Journal of Computational Physics 235 (2013) 683–700

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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

A Neumann–Neumann preconditioned iterative substructuring approach for computing solutions to Poisson's equation with prescribed jumps on an embedded boundary

Gregory H. Miller [a,*], Elbridge Gerry Puckett [b]

[a] Department of Chemical Engineering and Materials Science, University of California, One Shields Ave., Davis, CA 95616, United States
[b] Department of Mathematics, University of California, One Shields Ave., Davis, CA 95616, United States

ARTICLE INFO

Article history:
Received 21 December 2011
Received in revised form 9 July 2012
Accepted 17 October 2012
Available online 7 November 2012

Keywords:
Poisson problem
Heat equation
Jump conditions
Irregular domain
Finite volume method
Embedded boundary method
Multifluid problems
Surface tension
Domain decomposition
Iterative substructuring

ABSTRACT

In multifluid problems with surface tension the fluid pressure and its derivative are discontinuous at fluid interfaces. We present a Cartesian grid embedded boundary method for numerically resolving these discontinuities in which we use Neumann–Neumann preconditioned iterative substructuring to solve the governing equations. We validate this method by computing several well-known Poisson problems with discontinuous coefficients, and we compare its performance to an approach based on simple iteration. By analogy with the conjugate gradient method, we hypothesize that the scaling of the Neumann–Neumann preconditioned iterative substructuring is $\mathcal{O}(h^{-D}\ln h^{-1})$ where h is the cell size and $D = 2, 3$ is the dimensionality of the problem. In contrast, we show that the simple iterative procedure scales like $\mathcal{O}(h^{-(D+1)})$ and is slower by a factor of 4000 for a small (i.e., 64×64 cell) model calculation with physical parameters corresponding to a 1.5 mm air bubble in water. We present an analytical model to explain the scaling of this iterative procedure.

1. Introduction

This work is motivated by the goal of modeling the interface between two fluids in incompressible two-phase flow on uniform Cartesian grids in which the fluid interface is represented as an embedded boundary in those cells that intersect the interface. The equations governing incompressible two-phase flow are subject to jump conditions at the fluid interface in which both the pressure and its gradient are discontinuous while the velocity is continuous but has a discontinuous gradient. Consequently, differential operators in the momentum equations, such as the gradient and the Laplacian, become unbounded in regions abutting the interface. This problem is often addressed by regularizing or 'smoothing' the operator or its discretization. However such smoothing introduces an additional computational parameter that governs the size of the region over which the operator is regularized, which, if the computed solution is to converge to the true solution of the problem, must also go to zero as the grid size goes to zero. This approach also leads to a region in a neighborhood of the interface over which values of quantities defined on the interface are spread out or 'smeared' resulting in a representation of the interface that is not sharp.

* Corresponding author. Tel.: +1-530-752-7186; fax: +1-530-752-1031.
E-mail addresses: grgmiller@ucdavis.edu (G.H. Miller), egpuckett@ucdavis.edu (E.G. Puckett).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.10.023

Here we study an alternate approach in which we discretize each of the fluid domains $\Omega_1,\ \Omega_2$ separately so the governing equations are well-defined within each domain and the discontinuities or jumps in the dependent variables become boundary conditions on the boundary, $\partial\Omega_{12} = \partial\Omega_1 \cap \partial\Omega_2$, between the two domains. This approach is attractive for several reasons. It builds on well-established single domain computational algorithms for solving the governing partial differential equations (PDEs) in a single fluid domain. It also provides a sharp representation of the interface, thereby allowing better resolution of gradients normal to the interface leading to, for example, improved accuracy of algorithms for modeling the transport of quantities across interfaces, as well as more accurate models of quantities such as surfactants, that flow on the interface itself.

In this article we examine two approaches for coupling implicit Poisson solvers with jump boundary conditions in the context of a finite volume discretization of the governing PDEs on Cartesian grids, in which we use the embedded boundary approach to model irregular geometries such as a fluid interface. One of these approaches is the algorithm of Crockett et al. [6] (CCG), which synchronizes boundary conditions between the two domains with each V-cycle of a multigrid solver. In our approach, which is based on the so-called iterative substructuring method, we use an iterative Krylov space method to directly determine the boundary conditions on the interface $\partial\Omega_{12}$ between the two domains. We will refer to this latter approach as NNIS, for *Neumann–Neumann preconditioned iterative substructuring*. Both approaches use the same discretization of the governing PDE, the same boundary condition stencils, and identical multigrid solvers.

1.1. The governing equations

Incompressible two-phase flow is governed by the following momentum and continuity equations,

$$\frac{\partial}{\partial t}\mathbf{v}_\alpha + \mathbf{v}_\alpha \cdot \nabla\mathbf{v}_\alpha + \frac{1}{\rho_\alpha}\nabla P = \nu_\alpha \Delta \mathbf{v}_\alpha + \frac{1}{\rho_\alpha}\nabla\cdot\boldsymbol{\tau}_\alpha, \tag{1a}$$

$$\nabla\cdot\mathbf{v}_\alpha = 0, \tag{1b}$$

where we have used the subscript $\alpha = 1,2$ to distinguish between phase or fluid 1 and 2. In (1) $\mathbf{v}$ is the velocity, P the pressure, ρ the density, ν the kinematic Newtonian viscosity, and $\boldsymbol{\tau}$ is an extra stress that one can use describe, for example, non-Newtonian rheology such as viscoelasticity.

The surface tension between the two phases results in a force per unit area, $\mathbf{f}$, on the surface of the interface $\partial\Omega_{12}$, which is given by

$$\mathbf{f} = -\sigma\kappa\mathbf{n}_1, \tag{2}$$

where σ is a phenomenological coefficient, κ is the curvature of the interface, and $\mathbf{n}$ is a unit vector normal to $\partial\Omega_{12}$. We employ the convention that $\mathbf{n}_\alpha$ is the outward-directed unit normal on $\partial\Omega_\alpha$ and hence, $\mathbf{n}_2 = -\mathbf{n}_1$ on $\partial\Omega_{12}$.

Let ψ be a level set function such that the zero contour $\psi = 0$ coincides with the interface. Then the curvature κ of the interface is given by

$$\kappa = \nabla\cdot\left(\frac{\nabla\psi}{|\nabla\psi|}\right). \tag{3}$$

We choose the sign convention $\psi < 0$ in Ω_1, which fixes a sign convention for σ.

For the problem (1), (2) with σ constant (i.e., no Marangoni forces), the boundary conditions on the pressure P are given by,

$$[\![P]\!] = [\![\tau_{nn}]\!] + \mathbf{n}_1\cdot\mathbf{f}. \tag{4a}$$

$$\left[\!\!\left[\frac{1}{\rho}(\mathbf{n}_1\cdot\nabla)P\right]\!\!\right] = \left[\!\!\left[\frac{1}{\rho}\mathbf{n}_1\cdot(\nabla\cdot\boldsymbol{\tau})\right]\!\!\right] + [\![\mathbf{n}_1\cdot\nu\Delta\mathbf{v}]\!], \tag{4b}$$

where the notation $[\![\phi]\!]$ denotes the jump $\phi_2 - \phi_1$ on $\partial\Omega_{12}$. The boundary conditions on the velocity are

$$[\![\mathbf{v}]\!] = 0, \tag{5a}$$

$$[\![(\mathbf{n}_1\cdot\nabla)(\mathbf{v}\cdot\mathbf{n})]\!] = 0, \tag{5b}$$

$$[\![\rho\nu(\mathbf{n}_1\cdot\nabla)\mathbf{v}]\!] = -\mathbf{n}_1\cdot[\![\boldsymbol{\tau}]\!] + [\![\tau_{nn}]\!]\mathbf{n}_1. \tag{5c}$$

A Hodge decomposition (e.g., see [5]) of the momentum Eq. (1a) yields the governing equation for the pressure in each phase:

$$\Delta P = \nabla\cdot(\nabla\cdot\boldsymbol{\tau} + \rho[\nu\Delta\mathbf{v} - \mathbf{v}_t - \mathbf{v}\cdot\nabla\mathbf{v}]). \tag{6}$$

We solve (6) on Ω_1 and Ω_2 separately, with the boundary conditions (4a,b) coupling the two single-phase solutions (e.g., see Fig. 1). Although the domains are time dependent, discretizations of the time-dependent problem can be based on fixed-time solutions of (6) (e.g., see [23]). Similarly, with operator splitting the viscous operator in (1a) can be discretized using a sequence of Helmholtz operators at discrete time intervals with jump conditions (5) (see [23]).

