

## A VIRTUAL NODE ALGORITHM FOR HODGE DECOMPOSITIONS OF INVISCID FLOW PROBLEMS WITH IRREGULAR DOMAINS\*

RUSSELL HOWES<sup>†</sup>, CRAIG SCHROEDER<sup>‡</sup>, AND JOSEPH M. TERAN<sup>‡</sup>

**Abstract.** We present an efficient discrete Hodge decomposition for velocity fields defined over irregular domains in two and three dimensions using a virtual node framework. The method is designed for use in the exact projection discretization of incompressible flow. We leverage the Poisson framework initially developed in [1] and [2]. This approach uses a signed distance function to represent the irregular domain embedded in a Cartesian grid and uses a variational approach to create a symmetric positive definite linear system. We present a novel modification to the previous approach that yields a 5-point stencil (7-point in 3D) across the entire computational domain. The original algorithm required a 9-point stencil (27-point in 3D) near the embedded irregular boundary. We show that this new condensed stencil enables a decomposition of the form  $\mathbf{A} = \mathbf{G}^T \mathbf{M}^{-1} \mathbf{G}$ , where  $\mathbf{M}$  is a diagonal weighting matrix and  $\mathbf{G}$  and  $\mathbf{D} = -\mathbf{G}^T$  are diagonal scalings of the standard central-difference gradient and divergence operators. We use this factored form as the basis of our discrete Hodge decomposition and show that this can be readily used for exact projection in incompressible flow. Numerical experiments suggest our method is second-order in  $L^\infty$  for pressures and first-order in  $L^\infty$ , second-order in  $L^1$  for velocities.

**Key words.** Level sets, Hodge decomposition, incompressible flow, MAC grid, virtual node algorithms.

**AMS subject classifications.** 76B99, 76M10, 76M20.

**1. Introduction.** Exact projection methods for incompressible flow are very effective because of their accuracy, stability, and relative ease of implementation [3]. The temporary introduction of artificial compressibility in the advection stage of these algorithms simplifies the interaction of the velocity and pressure. This intermediate compressible velocity field must then be projected to its nearest incompressible counterpart via Hodge decomposition. Projection is ultimately done with the solution of a Poisson equation for the pressure, and it is often stabilized with a MAC-style staggering of velocity and pressure variables [4]. This staggering naturally leads to second-order, discrete central difference gradient and divergence operators. The composition of these operators yields the standard 5-point Laplacian (7-point in 3D) for cell-centered pressures. Unfortunately, optimal accuracy is difficult to achieve for problems defined over irregular domains because the MAC staggering is designed for Cartesian grids. Many researchers have developed approaches that generalize the Cartesian MAC-based projection for regular domains to the irregular case, however it is very difficult to maintain the simplicity of the original approach without sacrificing accuracy or efficiency. For example, the immersed boundary method [5, 6] can be used to enforce boundary conditions on an irregular domain without any modification to the Cartesian case other than a change in the forcing terms. However, the regularized delta function conception of the right hand side terms degrades the convergence to first-order. The immersed interface method [7, 8] can be used to preserve optimal accuracy, however the associated discrete systems are generally no longer symmetric. This deviation from the standard Laplacian discretization prevents the use of fast solvers, leading to considerable computational expense.

\*Received January 8, 2013; accepted for publication June 14, 2013.

<sup>†</sup>Correspondence to Russell Howes.

<sup>‡</sup>Department of Mathematics, University of California, Los Angeles, 520 Portola Plaza, Math Sciences Building 6363, Los Angeles, CA 90095, USA ({rhowes; craig; jteran}@math.ucla.edu).

Methods that utilize a level set representation of the irregular domain can be used to define embedded Cartesian discretizations that balance efficiency with improved accuracy by leveraging sub-cell geometric detail [9, 10, 11, 1, 2]. In a recent related work, we showed that optimal velocity accuracy can be achieved for Stokes flow with a virtual node approach [12] which uses such a level set representation. However, using a variational approach yields linear systems that are typically of symmetric KKT type, (see e.g. [13]) so fast solvers like those used for the standard Laplacian discretization are not available. Notably, Gibou et al. have recently shown that level set approaches are very effective for exact projection discretization of incompressible flow [14, 15]. In the present work, we take a virtual node approach (developed in [1] and [2]) to factor the Poisson equation in a manner similar to that presented in [14].

Although there are many methods capable of achieving second-order accurate velocities, these methods tend to either require expensive remeshing, such as with finite elements, or yield linear systems which are indefinite (for example [12]) or even asymmetric (such as [8]). By contrast, accurate and positive definite methods for solving the Poisson equation are relatively easy to construct, but these discretizations generally do not carry forward to the problem of exact projection. We demonstrate a method that is capable of condensing a 9-point stencil (27-point in 3d) into a 5-point stencil (7-point in 3d) in a manner that admits a factorization of the Poisson operator and leads immediately to an exact projection discretization. This alteration to the Poisson stencil retains the second-order convergence of the original Poisson operator, but it only leads to first-order velocities. While this method does not present an improvement in the order of accuracy presented by [14], it opens up a promising and unexplored avenue for discretizing the exact projection problem.

**2. Exact projection and Hodge decomposition.** Since our focus is the Hodge decomposition aspect of an exact projection discretization, we will ignore viscous terms and focus on the inviscid Euler equations

$$(1) \quad \begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial \rho}{\partial \mathbf{x}} \mathbf{u} &= 0 \\ \rho \left( \frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{u}}{\partial \mathbf{x}} \mathbf{u} \right) &= -\nabla p \\ \nabla \cdot \mathbf{u} &= 0, \end{aligned}$$

with Dirichlet normal velocity boundary conditions  $\mathbf{u} \cdot \mathbf{n} = U_{BC}$  on  $\partial\Omega$ . A simple splitting of these equations give rise to the following temporal discretization

$$(2) \quad \begin{aligned} \rho^n \left( \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \frac{\partial \mathbf{u}^n}{\partial \mathbf{x}} \mathbf{u}^n \right) &= \mathbf{0} \\ \rho^n \left( \frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} \right) &= -\nabla p^{n+1} \\ \frac{\rho^{n+1} - \rho^n}{\Delta t} + \frac{\partial \rho^n}{\partial \mathbf{x}} \mathbf{u}^{n+1} &= 0. \end{aligned}$$

If we take the divergence of the second equation and note that  $\nabla \cdot \mathbf{u}^{n+1} = 0$ , we can equivalently define this step as

$$(3) \quad \begin{aligned} \nabla \cdot \left( \frac{\Delta t}{\rho^n} \nabla p^{n+1} \right) &= \nabla \cdot \mathbf{u}^* \\ \mathbf{u}^{n+1} &= \mathbf{u}^* - \frac{\Delta t}{\rho^n} \nabla p^{n+1}, \end{aligned}$$

where the boundary conditions for the Poisson equation are then of Neumann type  $\frac{\Delta t}{\rho^n} \nabla p^{n+1} \cdot \mathbf{n} = \mathbf{u}^* \cdot \mathbf{n} - U_{BC}$ .

