmetric, while $\mathbf{A}^{uv} = (\mathbf{A}^{vu})^T$. Also, we have $\mathbf{G}_{ij}^u = \mathbf{D}_{ji}^u$, $\mathbf{G}_{ij}^v = \mathbf{D}_{ji}^v$, $\mathbf{C}_{ij}^u = \mathbf{B}_{ji}^u$ and $\mathbf{C}_{ij}^v = \mathbf{B}_{ji}^v$, so $\mathbf{G}^u = (\mathbf{D}^u)^T$, $\mathbf{G}^v = (\mathbf{D}^v)^T$, $\mathbf{C}^u = (\mathbf{B}^u)^T$ and $\mathbf{C}^v = (\mathbf{B}^v)^T$. Therefore, the system matrix $\mathbf{K}$ is symmetric.

References

[1] J. Bedrossian, J.H. von Brecht, S. Zhu, E. Sifakis, J.M. Teran, A second order virtual node method for elliptic problems with interfaces and irregular domains, J. Comput. Phys. 229 (2010) 6405–6426.
[2] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182–2189.
[3] J. Hellrung, L. Wang, E. Sifakis, J.M. Teran, A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions, J. Comput. Phys. 231 (4) (2012) 2015–2048.
[4] Y. Zhu, Y. Wang, J. Hellrung, A. Cantarero, E. Sifakis, J. Teran, A second-order virtual node algorithm for nearly incompressible linear elasticity in irregular domains, J. Comput. Phys. 231 (21) (2012) 7092–7117.
[5] A. Lew, G. Buscaglia, A discontinuous-Galerkin-based immersed boundary method, Int. J. Numer. Methods Eng. 76 (4) (2008) 427–454.
[6] F. Harlow, J. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182–2189.
[7] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.
[8] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.
[9] D.M. McQueen, C.S. Peskin, A three-dimensional computational method for blood flow in the heart. II: Contractile fibers, J. Comput. Phys. 82 (1989) 289–297.
[10] C.S. Peskin, D.M. McQueen, A three-dimensional computational method for blood flow in the heart. I: Immersed elastic fibers in a viscous incompressible fluid, J. Comput. Phys. 81 (2) (1989) 372–405.
[11] C.S. Peskin, D.M. McQueen, Modeling prosthetic heart valves for numerical analysis of blood flow in the heart, J. Comput. Phys. 37 (1) (1980) 113–132.
[12] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[13] Z. Li, M.-C. Lai, The immersed interface method for the Navier–Stokes equations with singular forces, J. Comput. Phys. 171 (2) (2001) 822–842.
[14] B.E. Griffith, C.S. Peskin, On the order of accuracy of the immersed boundary method: higher order convergence rates for sufficiently smooth problems, J. Comput. Phys. 208 (2005) 75–105.
[15] B.E. Griffith, R.D. Hornung, D.M. McQueen, C.S. Peskin, An adaptive, formally second order accurate version of the immersed boundary method, J. Comput. Phys. 223 (1) (2007) 10–49.
[16] A.M. Roma, C.S. Peskin, M.J. Berger, An adaptive version of the immersed boundary method, J. Comput. Phys. 153 (2) (1999) 509–534.
[17] C.S. Peskin, B.F. Printz, Improved volume conservation in the computation of flows with immersed elastic boundaries, J. Comput. Phys. 105 (1993) 33–46.
[18] R.J. Leveque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.
[19] R.J. LeVeque, Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (1997) 709–735.
[20] S. Xu, Z.J. Wang, An immersed interface method for simulating the interaction of a fluid with moving boundaries, J. Comput. Phys. 216 (2006) 454–493.
[21] D.V. Le, B.C. Khoo, J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, J. Comput. Phys. 220 (2006) 109–138.
[22] L. Lee, R.J. LeVeque, An immersed interface method for incompressible Navier–Stokes equations, SIAM J. Sci. Comput. 25 (2003) 832–856.
[23] T.Y. Hou, Z. Li, S. Osher, H. Zhao, A hybrid method for moving interface problems with application to the Hele-Shaw flow, J. Comput. Phys. 134 (1997) 236–252.
[24] Z. Li, K. Ito, M.-C. Lai, An augmented approach for Stokes equations with a discontinuous viscosity and singular forces, Comput. Fluids 36 (3) (2007) 622–635.
[25] Z. Tan, D.V. Le, K.M. Lim, B.C. Khoo, An immersed interface method for the incompressible Navier–Stokes equations with discontinuous viscosity across the interface, SIAM J. Sci. Comput. 31 (2009) 1798–1819.
[26] Z. Tan, D.V. Le, Z. Li, K.M. Lim, B.C. Khoo, An immersed interface method for solving incompressible viscous flows with piecewise constant viscosity across a moving elastic membrane, J. Comput. Phys. 227 (2008) 9955–9983.
[27] X. Zhong, A new high-order immersed interface method for solving elliptic equations with imbedded interface of discontinuity, J. Comput. Phys. 225 (1) (2007) 1066–1099.
[28] Y.C. Zhou, S. Zhao, M. Feig, G.W. Wei, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources, J. Comput. Phys. 213 (2006) 1–30.
[29] S. Xu, Z.J. Wang, A 3d immersed interface method for fluid–solid interaction, Comput. Methods Appl. Mech. Eng. 197 (25–28) (2008) 2068–2086.
[30] X.-D. Liu, R.P. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys. 160 (2000) 151–178.
[31] M. Kang, R.P. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (2000) 323–360.
[32] M. Hyman, Non-iterative numerical solution of boundary-value problems, Appl. Sci. Res. Sec. B 2 (1) (1952) 325–351.
[33] V. Sau'lev, On solving boundary value problems on high-performance computers by fictitious domain methods, Siberian Math. J. 4 (1963) 912–925.
[34] R. Glowinski, T. Pan, T. Hesla, D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow 25 (5) (1999) 755–794.
[35] R. Glowinski, T.-W. Pan, J. Periaux, A fictitious domain method for external incompressible viscous flow modeled by Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 112 (1–4) (1994) 133–148.
[36] R. Glowinski, T. Pan, T. Hesla, D. Joseph, J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flows, J. Comput. Phys. 169 (2001) 363–426.
[37] L. Parussini, V. Pediroda, Fictitious domain approach with hp-finite element approximation for incompressible fluid flow, J. Comput. Phys. 228 (2009) 3891–3910.
[38] L. Parussini, Fictitious domain approach via Lagrange multipliers with least squares spectral element method, J. Sci. Comput. 37 (2008) 316–335.
[39] F. Bertrand, P. Tanguy, F. Thibault, A three-dimensional fictitious domain method for incompressible fluid flow problems, Int. J. Numer. Methods Fluids 25 (6) (1997) 719–736.
[40] J.M. Teran, C.S. Peskin, Tether force constraints in Stokes flow by the immersed boundary method on a periodic domain, SIAM J. Sci. Comput. 31 (5) (2009) 3404–3416.
[41] G. Biros, L. Ying, D. Zorin, A fast solver for the Stokes equations with distributed forces in complex geometries, J. Comput. Phys. 193 (1) (2004) 317–348.
[42] V. Rutka, A staggered grid-based explicit jump immersed interface method for two-dimensional Stokes flows, Int. J. Numer. Methods Fluids 57 (10) (2008) 1527–1543.