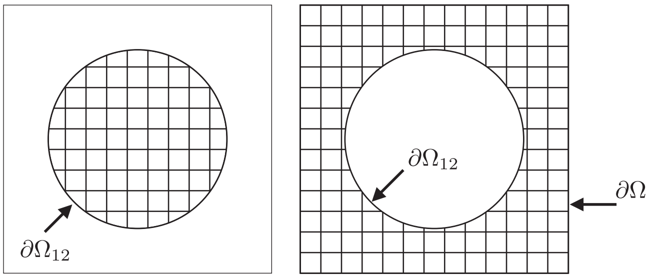

Fig. 1. Decomposition of domains Ω_1 (left) and Ω_2 (right) into two disjoint sets, each discretized with the same Cartesian grid using the embedded boundary approach to irregular geometry.

We note that a force analogous to surface tension arises in the study of thin films, including lipid bilayers suspended in a fluid [13]. Other related problems include determining the electrostatic potential associated with a surface charge or surface dipole density [28] and heat transfer in heterogeneous materials [2].

Prior work involving modeling a force concentrated on the interface between two fluids includes the following. In the immersed boundary method Peskin [24,25] represented tension in the wall of a human heart as a singular body force by convolving it with a discretization of Dirac delta functions located at points on the wall. Brackbill et al. [1] expressed the surface force due to the surface tension between two fluids as a regularized singular body force using the gradient of a color function. Sussman et al. [31] and Chang et al. [4] used a one-dimensional version of Peskin's discretization of the Dirac delta function coupled to a level set method to model surface tension in two phase flow. Sussman and Puckett [30] combined the level set and volume of fluid approaches, again using a discrete Dirac delta function to express the force due to surface tension as a regularized body force in the momentum equations. In the immersed interface method, LeVeque and Li [18] avoid the use of a discrete delta function δ entirely by inserting the value of the jump in the dependent variables at the interface in a modified finite difference stencil in a neighborhood of the interface. All of these approaches allow a single discretization of the Laplacian in (6) over the combined domain $\Omega = \Omega_1 \cup \Omega_2$ with modifications to the right-hand side of the momentum equations to enforce the boundary conditions at the fluid interface.

1.2. The Poisson problem

We generalize Eq. (6) with boundary conditions given by (4) by writing the generic two-phase flow problem in the following way,

$$\mathcal{L}_1 u_1 = f_1 \quad \text{on} \quad \Omega_1, \tag{7a}$$

$$\mathcal{L}_2 u_2 = f_2 \quad \text{on} \quad \Omega_2, \tag{7b}$$

$$[\![u]\!] = j_D \quad \text{on} \quad \partial\Omega_{12}, \tag{7c}$$

$$\mathbf{n}_1 \cdot [\![\mathcal{D}\nabla u]\!] = j_N \quad \text{on} \quad \partial\Omega_{12}, \tag{7d}$$

together with some boundary conditions on $\partial\Omega$, the boundary of the composite domain $\Omega \equiv \Omega_1 \cup \Omega_2$. In (7) the symbol $\mathcal{L}$ denotes a linear differential operator, the domains Ω_1, Ω_2 are a nonoverlapping partition of Ω (i.e, $\Omega_1 \cap \Omega_2 = \emptyset$) and $\mathcal{D}$ is a piecewise constant coefficient. We will refer to $-\mathcal{D}\nabla u$ as a *flux*.

We associate with the discretization of the operator $\mathcal{L}$ a matrix representation **A**, which operates on points within the domain and on points located on the domain boundary. Since the grid function **u** is double-valued on the interface $\partial\Omega_{12}$, we distinguish values $\mathbf{u}_{1b}$ associated with the Ω_1 side of the boundary and values $\mathbf{u}_{2b}$ associated with the Ω_2 side. This leads to the following block-partitioned system of equations:

$$\left(\begin{array}{c|c|c|c} \mathbf{A}_{11} & & \mathbf{A}_{1b} & \\ \hline & \mathbf{A}_{22} & & \mathbf{A}_{2b} \\ \hline & & -\mathbf{I} & \mathbf{I} \\ \hline \mathbf{G}_{11} & \mathbf{G}_{22} & \mathbf{G}_{1b} & \mathbf{G}_{2b} \end{array}\right) \left(\begin{array}{c} \mathbf{u}_1 \\ \hline \mathbf{u}_2 \\ \hline \mathbf{u}_{1b} \\ \hline \mathbf{u}_{2b} \end{array}\right) = \left(\begin{array}{c} \mathbf{f}_1 \\ \hline \mathbf{f}_2 \\ \hline \mathbf{j}_D \\ \hline \mathbf{j}_N \end{array}\right). \tag{8}$$

In (8) the matrix discretization of the operator $\mathcal{L}_1$ is partitioned into a matrix $\mathbf{A}_{11}$ operating on points interior to the domain and a matrix $\mathbf{A}_{1b}$ operating on the points on the boundary of the domain. The first block row of (8) corresponds to (7a), and the second is a similar partition of the discretization of $\mathcal{L}_2$ corresponding to (7b). The third block row is a discretization of (7c). In the discretization of (7d), the flux operators $-\mathcal{D}_\alpha \mathbf{n}_\alpha \cdot \nabla$ are given by the matrices $\mathbf{G}_{\alpha\alpha}$ and $\mathbf{G}_{\alpha b}$, which operate on interior and boundary points respectively. Since $\mathbf{n}_2 = -\mathbf{n}_1$ at a point on the interface, the explicit sign difference seen in the third block row is implicit in the definitions of the matrices $\mathbf{G}_{\alpha\alpha}$ and $\mathbf{G}_{\alpha b}$ in the fourth block row.

Recently, Crockett et al. [6] presented a solution to the multimaterial Poisson and heat problems that is based on approximating solutions of (7) by rearranging (8) in the following way,

$$\left(\begin{array}{c|c|c|c} \mathbf{A}_{11} & & \mathbf{A}_{1b} & \\ \hline & \mathbf{A}_{22} & & \mathbf{A}_{2b} \\ \hline \mathbf{G}_{11} & \mathbf{G}_{22} & \mathbf{G}_{1b}+\mathbf{G}_{2b} & \\ \hline \mathbf{G}_{11} & \mathbf{G}_{22} & & \mathbf{G}_{1b}+\mathbf{G}_{2b} \end{array}\right)\left(\begin{array}{c} \mathbf{u}_1 \\ \hline \mathbf{u}_2 \\ \hline \mathbf{u}_{1b} \\ \hline \mathbf{u}_{2b} \end{array}\right) = \left(\begin{array}{c} \mathbf{f}_1 \\ \hline \mathbf{f}_2 \\ \hline \mathbf{j}_N - \mathbf{G}_{2b}\mathbf{j}_D \\ \hline \mathbf{j}_N + \mathbf{G}_{1b}\mathbf{j}_D \end{array}\right), \tag{9}$$

which they solve using the following iterative procedure:

$$\left(\begin{array}{c|c} \mathbf{A}_{11} & \\ \hline & \mathbf{A}_{22} \end{array}\right)\left(\begin{array}{c} \mathbf{u}_1 \\ \hline \mathbf{u}_2 \end{array}\right)^{(k+1)} = \left(\begin{array}{c} \mathbf{f}_1 - \mathbf{A}_{1b}\mathbf{u}_{1b}^{(k)} \\ \hline \mathbf{f}_2 - \mathbf{A}_{2b}\mathbf{u}_{2b}^{(k)} \end{array}\right), \tag{10a}$$

$$\left(\begin{array}{c|c} \mathbf{G}_{1b}+\mathbf{G}_{2b} & \\ \hline & \mathbf{G}_{1b}+\mathbf{G}_{2b} \end{array}\right)\left(\begin{array}{c} \mathbf{u}_{1b} \\ \hline \mathbf{u}_{2b} \end{array}\right)^{(k+1)} = \left(\begin{array}{c} \mathbf{j}_N - \mathbf{G}_{2b}\mathbf{j}_D - \mathbf{G}_{11}\mathbf{u}_1^{(k+1)} - \mathbf{G}_{22}\mathbf{u}_2^{(k+1)} \\ \hline \mathbf{j}_N + \mathbf{G}_{1b}\mathbf{j}_D - \mathbf{G}_{11}\mathbf{u}_1^{(k+1)} - \mathbf{G}_{22}\mathbf{u}_2^{(k+1)} \end{array}\right). \tag{10b}$$