In order to create an exact projection discretization we come up with a discrete volume-weighted approximation to  $\nabla$ , and denote it by  $\mathbf{G}$ . The Poisson equation for the pressure is then

$$(4) \quad \Delta t \mathbf{D} \mathbf{M}^{-1} \mathbf{G} p^{n+1} = \mathbf{D} \mathbf{u}^*,$$

where  $\mathbf{D} = -\mathbf{G}^T$  is the associated discrete approximation to the divergence and  $\mathbf{M}$  is a diagonal scaling that approximates  $\rho^n$  scaled by volume. In the sections that follow, we will show that the virtual node Poisson discretizations developed in [1] and [2] can be rewritten in a form that admits a scaled version of the standard MAC grid-based approximation of  $\nabla$  for  $\mathbf{G}$ . This modification to the original algorithm is the key step needed to apply it in Hodge decomposition-based exact projection. That is, with this decomposition we can see algebraically that  $\mathbf{D} \mathbf{u}^{n+1} = 0$  if we define  $\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \mathbf{M}^{-1} \mathbf{G} p^{n+1}$ .

**3. Discretization.** Our extension of [1] to approximate the Poisson problem (3) reduces the 9-point stencil of the previous work to a 5-point stencil. This is necessary to allow a decomposition of the Poisson matrix  $\mathbf{A} = \mathbf{D} \mathbf{M}^{-1} \mathbf{G}$  with a diagonal  $\mathbf{M}$  and the previously described  $\mathbf{D}$  and  $\mathbf{G}$ . For simplicity, we present our extension in two dimensions, describing how it differs from [1], and how we treat the right-hand side to enforce the boundary condition on the projected velocity  $\mathbf{u}^{n+1}$ . Finally, we briefly mention the slight modifications to [2] necessary to implement our method in three dimensions.

**3.1. Condensed stencil approach.** The energy-based discretization used in [1, 2] results in a 9-point stencil near the boundary in two dimensions. Away from the boundary, a novel quadrature rule was used to condense the stencil to the standard 5-point discretization. We present a novel modification of the stencil coefficients in [1, 2] that admits a 5-point stencil over the entire domain without sacrificing the second-order accuracy in  $L^\infty$  achieved in the original work.

As in [1, 2], we embed the domain  $\Omega$  in a regular Cartesian grid  $\mathcal{G}_p^h$  with equal grid spacing  $\Delta x = \Delta y = h$ . In our case, this grid is the subset of a standard MAC grid that has pressure degrees of freedom at its vertices, as shown in Figure 1(a). We include in the discretization all cells in  $\mathcal{G}_p^h$  that intersect  $\Omega$ , and refer to this subset of  $\mathcal{G}_p^h$  as  $\mathcal{C}_p^h = \{c_k \in \mathcal{G}_p^h, c_k \cap \Omega \neq \emptyset\} \subset \mathcal{G}_p^h$ , as shown in Figure 1(b). For convenience let  $\Omega_k = c_k \cap \Omega$ , and let  $\Omega_k^0$  be the same region transformed into coordinates  $[0, 1] \times [0, 1]$ . Since  $\Omega$  and its boundary  $\partial\Omega$  typically will not align with elements of the Cartesian grid, our discretization will include many pressure cells that only partially intersect with the domain  $\Omega$ . Some nodes of those cells will lie outside the domain. We refer to pressure nodes lying outside the domain as “virtual” nodes and their corresponding degrees of freedom as virtual degrees of freedom.

Also as in [1, 2], our discretization is designed by first assuming that our pressure field is piecewise bilinear over the cells in  $\mathcal{C}_p^h$ .  $p(\mathbf{x}) = \sum_{i=1}^{n_p} p_i N_i(\mathbf{x})$  for  $\mathbf{p} = (p_1, \dots, p_{n_p})^t \in \mathbf{R}^{n_p}$ . Here  $N_i(\mathbf{x})$  is the standard piecewise bilinear interpolation basis function associated with pressure grid vertex  $i$ ; and  $n_p$  denotes the number of degrees of freedom in the discretization, equal to the number of grid vertices that compose the cells of  $\mathcal{C}_p^h$ . Occasionally we will refer to the basis functions as  $N_{l,m}$ , where  $l$  and  $m$  represent the position of vertex  $i$  on the Cartesian grid. With this assumption, we start our approximation from the quadratic terms in the variational

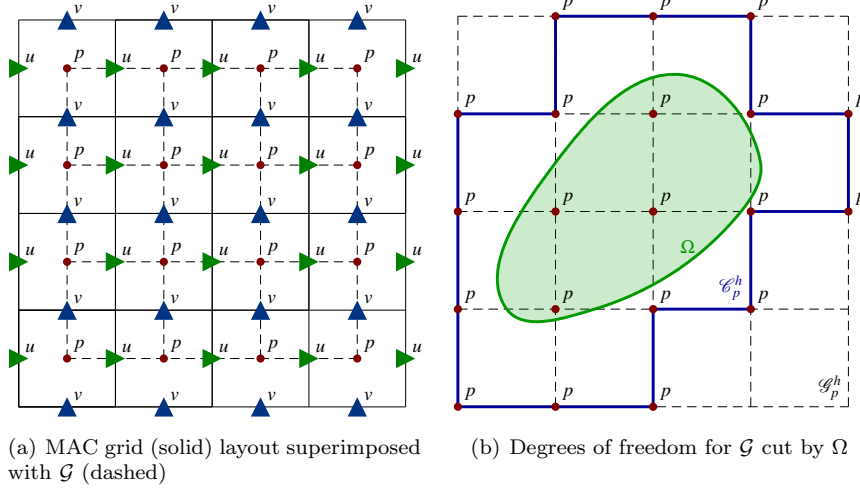


FIG. 1. Grid notation for our method. Our pressure grid has pressure nodes at the vertices, in contrast with the standard MAC grid (a) with pressures at cell centers. The computational domain (b) consists of all cells  $c_k \subset \mathcal{G}_p^h$  that intersect  $\Omega$ , and introduces virtual degrees of freedom for  $p$ , which lie outside of  $\Omega$ .

form of the Poisson equation

$$(5) \quad \psi(p) := \sum_{c_k \in \mathcal{C}_p^h} \frac{1}{2\rho} \sum_{r,s,r',s' \in \{0,1\}} \left( \int_{\Omega_k} \nabla N_{l+r,m+s} \cdot \nabla N_{l+r',m+s'} d\mathbf{x} \right) p_{l+r,m+s} p_{l+r',m+s'}$$

where  $l$  and  $m$  are the two-dimensional indices of the lower left node in pressure cell  $c_k$ . As in [1, 2], we perform the integration over cut cells using the divergence theorem to express each entry as a boundary integral. The cut cell boundary geometry is discretized from the level set representation of  $\Omega$ . The Hessian of this energy gives rise to the matrix in our variational approximation to the Poisson equation:  $E_{ij} := \frac{\partial^2}{\partial p_i \partial p_j} \psi(p)$ . We will now detail our approach for condensing the  $\mathbf{E} \in \mathbf{R}^{n_p \times n_p}$  stencil from up to 9 non-zero entries per row to at most 5 non-zero entries per row. We will refer to the condensed stencil matrix as  $\mathbf{A}$ .