[43] C. Daux, N. Moës, J. Dolbow, N. Sukumar, T. Belytschko, Arbitrary branched and intersecting cracks with the extended finite element method, Int. J. Numer. Methods Eng. 48 (12) (2000) 1741–1760.
[44] N. Sukumar, D. Chopp, N. Moës, T. Belytschko, Modeling holes and inclusions by level sets in the extended finite-element method, Comput. Methods Appl. Mech. Eng. 190 (46–47) (2001) 6183–6200.
[45] N. Moës, E. Béchet, M. Tourbier, Imposing Dirichlet boundary conditions in the extended finite element method, Int. J. Numer. Methods Eng. 67 (12) (2006) 1641–1669.
[46] J. Dolbow, A. Devan, Enrichment of enhanced assumed strain approximations for representing strong discontinuities: addressing volumetric incompressibility and the discontinuous patch test, Int. J. Numer. Methods Eng. 59 (1) (2004) 47–67.
[47] G. Wagner, N. Moës, W. Liu, T. Belytschko, The extended finite element method for rigid particles in Stokes flow, Int. J. Numer. Methods Eng. 51 (3) (2001) 293–313.
[48] R. Becker, E. Burman, P. Hansbo, A Nitsche extended finite element method for incompressible elasticity with discontinuous modulus of elasticity, Comput. Methods Appl. Mech. Eng. 198 (41–44) (2009) 3352–3360.
[49] A. Coppola-Owen, R. Codina, Improving Eulerian two-phase flow finite element approximation with discontinuous gradient pressure shape functions, Int. J. Numer. Methods Fluids 49 (12) (2005) 1287–1304.
[50] J. Chessa, T. Belytschko, An extended finite element method for two-phase fluids, J. Appl. Math. 70 (1) (2003) 10–17.
[51] A. Gerstenberger, W. Wall, An extended finite element method/Lagrange multiplier based approach for fluid–structure interaction, Comput. Methods Appl. Mech. Eng. 197 (19–20) (2008) 1699–1714.
[52] A. Almgren, J. Bell, P. Colella, T. Marthaler, A Cartesian grid projection method for the incompressible euler equations in complex geometries, SIAM J. Sci. Comput. 18 (5) (1997) 1289–1309.
[53] S. Marella, S. Krishnan, H. Liu, H. Udaykumar, Sharp interface Cartesian grid method. I: An easily implemented technique for 3d moving boundary computations, J. Comput. Phys. 210 (1) (2005) 1–31.
[54] Y. Ng, C. Min, F. Gibou, An efficient fluid-solid coupling algorithm for single-phase flows, J. Comput. Phys. 228 (23) (2009) 8807–8829.
[55] C. Batty, F. Bertails, R. Bridson, A fast variational framework for accurate solid–fluid coupling, ACM Trans. Graphics 26 (2007).
[56] I. Bijelonja, I. Demirdžić, S. Muzaferija, A finite volume method for incompressible linear elasticity, Comput. Methods Appl. Mech. Eng. 195 (44–47) (2006) 6378–6390.
[57] L. Beirão da Veiga, V. Gyrya, K. Lipnikov, G. Manzini, Mimetic finite difference method for the Stokes problem on polygonal meshes, J. Comput. Phys. 228 (19) (2009) 7215–7232.
[58] P. Barton, D. Drikakis, An Eulerian method for multi-component problems in non-linear elasticity with sliding interfaces, J. Comput. Phys. 229 (15) (2010) 5518–5540.
[59] D. Hill, D. Pullin, M. Ortiz, D. Meiron, An Eulerian hybrid weno centered-difference solver for elastic–plastic solids, J. Comput. Phys. 229 (24) (2010) 9053–9072.
[60] H. Han, X. Wu, A new mixed finite element formulation and the MAC method for the Stokes equations, SIAM J. Numer. Anal. 35 (2) (1998) 560–571.
[61] K. Ito, Z. Li, Interface conditions for Stokes equations with a discontinuous viscosity and surface sources, Appl. Math. Lett. 19 (3) (2006) 229–234.