The first step, namely, (10a), is the solution of the following Dirichlet problems on each domain separately,

$$\mathcal{L}_\alpha u_\alpha = f_\alpha \quad \text{on} \quad \Omega_\alpha, \tag{11a}$$

$$u_\alpha \quad = u_{\alpha b} \ \text{on} \quad \partial\Omega_{12} \tag{11b}$$

for $\alpha = 1, 2$, where the boundary conditions $\mathbf{u}_{\alpha b}$ are determined at the prior (i.e., the kth) iteration. The second step (10b) reevaluates the boundary conditions by solving simultaneous equations for the jump conditions and discretizations of the flux terms. In Crockett et al.'s implementation of this iterative procedure, they use multigrid to solve the block matrix equations in (10a), but in a manner that is incomplete; they use only one multigrid V-cycle per fluid per iteration. They demonstrate second-order accuracy for their method applied to complicated heat problems. However, they do not consider the speed of convergence of the iterative procedure in (10), nor show its application to the Poisson problem.

The system (8) also facilitates the partitioning of the domain $\Omega = \Omega_1 \cup \Omega_2$ as a means to achieve parallelism for large problems. This approach is widely used in the finite element community and there is a large literature describing it with several excellent comprehensive reviews [16,27,33]. In this setting, it is customary for one to solve (8) by block-triangularization of the matrix, in order to obtain the solution by back substitution:

$$\begin{aligned}\left[-\mathbf{G}_{11}\mathbf{A}_{11}^{-1}\mathbf{A}_{1b} + \mathbf{G}_{1b} - \mathbf{G}_{22}\mathbf{A}_{22}^{-1}\mathbf{A}_{2b} + \mathbf{G}_{2b}\right]\bar{\mathbf{u}} = \mathbf{j}_N - \mathbf{G}_{11}\mathbf{A}_{11}^{-1}\mathbf{f}_1 - \frac{1}{2}\mathbf{G}_{11}\mathbf{A}_{11}^{-1}\mathbf{A}_{1b}\mathbf{j}_D + \frac{1}{2}\mathbf{G}_{1b}\mathbf{j}_D \\ - \mathbf{G}_{22}\mathbf{A}_{22}^{-1}\mathbf{f}_2 + \frac{1}{2}\mathbf{G}_{22}\mathbf{A}_{22}^{-1}\mathbf{A}_{2b}\mathbf{j}_D - \frac{1}{2}\mathbf{G}_{2b}\mathbf{j}_D\end{aligned} \tag{12a}$$

$$\mathbf{u}_{1b} = \bar{\mathbf{u}} - \frac{1}{2}\mathbf{j}_D, \tag{12b}$$

$$\mathbf{u}_{2b} = \bar{\mathbf{u}} + \frac{1}{2}\mathbf{j}_D, \tag{12c}$$

$$\mathbf{A}_{22}\mathbf{u}_2 = \mathbf{f}_2 - \mathbf{A}_{2b}\mathbf{u}_{2b}, \tag{12d}$$

$$\mathbf{A}_{11}\mathbf{u}_1 = \mathbf{f}_1 - \mathbf{A}_{1b}\mathbf{u}_{1b}. \tag{12e}$$

Eq. (12a) is the most difficult to solve. However it exists on the codimension-one space of interface variables, which offers some computational advantages. The matrix on the left-hand side of (12a) is called the capacitance matrix, **C**. In the finite element community obtaining the solution of (7) by solving the system in (12) is known as the iterative substructure method and researchers often use a preconditioned conjugate gradient method to solve the system in (12). Recent developments in this field include so-called optimal Schwarz methods, where boundary operators are chosen to accelerate the iterative solution [8,9,20].

However the matrix **C** associated with our finite volume discretization of (1) is not symmetric positive definite. Therefore, we use the preconditioned Bi-CGSTAB method [35] with a multigrid preconditioner instead to solve the system in (12). Since our implementation of the numerical method described in this article is done is such a way that the capacitance matrix **C** is never explicitly produced (the values of the discretized quantities nodal points of each are simply stored in a list), this seems to be the most convenient method of preconditioning. We note, however, that over the last decade and a half there has been considerable work to develop domain decomposition based $\mathcal{H}$-LU factorizations based on so-called $\mathcal{H}$-Matrices (e.g., see [10]) for use as a preconditioner in the iterative solution of discretizations of various PDEs such as the convection–diffusion equation [11,12]. An alternate discretization from the one we have chosen may benefit from the use of such a domain decomposition based $\mathcal{H}$-LU factorization based preconditioner.

1.3. An overview of this article

The remainder of this article is organized as follows. In Section 2 we describe an embedded boundary finite volume implementation of the domain decomposition problem (12) in two and three space dimensions, and with adaptive grids. In Section 3 we compare the performance of NNIS with CCG. In a problem equivalent to the time-independent calculation of pressure associated with an air bubble in water, we demonstrate that NNIS is faster than CCG by a factor of 4000 or more. In Section 4 we describe the scaling behavior for both approaches. For simple iterations we introduce a simple analytical model to explain the convergence properties, the sensitivity to parameters, and the difference in convergence behavior for the heat and Poisson problems. In Section 5 we also apply NNIS to solve some discontinuous-coefficient Poisson problems. Conclusions are presented in Section 6.

2. The algorithms

We discretize space with uniform Cartesian cells of length h overlain by the boundary, as depicted in Fig. 1. We discretize the governing PDEs using a finite volume approach in which quadrature points on the centroid of the embedded boundary are used to incorporate the boundary conditions. Discretizations of the linear operators (e.g., the Poisson and heat operators) using the embedded boundary finite volume discretizations are described in [14], and modifications to permit adaptive mesh refinement (AMR) are described by Martin and Cartwright in [19].

In what follows we will write Eq. (12a) as $\mathbf{C}\bar{\mathbf{u}} = \mathbf{b}$. As we mentioned in Section 1.2 the matrix $\mathbf{C}$ is not symmetric positive definite and therefore we use the preconditioned Bi-CGSTAB method [35]. In order to implement the Bi-CGSTAB algorithm one must first determine $\mathbf{b}$, evaluate $\mathbf{C}\bar{\mathbf{u}}$ and evaluate a preconditioner.