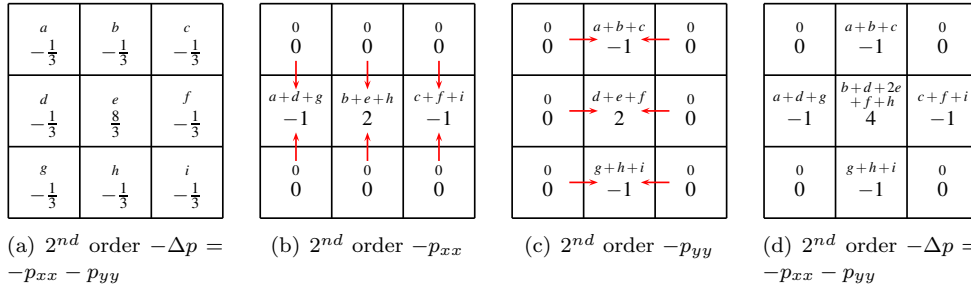


FIG. 2. An illustration of our ‘condensed stencil’ modification at a generic point in the domain. For a cell away from the boundary  $\partial\Omega$ , the 9-point second-order accurate stencil above (derived from bilinear finite elements) is condensed by our modification to the standard 5-point stencil.

The condensing procedure for a generic 9-point stencil, ignoring whether cells are cut or interior, is shown in Figure 2. The change of stencil coefficients can be thought of in terms of ‘pushing’ coefficients from the top and bottom of the stencil into the middle approximating the  $-p_{xx}$  portion of the Laplacian (Figure 2(b)), and from the left and right sides of the stencil into the center to approximate the  $-p_{yy}$  portion of the Laplacian (Figure 2(c)). The ‘condensing’ of the coefficients in this case leads to a 5-point stencil—the standard 5-point stencil if the center point is not incident on any cut cells.

At each node, the corresponding contributions to the stencil come from the four pressure cells incident to that node. For each pressure cell  $c_k$  in the discretization, we define  $\mathbf{E}_{c_k}$  to be the  $4 \times 4$  matrix containing the contributions of  $c_k$  to the matrix  $\mathbf{E}$ . We call  $\mathbf{E}_{c_k}$  the *element stiffness matrix* corresponding to  $c_k$ :  $(\mathbf{E}_{c_k})_{ij} = \int_{\Omega_k^0} a_{ij} d\mathbf{x}^0 = \int_{\Omega_k^0} \nabla N_i^0 \cdot \nabla N_j^0 d\mathbf{x}^0$ , where  $\Omega_k^0$  is the corresponding scaling of  $c_k \cap \Omega$  to the unit square and we denote the four basis functions supported over  $c_k$  as in Figure 4(b), written in a scaled coordinate frame local to the element:

$$(6) \quad N_0^0 = (1-x)(1-y) \quad N_1^0 = x(1-y) \quad N_2^0 = (1-x)y \quad N_3^0 = xy.$$

The element stiffness matrix can be written as

$$\mathbf{E}_{c_k} = \int_{\Omega_k^0} \begin{pmatrix} \nabla N_0^0 \\ \nabla N_1^0 \\ \nabla N_2^0 \\ \nabla N_3^0 \end{pmatrix} \begin{pmatrix} \nabla N_0^0 \\ \nabla N_1^0 \\ \nabla N_2^0 \\ \nabla N_3^0 \end{pmatrix}^T d\mathbf{x}^0.$$

Substituting in (6) yields the polynomial matrix

$$(7) \quad \mathbf{E}_{c_k} = \int_{\Omega_k^0} \begin{pmatrix} w^2 + z^2 & -z^2 - xw & -w^2 - yz & xw + yz \\ -z^2 - xw & z^2 + x^2 & xw + yz & -x^2 - yz \\ -w^2 - yz & xw + yz & w^2 + y^2 & -y^2 - xw \\ xw + yz & -x^2 - yz & -y^2 - xw & x^2 + y^2 \end{pmatrix} d\mathbf{x}^0$$

where for brevity we write  $w = x - 1$  and  $z = y - 1$ . We further give these polynomials labels to make the condensation operation easier to follow: the label  $a_{00}$ , for example, represents the polynomial  $(x - 1)^2 + (y - 1)^2$ .

$$(8) \quad \mathbf{E}_{c_k} = \int_{\Omega_k^0} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} d\mathbf{x}^0.$$

We rearrange the elements of  $\mathbf{E}_{c_k}$  in the manner illustrated in Figure 3 for a single row to create a modified element stiffness matrix  $\mathbf{A}_{c_k}$ . This process maintains a stencil consistent with our Poisson problem while moving all nonzero terms of the cellwise stiffness matrix to entries of  $\mathbf{A}$  where the row and column correspond to equal or adjacent nodes. This condensing process for the four cells incident to a node produces a 5-point stencil for that node, similar to Figure 2, and results in the modified matrix

$$(9) \quad \mathbf{A}_{c_k} = \int_{\Omega_k^0} \begin{pmatrix} 2a_{00} + a_{01} + a_{02} & a_{01} + a_{03} & a_{02} + a_{03} & 0 \\ a_{10} + a_{12} & 2a_{11} + a_{10} + a_{13} & 0 & a_{12} + a_{13} \\ a_{20} + a_{21} & 0 & 2a_{22} + a_{20} + a_{23} & a_{21} + a_{23} \\ 0 & a_{31} + a_{30} & a_{32} + a_{30} & 2a_{33} + a_{31} + a_{32} \end{pmatrix} d\mathbf{x}^0,$$

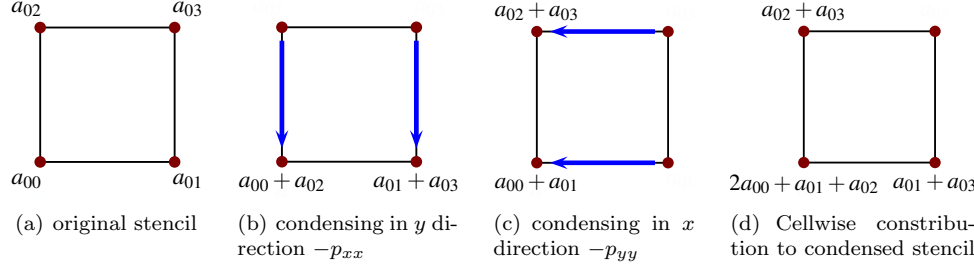


FIG. 3. We examine the changes made to the first row (corresponding to node 0) of the element stiffness matrix  $\mathbf{E}_{c_k}$  from [1] by the condensing procedure illustrated in Figure 2 to yield our element stiffness matrix (9). Condensing the coefficients as shown ensures that the nonzero terms in each row of  $\mathbf{A}$  correspond only to adjacent nodes on the same row or column, and allows for a 5-point stencil.

which combined with (7) and (8) simplifies to

$$(10) \quad \mathbf{A}_{c_k} = \int_{\Omega_k^0} \begin{pmatrix} 2-y-x & -1+y & -1+x & 0 \\ -1+y & 1-y+x & 0 & -x \\ -1+x & 0 & 1-x+y & -y \\ 0 & -x & -y & x+y \end{pmatrix} d\mathbf{x}^0.$$

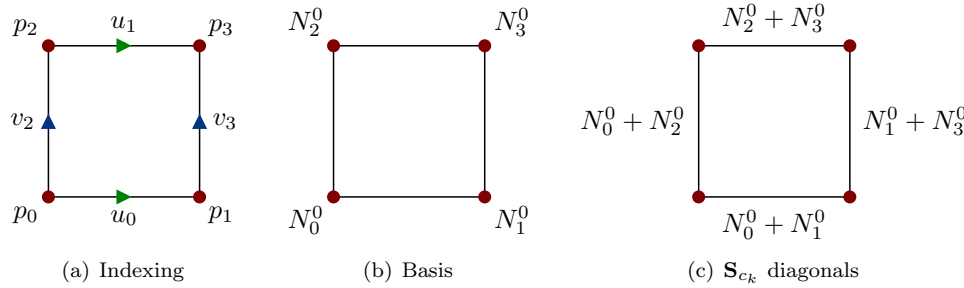


FIG. 4. The naming conventions used in referring to pressures and velocities (a) and pressure degrees of freedom on a grid cell (b). Additionally, (c) shows, at each velocity node, the basis functions contributing to that term of the cellwise weight matrix  $\mathbf{S}$ .

**3.2. Factorization.** Let  $\mathbf{P}_{c_k}$  be the matrix that relates the pressure indices for pressure cell  $c_k$  to indices in the full grid. That is, the matrix  $\mathbf{P}_{c_k}$  has four columns and as many rows as pressure nodes. It will have a single 1 per column with 0 everywhere else. Similarly, let  $\mathbf{Q}_{c_k}$  be the matrix that relates the MAC velocity indices incident on pressure cell  $c_k$  to indices in the full grid.  $\mathbf{Q}_{c_k}$  has four columns and as many rows as MAC faces, again with a single 1 per column. These operators will allow us to build the individual pressure cell contributions onto the full system in a rather convenient form later. Figure 4 shows the indexing convention we use in this work for the degrees of freedom associated with a pressure cell, and by extension, the columns of  $\mathbf{P}_{c_k}$  and  $\mathbf{Q}_{c_k}$ .

If  $\hat{\mathbf{G}}$  is the standard central difference operator defined over the entire domain,

then we can define a pressure cellwise  $\hat{\mathbf{G}}_{c_k}$  by restricting  $\hat{\mathbf{G}}$  to the pressure cell as

$$(11) \quad \hat{\mathbf{G}}_{c_k} = \mathbf{Q}_{c_k}^T \hat{\mathbf{G}} \mathbf{P}_{c_k} = \frac{1}{\Delta x} \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}$$

With this, we can factor  $\mathbf{A}_{c_k}$  as

$$\begin{aligned} \mathbf{A}_{c_k} &= \int_{\Omega_k^0} \begin{pmatrix} 2-y-x & -1+y & -1+x & 0 \\ -1+y & 1-y+x & 0 & -x \\ -1+x & 0 & 1-x+y & -y \\ 0 & -x & -y & x+y \end{pmatrix} d\mathbf{x}^0 \\ &= \int_{\Omega_k^0} \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}^T \begin{pmatrix} 1-y & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & 1-x & 0 \\ 0 & 0 & 0 & x \end{pmatrix} \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} d\mathbf{x}^0 \\ &= \hat{\mathbf{G}}_{c_k}^T \mathbf{S}_{c_k} \hat{\mathbf{G}}_{c_k}, \end{aligned}$$

where we have made the definition

$$(12) \quad \begin{aligned} \mathbf{S}_{c_k} &= \Delta x^2 \int_{\Omega_k^0} \begin{pmatrix} 1-y & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & 1-x & 0 \\ 0 & 0 & 0 & x \end{pmatrix} d\mathbf{x}^0 \\ &= \int_{\Omega_k} \begin{pmatrix} N_{l,m} + N_{l+1,m} & 0 & 0 & 0 \\ 0 & N_{l,m+1} + N_{l+1,m+1} & 0 & 0 \\ 0 & 0 & N_{l,m} + N_{l,m+1} & 0 \\ 0 & 0 & 0 & N_{l+1,m} + N_{l+1,m+1} \end{pmatrix} d\mathbf{x}. \end{aligned}$$

The matrix  $\mathbf{S}_{c_k}$  is just a diagonal matrix formed by integrating basis functions over the pressure cell.

Observe that pressure gradients are naturally computed at the MAC faces, which lie on edges of the pressure cells. Mathematically, we can say that  $\mathbf{Q}_{c_k}^T \hat{\mathbf{G}} = \mathbf{Q}_{c_k}^T \hat{\mathbf{G}} \mathbf{P}_{c_k} \mathbf{P}_{c_k}^T$ . Put another way, we can look solely at the MAC faces associated with a particular pressure cell, then ignore all pressures not associated with the cell. With this observation, it is possible to construct the full Poisson matrix from the individual pressure cell contributions  $\mathbf{A}_{c_k}$ .

$$\begin{aligned} \mathbf{A} &= \sum_k \mathbf{P}_{c_k} \mathbf{A}_{c_k} \mathbf{P}_{c_k}^T \\ &= \sum_k \mathbf{P}_{c_k} \hat{\mathbf{G}}_{c_k}^T \mathbf{S}_{c_k} \hat{\mathbf{G}}_{c_k} \mathbf{P}_{c_k}^T \\ &= \sum_k \mathbf{P}_{c_k} (\mathbf{Q}_{c_k}^T \hat{\mathbf{G}} \mathbf{P}_{c_k})^T \mathbf{S}_{c_k} (\mathbf{Q}_{c_k}^T \hat{\mathbf{G}} \mathbf{P}_{c_k}) \mathbf{P}_{c_k}^T \\ &= \sum_k \hat{\mathbf{G}}^T \mathbf{Q}_{c_k} \mathbf{S}_{c_k} \mathbf{Q}_{c_k}^T \hat{\mathbf{G}} \\ &= \hat{\mathbf{G}}^T \left( \sum_k \mathbf{Q}_{c_k} \mathbf{S}_{c_k} \mathbf{Q}_{c_k}^T \right) \hat{\mathbf{G}} \\ &= \hat{\mathbf{G}}^T \mathbf{S} \hat{\mathbf{G}}, \end{aligned}$$

where

$$\mathbf{S} := \sum_k \mathbf{Q}_{c_k} \mathbf{S}_{c_k} \mathbf{Q}_{c_k}^T.$$

This is a diagonal matrix since it is the sum of diagonal matrices. Each diagonal entry lives at an edge of the pressure grid and is formed by adding the pressure basis functions at the endpoints and integrating the result over the two pressure cells incident to the edge. The matrix  $\hat{\mathbf{G}}^T \mathbf{S} \hat{\mathbf{G}}$  is very easy to construct and apply since it is composed of standard central difference gradient and divergence operators, with a simple diagonal scaling in between.

The discussion so far has assumed constant density, omitting it from the discretization. If  $\mathbf{R}$  is a diagonal matrix defined over MAC faces whose diagonal entries are the density at MAC faces, then the appropriate system is  $\mathbf{A} = \hat{\mathbf{G}}^T \mathbf{R}^{-1} \mathbf{S} \hat{\mathbf{G}}$ , noting that  $\mathbf{R}$  and  $\mathbf{S}$  are diagonal and commute. With this factorization complete, we are ready to define  $\mathbf{G}$  and  $\mathbf{M}$ , which we do as

$$\mathbf{G} = \mathbf{S} \hat{\mathbf{G}} \quad \mathbf{M} = \mathbf{S} \mathbf{R}.$$

Now,  $\mathbf{A} = \hat{\mathbf{G}}^T \mathbf{R}^{-1} \mathbf{S} \hat{\mathbf{G}} = \mathbf{G}^T \mathbf{M}^{-1} \mathbf{G}$ , which is the desired form.