For $\mathbf{b}$:

I.1 Evaluate $\tilde{\mathbf{u}}_\alpha = \mathbf{A}_{\alpha\alpha}^{-1}(\mathbf{f}_\alpha \pm \mathbf{A}_{\alpha b}\mathbf{j}_D/2)$, with $+, -$ for $\alpha = 1, 2$, respectively. These are solutions of Dirichlet problems,

$$\mathcal{L}_\alpha \tilde{\mathbf{u}}_\alpha = \mathbf{f}_\alpha \qquad \Omega_\alpha, \tag{13a}$$

$$\tilde{\mathbf{u}}_1 \quad = -\frac{1}{2}\mathbf{j}_D \quad \partial\Omega_{12}, \tag{13b}$$

$$\tilde{\mathbf{u}}_2 \quad = +\frac{1}{2}\mathbf{j}_D \quad \partial\Omega_{12}. \tag{13c}$$

The boundary conditions on $\partial\Omega \cap \partial\Omega_\alpha$ are those associated with (7).
I.2Cal-
culate the $\partial\Omega_{12}$ boundary flux for domain α, $\tilde{\mathbf{g}}_\alpha = -\mathcal{D}_\alpha \mathbf{n}_\alpha \cdot \nabla \tilde{\mathbf{u}}_\alpha$:

$$\tilde{\mathbf{g}}_1 = \mathbf{G}_{11}\tilde{\mathbf{u}}_1 - \mathbf{G}_{1b}\frac{1}{2}\mathbf{j}_D, (14a)$$

$$\tilde{\mathbf{g}}_2 = \mathbf{G}_{22}\tilde{\mathbf{u}}_2 + \mathbf{G}_{2b}\frac{1}{2}\mathbf{j}_D. (14b)$$

I.3 Finally,

$$\mathbf{b} = \mathbf{j}_N - \tilde{\mathbf{g}}_1 - \tilde{\mathbf{g}}_2. \tag{15}$$

For the evaluation of $\mathbf{C}\bar{\mathbf{u}}$:

II.1 Evaluate $\hat{\mathbf{u}}_\alpha = -\mathbf{A}_{\alpha\alpha}^{-1}\mathbf{A}_{\alpha b}\bar{\mathbf{u}}$. These are solutions to the homogeneous PDE (7a) with inhomogeneous Dirichlet boundary condition $\bar{\mathbf{u}}$ on $\partial\Omega_{12}$,

$$\mathcal{L}_\alpha \hat{\mathbf{u}}_\alpha = 0 \quad \Omega_\alpha, \tag{16a}$$

$$\hat{\mathbf{u}}_\alpha \quad = \bar{\mathbf{u}} \quad \partial\Omega_{12} \tag{16b}$$

and with homogeneous boundary conditions on $\partial\Omega \cap \partial\Omega_\alpha$. For instance, use homogeneous Dirichlet conditions where a Dirichlet condition is prescribed in (7), and a homogeneous Neumann condition where a Neumann condition is prescribed.

II.2 Calculate the corresponding flux $\hat{\mathbf{g}}_\alpha = -\mathcal{D}_\alpha \mathbf{n}_\alpha \cdot \nabla \hat{\mathbf{u}}_\alpha$,

$$\hat{\mathbf{g}}_\alpha = \mathbf{G}_{\alpha\alpha}\hat{\mathbf{u}}_\alpha + \mathbf{G}_{\alpha b}\bar{\mathbf{u}}. \tag{17}$$

II.3 Then,

$$\mathbf{C}\bar{\mathbf{u}} = \hat{\mathbf{g}}_1 + \hat{\mathbf{g}}_2. \tag{18}$$

Note that the principle of superposition gives

$$\mathbf{u}_\alpha = \tilde{\mathbf{u}}_\alpha + \hat{\mathbf{u}}_\alpha, \tag{19}$$

as the final solution to (7) on the interiors of domains Ω_1, Ω_2 when $\hat{\mathbf{u}}$ results from algorithm **II**, the application of operator $\mathbf{C}$, to the converged solution $\bar{\mathbf{u}}$. This calculation is not a side effect of the Bi-CGSTAB algorithm, and must be done separately.

In order to precondition Bi-CGSTAB one needs to apply a matrix $\mathbf{K}$ that approximates $\mathbf{C}^{-1}$. Writing $\mathbf{C} = \mathbf{S}_1 + \mathbf{S}_2$; the sum of linear operators applied to domains Ω_1 and Ω_2, respectively, one model for $\mathbf{K}$ is a weighted sum of $\mathbf{S}_1^{-1}$ and $\mathbf{S}_2^{-1}$. This is known as the Neumann–Neumann preconditioner. It takes its name from the observation that $\mathbf{S}_\alpha$ is a mapping of boundary values to boundary fluxes. Consequently, $\mathbf{S}_\alpha^{-1}$ is the mapping of boundary fluxes to boundary values – a Neumann problem. To evaluate $\mathbf{Kp}$ with weighting factors $w_1,\ w_2$:

III.1 Evaluate $\check{\mathbf{u}}_\alpha$ as the solution to the homogeneous PDEs with inhomogeneous Neumann boundary conditions on $\partial\Omega_{12}$,

$$\mathcal{L}_\alpha \check{\mathbf{u}}_\alpha = 0 \quad \Omega_\alpha, \tag{20a}$$

$$-\mathcal{D}_\alpha \mathbf{n}_\alpha \cdot \nabla \check{\mathbf{u}}_\alpha = \mathbf{p} \quad \partial\Omega_{12} \tag{20b}$$

and with homogeneous boundary conditions on $\partial\Omega \cap \partial\Omega_\alpha$ as in step II.1.

III.2 Interpolate $\check{\mathbf{u}}_\alpha$ to the boundary to obtain a value $\check{\mathbf{u}}_{\alpha b}$ consistent with the solution $\check{\mathbf{u}}_\alpha$ and the given flux, cf. (17),

$$\check{\mathbf{u}}_{\alpha b} = \mathbf{G}_{\alpha b}^{-1}(\mathbf{p} - \mathbf{G}_{\alpha\alpha}\check{\mathbf{u}}_\alpha). \tag{21}$$

Note that $\mathbf{G}_{\alpha b}$ is an invertible diagonal matrix.

III.3 Then,

$$\mathbf{Kp} = w_1 \check{\mathbf{u}}_{1b} + w_2 \check{\mathbf{u}}_{2b}. \tag{22}$$

The Poisson problems with Neumann boundary conditions have a null space, so operations (20) require some care. The approach we take is to use the generalized Helmholtz operator

$$\mathcal{L} = \epsilon \mathbf{I} + \Delta, \tag{23}$$

in place of the Laplacian, for the preconditioner only, with some small negative value of ϵ.

In the present implementation of this method, we use AMR with the following restrictions: (i) on each domain, the interfaces $\partial\Omega_\alpha \setminus \partial\Omega$ are discretized with the finest resolution; (ii) the finest resolution on Ω_1 is the finest resolution on Ω_2; and (iii) the width of the refined zone at interfaces is sufficient to support the flux operator $\mathbf{G}$. With these assumptions, boundary variable $\bar{\mathbf{u}}$ of the iterative substructure method may be computed everywhere on the finest domain, and need be computed only on the finest domain. To support operations on the AMR hierarchy, boundary values at the finest level are restricted to the coarser levels using boundary area weighted averaging.

The flux operator $\mathbf{G}$ may be discretized using the gradient stencil described by Johansen and Colella [14], as implemented by Crockett et al. [6]. An alternative approach that is stable and second-order accurate is to use a stencil based on least squares. Let $\mathbf{x}_b$ be a point on the interface $\partial\Omega_{12}$ that approximates the centroid of the intersection of the interface with the boundary of cell $\mathbf{i}_b$. The center of cell $\mathbf{i}$ is $\mathbf{x_i} = h(\mathbf{i} + \frac{1}{2}\mathbf{1})$ where $\mathbf{1}$ is the vector of ones, and h is the grid size at the finest refinement level.

For all cells in the neighborhood of $\mathbf{i}_b$, a truncated Taylor series expansion can be constructed. Centered at the interface point $\mathbf{x}_b$,

$$\sum_{|\mathbf{p}|=1}^{P} \frac{1}{\mathbf{p}!} \frac{\partial^{\mathbf{p}} u}{\partial \mathbf{x}^{\mathbf{p}}} (\mathbf{x_i} - \mathbf{x}_b)^{\mathbf{p}} = u_{\mathbf{i}} - u_b + \mathcal{O}(h^{P+1}). \tag{24}$$

Here, we use a multiindex notation: D is the dimensionality of the problem, $\mathbf{p}$ is a vector of nonnegative integers in $\mathbb{N}_0^D$, $\mathbf{x}^{\mathbf{p}}$ is $\prod_i^D x_i^{p_i}$, $\mathbf{p}!$ is $\prod_i^D p_i!$, $P = \sum_i^D p_i$, and $\partial^{\mathbf{p}}/\partial\mathbf{x}^{\mathbf{p}} = \partial^P/(\partial x_1^{p_1}\partial x_2^{p_2}\ldots\partial x_D^{p_D})$.

System (24) can be written $\mathbf{Ad} = \mathbf{u} - u_b\mathbf{1}$, where $\mathbf{d}$ is the vector of partial derivatives, and $\mathbf{A}$ contains the coefficients $(\mathbf{x} - \mathbf{x}_b)^{\mathbf{p}}$. We require that $\mathbf{A}$ is $n \times m$, with $n \geqslant m \geqslant D$. That is, the system is overdetermined and provides the first derivatives of u. Matrix $\mathbf{A}$ has Householder decomposition $\mathbf{A} = \mathbf{QR}$, with unitary $\mathbf{Q}$ and right triangular $\mathbf{R}$, so

$$\mathbf{Rd} = \mathbf{Q}^T\mathbf{u} - u_b\mathbf{Q}^T\mathbf{1}. \tag{25}$$

This system is solved by back substitution. Symbolically, we compute

$$s_\alpha = -\mathcal{D}_\alpha \mathbf{n}_\alpha^T (\mathbf{I}|\mathbf{0}) \mathbf{R}^{-1} \mathbf{Q}^T \mathbf{u}, \tag{26a}$$

$$r_\alpha = \mathcal{D}_\alpha \mathbf{n}_\alpha^T (\mathbf{I}|\mathbf{0}) \mathbf{R}^{-1} \mathbf{Q}^T \mathbf{1}. \tag{26b}$$

The block-partitioned matrix $(\mathbf{I}|\mathbf{0})$ selects from the solution $(\mathbf{R}^{-1}\mathbf{Q}^T\ldots)$ the D components corresponding to first derivatives centered at $\mathbf{x}_b$. Then,

$$g_\alpha = s_\alpha + r_\alpha u_{\alpha b} \tag{27}$$

is the calculation of the boundary flux occurring in Eqs. (14) and (17), and

$$u_{\alpha b} = \frac{g_\alpha - s_\alpha}{r_\alpha} \tag{28}$$

is the inverse occurring in (21).

To obtain slopes with second-order accuracy, we use $P = 2$ in the Taylor expansion [34]. The neighborhood sampled for the least squares problem is the 7^D block of cells centered at $\mathbf{i}_b$. Typically, $\sim 40\%$ of these are outside domain Ω_α, and are therefore not included. Also excluded is cell $\mathbf{i}_b$. For stability, it is necessary that r_α be nonzero (matrix $\mathbf{G}_{\alpha b}$ is the diagonal matrix with elements r_α). To understand why r_α may be small, consider the case $P = 1$. Then r_α is proportional to the mean distance of the sampled points from $\mathbf{x}_b$ in the direction $\mathbf{n}$. So, to make $|r_\alpha|$ be $\geqslant \mathcal{O}(h)$ we select only points in the neighborhood for which $\mathbf{n} \cdot (\mathbf{i} - \mathbf{i}_b)$ is negative. Then the mean is comprised of terms with uniform sign, and the method is stable.

3. Performance of NNIS in relation to a simple iterative procedure

3.1. Two space dimensions

As an example, we consider the following problem. In two space dimensions

$$\begin{aligned}
&\Omega : [-1/10, 1/10]^2, \\
&\Omega_1 : \left\{ r, \theta | r \leqslant \left(1 + \frac{1}{2}\cos(2\theta)\right) \Big/ 20 \right\}, \\
&\Delta u = 0, \\
&[\![u]\!] = \sigma\kappa \quad \partial\Omega_{12}, \\
&\mathbf{n}_1 \cdot \left[\!\!\left[\frac{1}{\rho}\nabla u\right]\!\!\right] = 0 \quad \partial\Omega_{12}, \\
&u = 0 \quad \partial\Omega,
\end{aligned} \tag{29}$$

with parameters $\sigma = 72.8$, $\rho_1 = 1.2 \times 10^{-3}$ (interior), and $\rho_2 = 1.0$ (exterior). With CGS units, this corresponds to the physical problem of the (time-independent) determination of pressure due to the surface tension of a cardioid air bubble in water, with a maximum diameter of 1.5 mm.

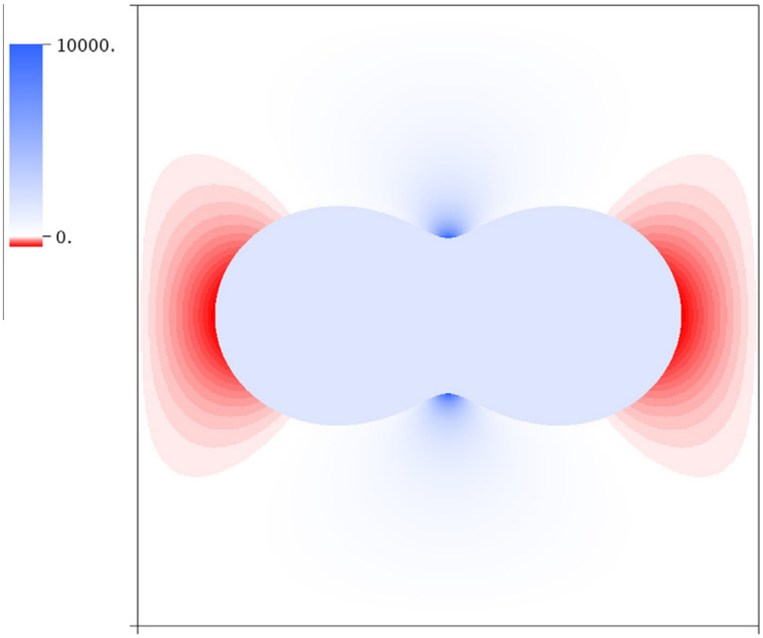

Fig. 2. Solution to problem (29).

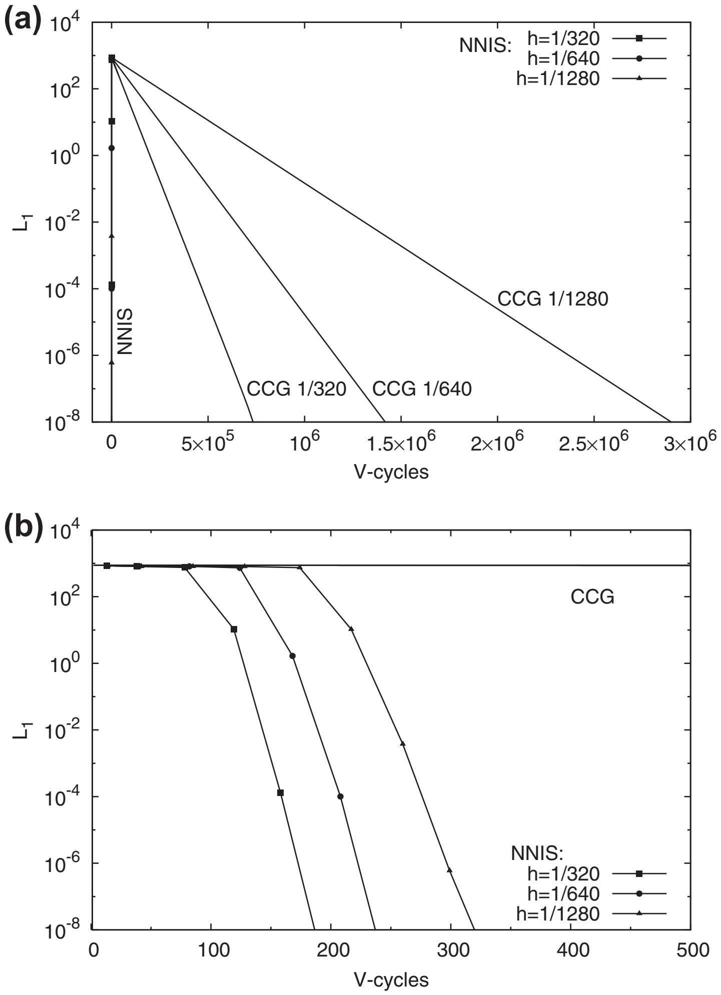

Fig. 3. Convergence of the problem in (29) with parameters corresponding to an air bubble in water, using the NNIS and CCG algorithms. For NNIS, symbols mark the end of a Bi-CGSTAB iteration. The ordinate (log scale) is the normalized L_1 norm of the solution error (30), and the abscissa (linear scale) is the total number of multigrid V-cycles: a measure of operation count. The two frames contain the same information on different scales: (a) on a scale where CCG convergence is evident, NNIS convergence is indistinguishable from the y axis; (b) on a scale where NNIS convergence is resolved CCG appears unchanging.