**3.3. Right-hand side.** As noted in Section 2, the Poisson equation in (3) has Neumann boundary conditions

$$(13) \quad \frac{\Delta t}{\rho^n} \nabla p^{n+1} \cdot \mathbf{n} = \mathbf{u}^* \cdot \mathbf{n} - U_{BC}$$

so we want the right-hand side of our discretized system to approximate the right-hand side of the corresponding weak form for  $p$ :

$$(14) \quad \int_{\Omega} \nabla q \cdot \nabla p \, d\mathbf{x} - \int_{\Omega} (\nabla \cdot \mathbf{u}^*) q \, d\mathbf{x} + \int_{\partial\Omega} (\mathbf{u}^* \cdot \mathbf{n} - U_{BC}) q \, d\mathbf{S}(\mathbf{x})$$

where  $q = \sum_i q_i N_i$  is a test function. Near the embedded boundary, the operator  $(\mathbf{G}^T \mathbf{u}^*)_i$  approximates not  $-\int_{\Omega} (\nabla \cdot \mathbf{u}^*) N_i \, d\mathbf{x}$  but  $-\int_{\Omega} (\nabla \cdot \mathbf{u}^*) N_i \, d\mathbf{x} + \int_{\partial\Omega} \mathbf{u}^* \cdot \mathbf{n} N_i \, d\mathbf{S}(\mathbf{x})$ , which we can understand by recalling the weak form used in constructing the original stiffness matrix  $\mathbf{E}$ . Therefore, to satisfy the Neumann boundary condition (13) we need to approximate the rest of the boundary condition

$$-\int_{\partial\Omega} U_{BC} N_i \, d\mathbf{S}(\mathbf{x}).$$

We approximate  $U_{BC}$  with a linear interpolant  $u_{BC}$  and solve

$$(15) \quad \Delta t \mathbf{G}^T \mathbf{M}^{-1} \mathbf{G} p^{n+1} = \mathbf{G}^T \mathbf{u}^* - \mathbf{u}_{BC}; \quad (\mathbf{u}_{BC})_i = \int_{\partial\Omega} u_{BC} N_i \, d\mathbf{S}(\mathbf{x}).$$

Our projection step then entails solving

$$(16) \quad \begin{aligned} \Delta t \mathbf{G}^T \mathbf{M}^{-1} \mathbf{G} p^{n+1} &= \mathbf{G}^T \mathbf{u}^* - \mathbf{u}_{BC}; \\ \mathbf{u}^{n+1} &= \mathbf{u}^* - \Delta t \mathbf{M}^{-1} \mathbf{G} p^{n+1}. \end{aligned}$$



**3.4. Modifications for three dimensions.** Implementation of our condensed stencil method in three dimensions is also a straightforward extension of the existing virtual node approach for Poisson problems and has the effect of condensing a 27-point stencil into a 7-point stencil which is equivalent to the standard 7-point stencil away from the boundary. As the stencil in Figure 2 was created by ‘pushing’ terms along the  $x$  direction to approximate  $-p_{yy}$ , and similarly along the  $y$  direction to approximate  $-p_{xx}$ , in three dimensions we push terms of the 27-point stencil along the  $x$  and  $y$  directions to approximate  $-p_{zz}$ , and a similar approach generates approximations to  $-p_{xx}$  and  $-p_{yy}$ .

Each pressure grid cell in 3D has eight adjacent pressure nodes and four each of  $x, y$ , and  $z$  velocities (see Figure 5). Therefore, the sizes of the cellwise matrices  $\mathbf{G}_{c_k}$ ,  $\hat{\mathbf{G}}_{c_k}$ ,  $\mathbf{S}_{c_k}$ , and  $\mathbf{A}_{c_k}$  differ from the 2D case, as do the matrices  $\mathbf{P}_{c_k}$  and  $\mathbf{Q}_{c_k}$  described in Section 3.2 relating indices for a pressure cell to indices in the full grid. For example,  $\hat{\mathbf{G}}$  is a 12-by-8 matrix defined in a straightforward manner like in (11).

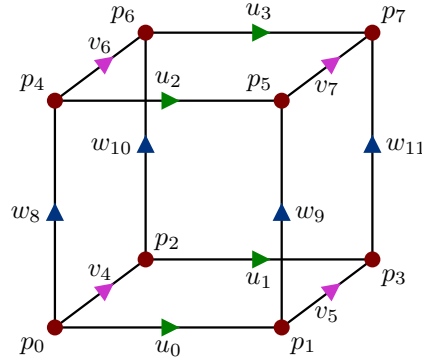


FIG. 5. Naming conventions used in referring to pressures and velocities on a grid cell in three dimensions. Note that velocities are located at cell edges in our grid, rather than at cell faces as with a MAC grid.

The virtual node cellwise stiffness matrix  $\mathbf{E}_{c_k}$  is an 8-by-8 matrix containing contributions of the cell  $c_k$  to the stiffness matrix: In three dimensions  $(\mathbf{E}_{c_k})_{ij} = \int_{\Omega_k^0} a_{ij} d\mathbf{x}^0 = dx \int_{\Omega_k^0} \nabla N_i^0 \cdot \nabla N_j^0 d\mathbf{x}^0$ . The  $dx$  scaling appears in the 3D case because the cell volume scales as  $dx^3$  and the gradient functions each scale as  $1/dx$ . The condensed element stiffness matrix  $\mathbf{A}_{c_k}$  is derived in a manner analogous to the 2D case, condensing all coefficients to matrix entries where the row node and column node are adjacent. This leads to four nonzero entries per row: the diagonal entry and the columns corresponding to the three adjacent nodes.  $\mathbf{P}_{c_k}$  has eight rows,  $\mathbf{Q}_{c_k}$  has twelve rows, and the cellwise scaling matrix  $\mathbf{S}_{c_k}$  is a 12-by-12 diagonal matrix similar in form to the right-hand side of (13). The diagonal entry of  $\mathbf{S}_{c_k}$  for each velocity node comes from the integrals of the basis functions corresponding to the two adjacent pressure nodes. The diagonal entries of  $\mathbf{S}$  are created by summing the contributions of the four pressure cells adjacent to each velocity.

Two nontrivial differences in the 3D implementation of the virtual node algorithm are detailed in [2]. First, the domain is initialized from the signed distance function  $\phi$  over a grid cell divided into tetrahedra to create a polyhedral approximation to the domain. Second, the method for evaluating integrals over the pressure cells also differs

slightly, combining use of the divergence theorem with a quadrature-based approach to evaluating integrals over triangles.

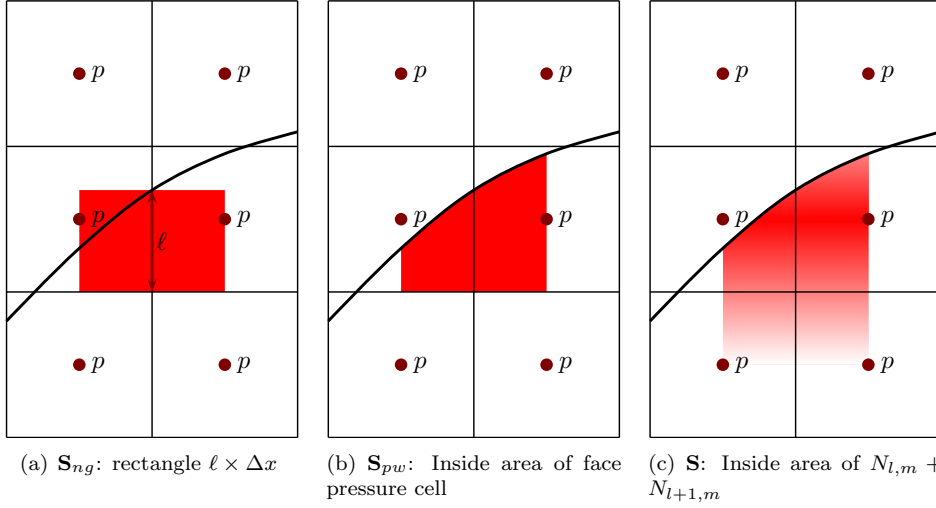


FIG. 6. Illustration of different constructions of the scaling matrix  $\mathbf{S}$ :  $\mathbf{S}_{ng}$  from [14], an alternative formulation  $\mathbf{S}_{pw}$  as described in Section 3.5, and the  $\mathbf{S}$  resulting from our condensed stencil factorization explained in Section 3.2.

**3.5. Related discretizations.** The discretization we have proposed is closely related to the discretization from [14]. Their discretization also factors nicely, as was demonstrated in [16]. Expressed in our notation, their discretization is identical except with a different  $\mathbf{S}$ , which we refer to as  $\mathbf{S}_{ng}$ . The value of  $\mathbf{S}_{ng}$  at a MAC face is proportional to the length of that face that is in the interior. Since  $\mathbf{S}$  scales as area (in 2D), we find that  $\mathbf{S}_{ng} = \ell \times \Delta x$  corresponds to the rectangular portion of the MAC face's associated pressure cell intersecting the face at the same point where the interface intersects the MAC face. This is illustrated in Figure 6(a).

While our pressure basis  $\{N_i\}$  is multilinear, an alternative definition of  $\mathbf{S}$  with a piecewise constant pressure basis would result in a different scaling matrix which we will call  $\mathbf{S}_{pw}$ . The entries of  $\mathbf{S}_{pw}$  correspond to the area of the MAC face pressure cell that is in the interior as shown in Figure 6(b). If the interface is well-resolved and does not slice out a corner from this pressure cell, the entries in  $\mathbf{S}_{ng}$  and  $\mathbf{S}_{pw}$  will be very close. The entries in  $\mathbf{S}$  are computed similarly to those in  $\mathbf{S}_{pw}$ , except that a weighted area is computed over a wider region as shown in Figure 6(c), effectively smoothing out the entries in  $\mathbf{S}$ .

Note that we use  $\mathbf{G}^T \mathbf{u}^{n+1} = \hat{\mathbf{G}}^T \mathbf{S} \mathbf{u}^{n+1} = 0$  as our incompressibility condition rather than the perhaps more intuitive central differenced condition  $\hat{\mathbf{G}}^T \mathbf{u}^{n+1} = 0$ . This is in line with the condition used by [14], where the incompressibility condition takes on an intuitive flux-based interpretation.

**4. Examples.** For each example, we take a vector field  $(u^*, v^*)$  or  $(u^*, v^*, w^*)$  to be the sum of an incompressible, divergence-free component and the gradient of a pressure. We apply our projection method to the vector field and compare the solution with the exact solution of the incompressible component. Each 2D example was discretized on a variety of  $N \times N$  grids for resolutions up to  $562^2$ . Our 3D

example was discretized on a series of  $N \times N \times N$  grids where  $N$  ranged up to 320. With each example we provide a graphic depicting the embedded domain, a plot of the computed pressure, and error plots for pressure and for the  $x$ -component of the incompressible velocity. In each case we perform linear regression analysis on the error data to obtain estimates of the order of accuracy for the pressure in the  $L^\infty$  norm and for velocities in the  $L^1$ ,  $L^2$ , and  $L^\infty$  norms. (Although we only present data for the  $x$ -component of velocities, the other velocities yield equivalent results.) Generally, our method produces pressures which are second-order accurate in  $L^\infty$  and velocities which are first-order in  $L^\infty$  but second in  $L^1$ ; however, projection of velocities with zero incompressible component is accurate to second-order in  $L^\infty$ .

**4.1. Two-dimensional projection example 1.** In our first two-dimensional example we project a gradient field velocity used in [14]:

$$\begin{aligned} u^* &= (x^2 - \pi x)(\pi y^2/2 - y^3/3) \\ v^* &= (y^2 - \pi y)(\pi x^2/2 - x^3/3) \end{aligned}$$

The embedded Neumann boundary  $\partial\Omega_n$  of the domain is bounded by the curve defined by:

$$\begin{aligned} t_0 &= .00132 \\ r_0 &= .02\sqrt{5} \\ r(t) &= .5 + .2 \sin(5t) \\ X(\theta) &= r_0 + r(\theta + t_0) \cos(\theta + t_0) \\ Y(\theta) &= r_0 + r(\theta + t_0) \sin(\theta + t_0). \end{aligned}$$

This curve is used in [1]. The domain is shown in Figure 7. The computed order of accuracy for pressure in the maximum norm is 1.990 and for velocity in the maximum norm is 1.876, see Figure 7. We computed second-order accurate velocities in  $L^\infty$  whenever projecting a fully irrotational, gradient velocity field.

**4.2. Two-dimensional projection example 2.** We next project the velocity field given by

$$\begin{aligned} u^* &= x + 2\pi \cos(2\pi x) \sin(2\pi y) \\ v^* &= -y + 2\pi \sin(2\pi x) \cos(2\pi y) \end{aligned}$$

The embedded Neumann boundary  $\partial\Omega_n$  of the domain is bounded by the curve defined by:

$$\begin{aligned} t_0 &= .45234 \\ \theta(t) &= t + \sin(4t) \\ r(t) &= .60125 + .24012 \cos(4t + \pi/2) \\ X(\theta) &= r(t + t_0) \cos(\theta(t + t_0)) \\ Y(\theta) &= r(t + t_0) \sin(\theta(t + t_0)), \end{aligned}$$

also used in [1], and the domain is shown in Figure 8. We computed the orders of accuracy to be 2.000 for pressure in  $L^\infty$  and .947 and 1.955 for velocity in  $L^\infty$  and  $L^1$  respectively.