A solution in 2D is shown in Fig. 2. To stabilize the Neumann problems we use $\epsilon = -10^{-3}/h^2$ in (23). In adaptive calculations, this ϵ calculation uses the smallest h in the hierarchy. In elasticity problems Le Tallec et al. [17] suggest the weighting factors in the preconditioner (22) be proportional to the material stiffness. For the Poisson problem the analogous construction would weigh by $w_\alpha \propto 1/\rho_\alpha$ which gives $w_2 \ll w_1$. By taking $w_2 = 0$ and $w_1 = 1/2$ we approximate stiffness weighing and can omit altogether the Neumann problem in the water domain. This improves performance by a factor of 4, approximately, relative to the choice $w_1 = w_2 = 1/4$. Fig. 3 shows the convergence of the Neumann–Neumann preconditioned iterative substructure method estimated comparing the evolving solution vector against the final converged result. The norm L_1 of solution $\mathbf{u}$ is volume-weighted,

$$L_1 = \frac{\int_{\Omega_1} |\mathbf{u}_1 - \mathbf{u}_1^*| dV + \int_{\Omega_2} |\mathbf{u}_2 - \mathbf{u}_2^*| dV}{\int_{\Omega_1} dV + \int_{\Omega_2} dV}, \qquad (30)$$

with $\mathbf{u}^*$ being the final converged result. This error is plotted against the total number of multigrid V-cycles, which is a measure of total work. For CCG, (10), there are two V-cycles per iteration; one for each phase. Both methods use the same multigrid solver and the same boundary condition stencils. The only difference is the structure of the iteration: (10) vs. (12). For this problem, with the grid resolutions shown, NNIS converges faster than CCG by a factor exceeding 4×10^3. CCG shows a clear dependence on grid size: halving h doubles the number of iterations necessary for convergence. In contrast, NNIS shows a systematic but weak grid size dependence.

A grid refinement study, analyzed by Richardson extrapolation, is presented in Table 1 (with $\rho_1 = \rho_2 = 1$, and $\sigma = 0.2$). Second-order accuracy is observed in L_1, consistent with embedded boundary discretization of the Poisson operator [14]. The

Table 1
Error estimates for solution of (29) using NNIS. In this calculation $\sigma = 0.2,\ \rho_1 = 1,\ \rho_2 = 1$, and weights are $w_1 = 1/4,\ w_2 = 1/4$. Richardson extrapolation is used, comparing solutions with cell length h with solutions with length $2h$.

$(5h)^{-1}$	Iterations	L_1	Rate	L_∞	Rate
64	7				
128	6	5.018×10^{-5}		2.573×10^{-2}	
256	8	1.731×10^{-5}	1.54	1.407×10^{-2}	0.87
512	12	3.852×10^{-6}	2.17	7.005×10^{-3}	1.01
1024	16	8.790×10^{-7}	2.13	3.448×10^{-3}	1.02

Table 2
Error estimates for solution of (29) using NNIS. In this calculation $\sigma = 0.2,\ \rho_1 = 1000,\ \rho_2 = 1$, and weights are $w_1 = 0,\ w_2 = 1/2$. Richardson extrapolation is used, comparing solutions with cell length h with solutions with length $2h$.

$(5h)^{-1}$	Iterations	L_1	Rate	L_∞	Rate
64	4				
128	5	5.137×10^{-5}		5.753×10^{-2}	
256	7	1.259×10^{-5}	2.03	3.136×10^{-2}	0.88
512	11	2.649×10^{-6}	2.25	1.506×10^{-2}	1.06
1024	15	6.668×10^{-7}	1.99	7.237×10^{-3}	1.06

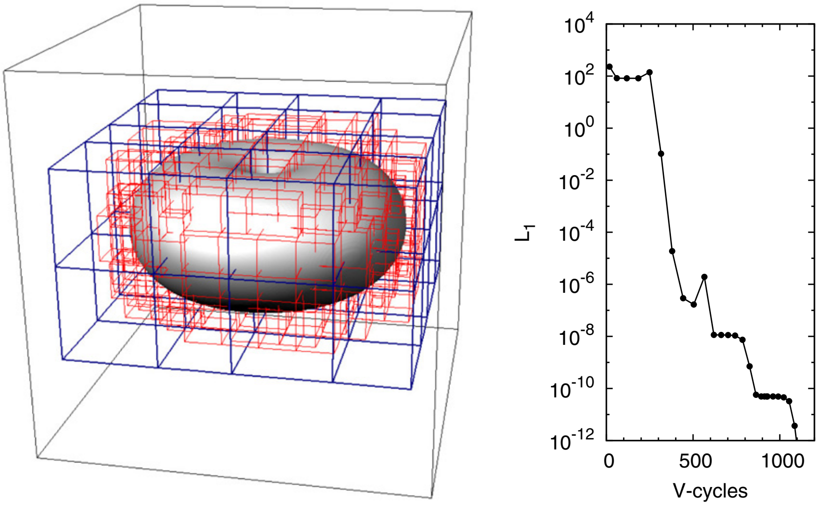

Fig. 4. Problem (29) with a 3D geometry derived from the 2D case by rotation about the y axis, with adaptive grids. The base grid is 32^3 cells with $h = 1/640$. There are two levels of refinement: patches in which $h = 1/1280$ are shown in blue, and patches in which $h = 1/2560$ are shown in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

L_∞ (max) error converges at first order, also as expected. In Fig. 3 the solution error, for given resolution, is measured against the converged result using the iterative substructuring approach. Therefore, with convergence established by Table 1, Fig. 3 shows that iterative substructuring converges to the same solution as simple iteration. The effect of density contrast is shown in Table 2 where $\rho_1 = 1000$ and $\rho_2 = 1$.

3.2. Three space dimensions

The NNIS method also works well in three dimensions. In three dimensions Ω is $[-1/10, 1/10]^3$ and Ω_1 is a volume of revolution about the y axis of the two-dimensional domain Ω_1 as shown in Fig. 4 together with the locally refined grid structure.

4. Scaling arguments

4.1. Scaling of NNIS

For the conjugate gradient method, the norm of the solution error **e** is bounded by a function of the condition number,

$$\|\mathbf{e}^{(k)}\|_{\mathbf{KC}} \leqslant 2\left(\frac{\sqrt{\operatorname{cond}\mathbf{KC}}-1}{\sqrt{\operatorname{cond}\mathbf{KC}}+1}\right)^k \|\mathbf{e}^{(0)}\|_{\mathbf{KC}}, \tag{31}$$

with $\|\mathbf{e}\|_{\mathbf{KC}} = \sqrt{\mathbf{e}^T\mathbf{KCe}}$ [21,15]. Some numerical experiments with Neumann–Neumann preconditioned conjugate gradients show that the number of iterations to achieve convergence is essentially independent of grid size [26,16]. However, Le Tallec et al. [17] derive

$$\operatorname{cond}(\mathbf{KC}) \propto \left(1+\ln^2\frac{H}{h}\right), \tag{32}$$

to express the grid size dependence of the condition number. Here, H is a characteristic domain size. The combination of (31) with (32) suggests that the number of Neumann–Neumann preconditioned conjugate gradient iterations scale like $\mathcal{O}(\ln h^{-1})$ in the limit $h \to 0$.

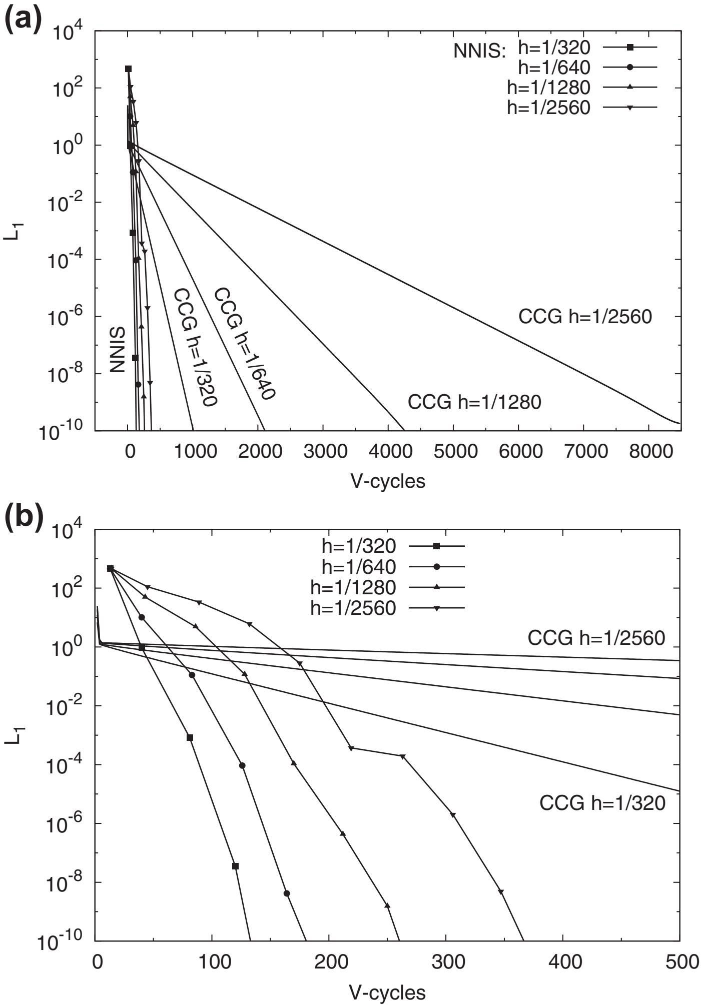

Fig. 5. Convergence of the problem complementary to the one in (29); namely, the problem in which the parameters correspond to a water drop in air: $\rho_1 = 1,\ \rho_2 = 1.2 \times 10^{-3},\ \sigma = 72.8$ with $w_1 = 0,\ w_2 = 1/2$. The NNIS algorithm shows systematic but weak h dependence, whereas the CCG iterative procedure shows that the number of V-cycles scales with $1/h$. Note the rate of convergence rate for the CCG iterative procedure has improved in a manner consistent with the predictions of the theoretical model in (41).

A bound for the norm of the Bi-CGSTAB solution error can be written

$$\|\mathbf{e}^{(k)}\|_{\mathbf{M}} \leqslant \Gamma \inf_{\Phi_k} \|\Phi_k(\mathbf{KC})\|_2 \left(1 - \frac{\lambda_{\min}(\mathbf{S})^2}{\sigma_{\max}(\tilde{\mathbf{A}})^2}\right)^{k/2} \|\mathbf{e}^{(0)}\|_{\mathbf{M}}, \tag{33a}$$

with

$$\tilde{\mathbf{A}} = \mathbf{M}^{1/2}\mathbf{KCM}^{1/2}, \tag{33b}$$

$$\mathbf{S} = \frac{1}{2}\left(\tilde{\mathbf{A}} + \tilde{\mathbf{A}}^T\right), \tag{33c}$$

where Φ_k is a polynomial of degree less than or equal to k such that $\Phi_k(0) = 1$, and Γ is a constant. $\mathbf{M}$ is a positive definite matrix, and a particular choice based on the Lanczos tridiagonalization of $\mathbf{KC}$ gives $\Gamma = 1$. Eigenvalues are denoted λ and singular values are σ. The $\|\Phi_k\|_2$ term on the right-hand side is equivalent to (31) when $\mathbf{KC}$ is positive definite, and is otherwise not easily bound. The following term on the right-hand side of (33a) results from the biconjugate construction [7,3]. The complexity of result (33a) makes a simple scaling argument difficult to derive.

We hypothesize that Neumann–Neumann preconditioned Bi-CGSTAB converges in a manner similar to Neumann–Neumann preconditioned conjugate gradients. If the number of V-cycles per Bi-CGSTAB iteration is constant, as supported by Figs. 3 and 5, and the number of operations per V-cycle is $\mathcal{O}(h^{-D})$, then our hypothesis suggests the NNIS algorithm scales as $\mathcal{O}(h^{-D} \ln h^{-1})$.

4.2. A theoretical model for the performance of the simple iterative procedure

4.2.1. The Poisson problem

In order to understand the performance and convergence properties of the simple iterative procedure we introduce a model problem that is amenable to analytic treatment. (See also [22].) Let Ω be a sphere of radius A, and let Ω_1 be a concentric sphere of radius $a < A$. The domain Ω_2 is then the spherical annulus $a < r < A$. Let the boundary condition at A be homogeneous Dirichlet, and let the source terms $\mathbf{f}_\alpha$ be zero. Then, if the boundary values at a are b for domain Ω_1, and B for domain Ω_2, the solutions to $\Delta\psi = 0$ are

$$\psi_1 = b_\ell^m \left(\frac{r}{a}\right)^\ell Y_\ell^m, \tag{34}$$

$$\psi_2 = B_\ell^m \left[\frac{\left(\frac{A}{a}\right)^\ell \left(\frac{a}{r}\right)^{\ell+1} - \left(\frac{a}{A}\right)^{\ell+1} \left(\frac{r}{a}\right)^\ell}{\left(\frac{A}{a}\right)^\ell - \left(\frac{a}{A}\right)^{\ell+1}}\right] Y_\ell^m, \tag{35}$$

where Y_ℓ^m is the spherical harmonic function of degree ℓ and order m, b_ℓ^m and B_ℓ^m are the spherical harmonic expansions of the boundary values, and the sums $\sum_{\ell=0}^\infty \sum_{m=-\ell}^\ell$ are implicit. If the flux operator is computed by differentiating a quadratic interpolation polynomial, sampling the functions ψ at $a,\ a \pm h$, and $a \pm 2h$, then

$$G_{11}\psi_1 = -\frac{\mathcal{D}_1}{2h} b_\ell^m Y_\ell^m \left[\left(\frac{a-2h}{a}\right)^\ell - 4\left(\frac{a-h}{a}\right)^\ell\right], \tag{36a}$$

$$G_{1b}\psi_1 = -\frac{3\mathcal{D}_1}{2h} b_\ell^m Y_\ell^m, \tag{36b}$$

$$G_{22}\phi_2 = -\frac{\mathcal{D}_2 B_\ell^m Y_\ell^m}{2h\left[\left(\frac{A}{a}\right)^\ell - \left(\frac{a}{A}\right)^{\ell+1}\right]} \left[\left(\frac{A}{a}\right)^\ell \left[\left(\frac{a}{a+2h}\right)^{\ell+1} - 4\left(\frac{a}{a+h}\right)^{\ell+1}\right] - \left(\frac{a}{A}\right)^{\ell+1} \left[\left(\frac{a+2h}{a}\right)^\ell - 4\left(\frac{a+h}{a}\right)^\ell\right]\right], \tag{36c}$$