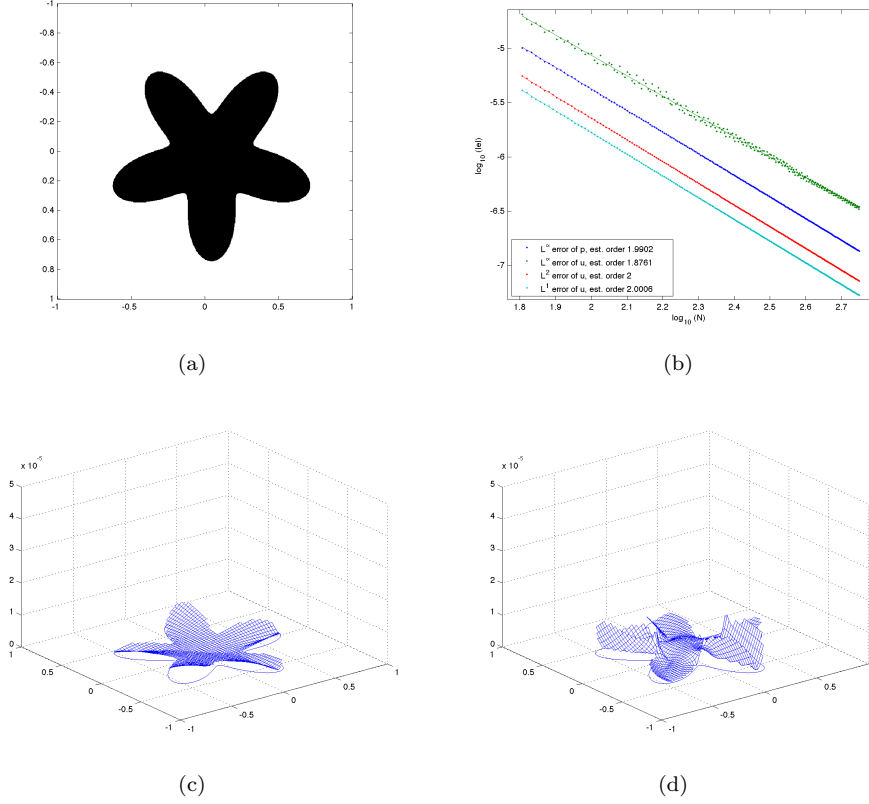


FIG. 7. Figures for Example 4.1: (a) geometry of  $\partial\Omega$  at  $N = 80$ , (b) convergence plot of the errors, and error plots of the pressure (c) and  $x$ -velocity (d) at  $N = 80$ .

**4.3. Two-dimensional projection example 3.** In this example our projected velocity is given as

$$\begin{aligned} u^* &= \sin(\pi x) \cos(\pi y) + (x+1)/((x+1)^2 + (y+4)^2) \\ v^* &= -\cos(\pi x) \sin(\pi y) + (y+4)/((x+1)^2 + (y+4)^2) \end{aligned}$$

The domain  $\Omega$  is defined to be the square  $[-1, 1] \times [-1, 1]$  with a circle removed of radius .7 and centered at the origin. Grid-aligned Neumann boundary conditions are defined at the edges of the square, while embedded Neumann boundary conditions are used at the circle boundary. See Figure 9. We estimated an order of accuracy of 1.988 for pressure in  $L^\infty$ , and .819 and 1.972 for velocity in  $L^\infty$  and  $L^1$  respectively.

**4.4. Three-dimensional projection example.** Finally, we present an application of our decomposition method in three dimensions. Here our projected velocity is given by

$$\begin{aligned} u^* &= .5 - y - \sin(\pi(x+y+z)) \\ v^* &= .5 - z - \sin(\pi(x+y+z)) \\ w^* &= .5 - x - \sin(\pi(x+y+z)) \end{aligned}$$

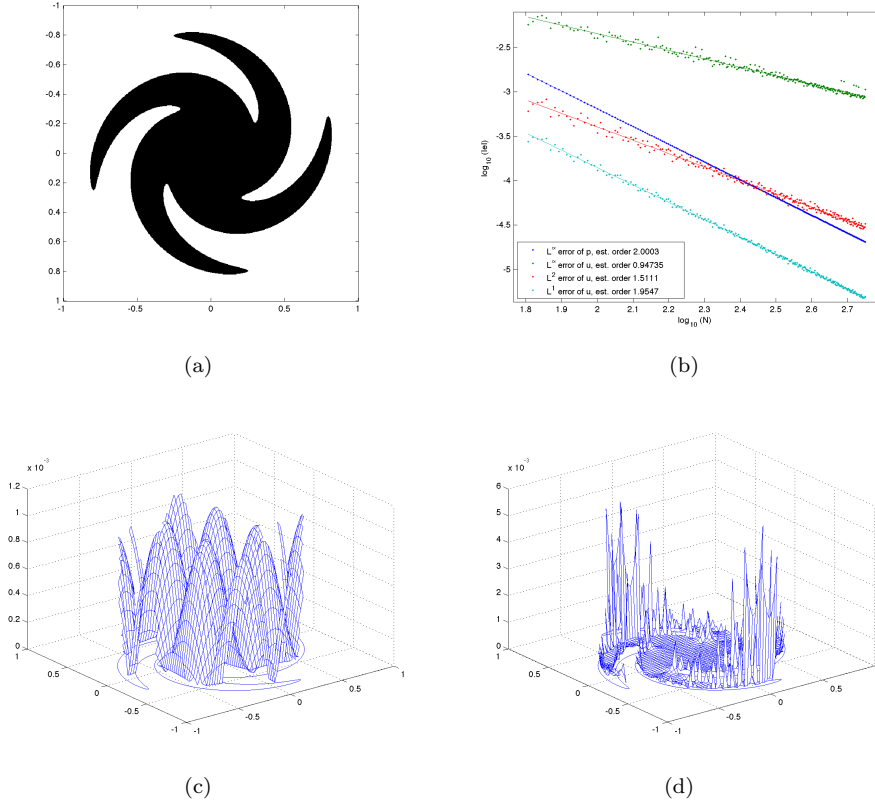


FIG. 8. Figures for Example 4.2: (a) geometry of  $\partial\Omega$  at  $N = 80$ , (b) convergence plot of the errors, and error plots of the pressure (c) and  $x$ -velocity (d) at  $N = 80$ .

and the domain, shown in Figure 10, is a sphere of radius .35 centered at  $(x, y, z) = (.4, .5, .5)$ . As in the two-dimensional case, we observe second-order accuracy (1.974) in  $L^\infty$  for the computed pressure, and order .993 in  $L^\infty$  and 1.986 in  $L^1$  for velocity.

**Acknowledgments.** This work was supported in part by grants from NSF (DMS-0502315, DMS-0652427, CCF-0830554, DGE-1144087), DOE (09-LR-04-116741-BERA), ONR (N000140310071, N000141010730, N000141210834), the Intel STCVisual Computing Grant (20112360), and the DOD (through the National Defense Science & Engineering Graduate Fellowship Program), as well as a gift from Disney Research.

#### REFERENCES

- [1] J. BEDROSSIAN, J. VON BRECHT, S. ZHU, E. SIFAKIS, AND J. TERAN, *A second order virtual node method for elliptic problems with interfaces and irregular domains*, J. Comp. Phys., 229 (2010), pp. 6405–6426.
- [2] J. HELLRUNG, L. WANG, E. SIFAKIS, AND J. TERAN, *A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions*, J. Comp. Phys., 231:4 (2012), pp. 2015–2048.
- [3] A. CHORIN, *A numerical method for solving incompressible viscous flow problems*, J. Comp. Phys., 2 (1967), pp. 12–26.

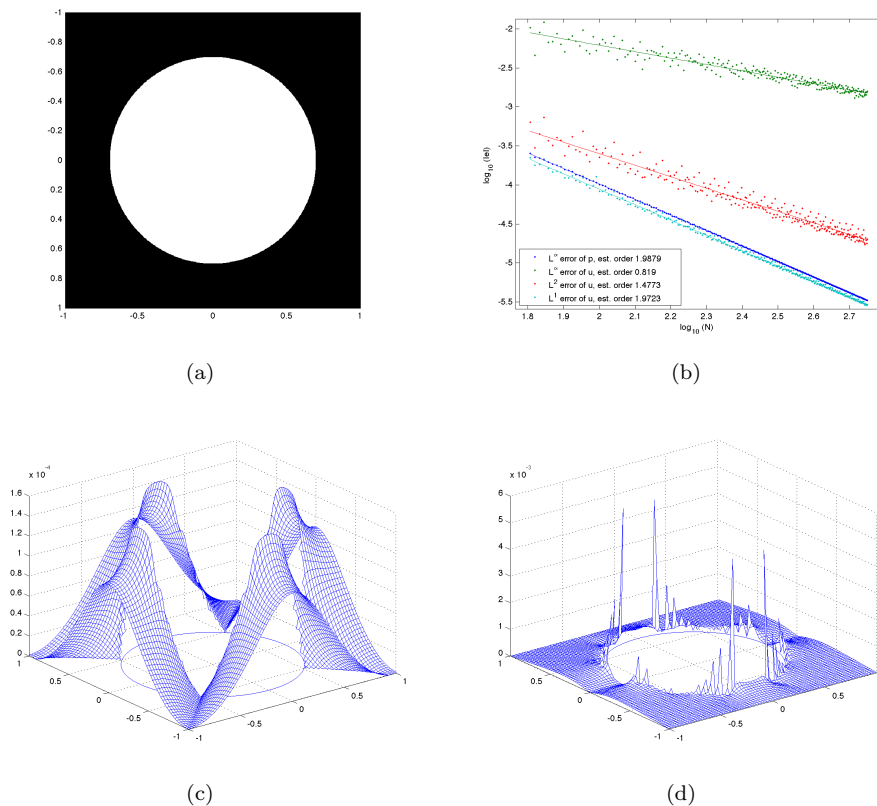


FIG. 9. Figures for Example 4.3: (a) geometry of  $\partial\Omega$  at  $N = 80$ , (b) convergence plot of the errors, and error plots of the pressure (c) and  $x$ -velocity (d) at  $N = 80$ .

- [4] F. HARLOW AND J. WELCH, *Numerical calculation of timedependent viscous incompressible flow of fluid with a free surface*, Phys. Fl., 8 (1965), pp. 2182–2189.
- [5] J. TERAN AND C. PESKIN, *Tether force constraints in stokes flow by the immersed boundary method on a periodic domain*, SIAM J. Sci. Comp., 31:5 (2009), pp. 3404–3416.
- [6] C. PESKIN, *The immersed boundary method*, Act. Num., 11 (2002), pp. 479–517.
- [7] R. LEVEQUE AND Z. LI, *The immersed interface method for elliptic equations with discontinuous coefficients and singular sources*, SIAM J. Numer. Anal., 31 (1994), pp. 1019–1044.
- [8] Z. LI AND K. ITO, *The Immersed Interface Method – Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*, SIAM Frontiers in Applied mathematics, 2006.
- [9] S. OSHER AND R. FEDKIW, *Level set methods and dynamic implicit surfaces*, Springer-Verlag, 2002.
- [10] F. GIBOU, R. FEDKIW, L. CHENG, AND M. KANG, *A second order accurate symmetric discretization of the poisson equation on irregular domains*, J. Comput. Phys., 176 (2002), pp. 205–227.
- [11] F. GIBOU AND R. FEDKIW, *A fourth order accurate discretization for the laplace and heat equations on arbitrary domains, with applications to the stefan problem*, J. Comp. Phys., 202 (2005), pp. 577–601.
- [12] D. ASSENCIO AND J. TERAN, *A second order virtual node algorithm for stokes flow problems with interfacial forces, discontinuous material parameters and irregular domains*, to appear, J. Comp. Phys.
- [13] J. DOLBOW AND A. DEVAN, *Enrichment of enhanced assumed strain approximations for representing strong discontinuities: addressing volumetric incompressibility and the discontinuous patch test*, Int. J. Num. Meth. Eng., 59:1 (2004), pp. 47–67.
- [14] Y. NG, C. MIN, AND F. GIBOU, *An efficient fluid-solid coupling algorithm for single-phase*

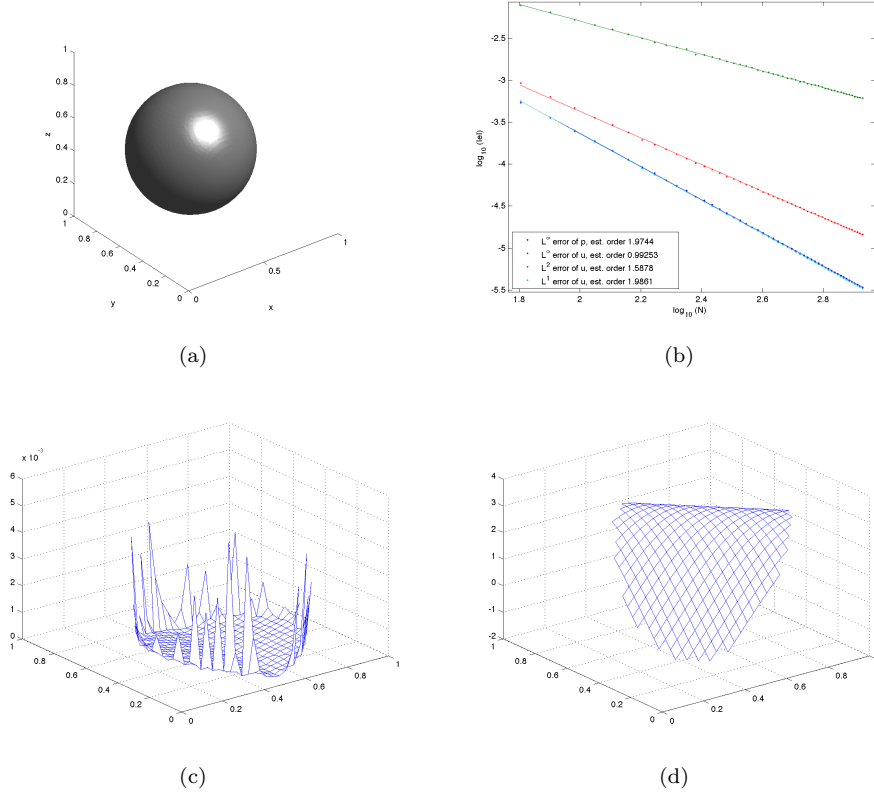


FIG. 10. Figures for Example 4.4: (a) geometry of  $\partial\Omega$  at  $N = 32$ , (b) convergence plot of the errors, and  $z$ -slices of the  $x$ -velocity error (c) and  $x$ -velocity (d) at  $N = 32$ .

- flows, J. Comp. Phys., 228:23 (2009), pp. 8807–8829.
- [15] F. GIBOU AND C. MIN, *Efficient symmetric positive definite second-order accurate monolithic solver for fluid/solid interactions*, J. Comp. Phys., 231:8 (2012), pp. 3246–3263.
- [16] C. SCHROEDER, Z. ZHENG, AND R. FEDKIW, *Implicit surface tension formulation with a lagrangian surface mesh on an eulerian simulation grid*, J. Comput. Phys., 231 (2012), pp. 2092–2115.