$$G_{2b}\phi_2 = -\frac{3\mathcal{D}_2 B_\ell^m Y_\ell^m}{2h} \tag{36d}$$

and CCG reduces to

$$\begin{aligned}(b_\ell^m)^{(k+1)} = {} & \frac{2h}{3(\mathcal{D}_1+\mathcal{D}_2)}(j_N)_\ell^m - \frac{\mathcal{D}_2}{\mathcal{D}_1+\mathcal{D}_2}(j_D)_\ell^m - \frac{\mathcal{D}_1}{3(\mathcal{D}_1+\mathcal{D}_2)}\left[\left(\frac{a-2h}{a}\right)^\ell - 4\left(\frac{a-h}{a}\right)^\ell\right](b_\ell^m)^{(k)} \\ & - \frac{\mathcal{D}_2\left[\left(\frac{A}{a}\right)^\ell\left[\left(\frac{a}{a+2h}\right)^{\ell+1} - 4\left(\frac{a}{a+h}\right)^{\ell+1}\right] - \left(\frac{a}{A}\right)^{\ell+1}\left[\left(\frac{a+2h}{a}\right)^\ell - 4\left(\frac{a+h}{a}\right)^\ell\right]\right]}{3(\mathcal{D}_1+\mathcal{D}_2)\left[\left(\frac{A}{a}\right)^\ell - \left(\frac{a}{A}\right)^{\ell+1}\right]}(B_\ell^m)^{(k)},\end{aligned} \tag{37a}$$

$$(B_\ell^m)^{(k+1)} = \frac{2h}{3(\mathcal{D}_1+\mathcal{D}_2)}(j_N)_\ell^m + \frac{\mathcal{D}_2}{\mathcal{D}_1+\mathcal{D}_2}(j_D)_\ell^m - \frac{\mathcal{D}_1}{3(\mathcal{D}_1+\mathcal{D}_2)}\left[\left(\frac{a-2h}{a}\right)^\ell - 4\left(\frac{a-h}{a}\right)^\ell\right](b_\ell^m)^{(k)}$$
$$-\frac{\mathcal{D}_2\left[\left(\frac{A}{a}\right)^\ell\left[\left(\frac{a}{a+2h}\right)^{\ell+1} - 4\left(\frac{a}{a+h}\right)^{\ell+1}\right] - \left(\frac{a}{A}\right)^{\ell+1}\left[\left(\frac{a+2h}{a}\right)^\ell - 4\left(\frac{a+h}{a}\right)^\ell\right]\right]}{3(\mathcal{D}_1+\mathcal{D}_2)\left[\left(\frac{A}{a}\right)^\ell - \left(\frac{a}{A}\right)^{\ell+1}\right]}(B_\ell^m)^{(k)}, \tag{37b}$$

with spherical harmonic decomposition of the jumps j_N and j_D written $(j_N)_\ell^m$ and $(j_D)_\ell^m$, respectively. In addition to geometric simplification in 3 dimensions, we have assumed that step (10a) is solved completely. Abbreviating these equations as

$$\mathbf{b}^{(k)} = \mathbf{j} + \mathbf{A}\mathbf{b}^{(k-1)}, \tag{38}$$

where we have placed the indices denoting the iteration in parenthesis in order to distinguish these superscripts from exponents, one has the general solution

$$\mathbf{b}^{(k)} = (\mathbf{I}-\mathbf{A}^k)(\mathbf{I}-\mathbf{A})^{-1}\mathbf{j} + \mathbf{A}^k\mathbf{b}^{(0)}. \tag{39}$$

Convergence requires that the spectral radius of $\mathbf{A}$, $\rho(\mathbf{A})$, be less than unity, and

$$\|\mathbf{b}^{(k)} - \mathbf{b}^*\| \leqslant C\rho(\mathbf{A})\|\mathbf{b}^{(k-1)} - \mathbf{b}^*\|, \tag{40}$$

models the solution error when $\mathbf{b}^*$ is the converged solution, with C a constant depending on the choice of vector norm. For this particular matrix $\mathbf{A}$ the spectral radius is equal to the trace. In the limit $h \ll a$, this is given by

$$\rho(\mathbf{A}) \approx 1 + \frac{2h\left[-\left(\frac{a}{A}\right)^\ell(\mathcal{D}_1-\mathcal{D}_2)\ell + \left(\frac{A}{a}\right)^{\ell+1}(\mathcal{D}_2(\ell+1)+\mathcal{D}_1\ell)\right]}{3a\left[\left(\frac{a}{A}\right)^\ell - \left(\frac{A}{a}\right)^{\ell+1}\right](\mathcal{D}_1+\mathcal{D}_2)} + \mathcal{O}(h^2). \tag{41}$$

Convergence will cease in the limit $\rho \to 1$, which occurs as $h \to 0$. For given h, convergence slows as $\mathcal{D}_2/(\mathcal{D}_1+\mathcal{D}_2)$ increases. This suggests that the complementary problem – parameters corresponding to a water droplet in air – will converge far more rapidly, but will have the same scaling properties. This trend is observed (Fig. 5). The idealized theory predicts acceleration by a factor of 833 for $\ell = 0$, and a broadly consistent speedup of ≈ 740 is observed.

To assess the scaling, consider the attenuation γ of the initial error norm. Per (40), $\gamma \sim \rho^n$ in n iterations of the method. Writing (41) as $\rho \approx 1 - \epsilon h$, for small h one has $\gamma \approx 1 - nh\epsilon$. Thus, for fixed attenuation γ, n scales like $1/h$. This is exactly the behavior observed in Figs. 3 and 5 for the simple iteration procedure. With $\mathcal{O}(h^{-D})$ operations per V-cycle and $\mathcal{O}(h^{-1})$ V-cycles needed to achieve convergence, CCG scales as $\mathcal{O}(h^{-(D+1)})$. Relative to NNIS the cost of CCG scales as $\mathcal{O}(-h\ln h)^{-1}$, which is unbounded as $h \to 0$.

In fact, this idealized analytical model yields quantitative results that are broadly consistent with Fig. 3. For the $\ell = 0$ mode,

$$\ln\gamma \approx -\frac{2nh\mathcal{D}_2}{3a(\mathcal{D}_1+\mathcal{D}_2)}. \tag{42}$$

With model parameters $\mathcal{D}_1 = 1/\rho_1 \approx 833$, $\mathcal{D}_2 = 1$, and $a/h \approx 12$, which roughly corresponds to the 64 × 64 computation, the iterative procedure requires approximately 3.8×10^5 iterations, or 7.6×10^5 V-cycles, to reduce the residual by a factor of $\gamma = 10^{-11}$.

4.2.2. The heat equation

For the heat equation the same $n \propto 1/h$ behavior holds, but convergence is faster than for the Poisson problem. This can be illustrated with a model problem similar to the one for the Poisson equation. We discretize the heat equation in time with homogeneous IC as $(\mathbf{I} - k^{-2}\Delta)\psi = 0$ (backward Euler, with $k = 1/\sqrt{\Delta t\mathcal{D}}$), and the corresponding solution is

$$\psi_1 = b_\ell^m\sqrt{\frac{a}{r}}\frac{I_{\ell+\frac{1}{2}}(k_1 r)}{I_{\ell+\frac{1}{2}}(k_1 a)}Y_\ell^m, \tag{43a}$$

$$\psi_2 = B_\ell^m\sqrt{\frac{a}{r}}\left[\frac{I_{\ell+\frac{1}{2}}(k_2A)K_{\ell+\frac{1}{2}}(k_2r) - K_{\ell+\frac{1}{2}}(k_2A)I_{\ell+\frac{1}{2}}(k_2r)}{I_{\ell+\frac{1}{2}}(k_2A)K_{\ell+\frac{1}{2}}(k_2a) - K_{\ell+\frac{1}{2}}(k_2A)I_{\ell+\frac{1}{2}}(k_2a)}\right]Y_\ell^m, \tag{43b}$$

in terms of modified Bessel functions I and K. Constructing the fluxes as before, the spectral radius of the resulting matrix $\mathbf{A}$ is

$$
\begin{aligned}
\rho(\mathbf{A}) = &-\frac{\mathcal{D}_1 a^{\frac{1}{2}}}{3(\mathcal{D}_1+\mathcal{D}_2)}\left[\frac{I_{\ell+\frac{1}{2}}(k_1(a-2h))}{(a-2h)^{\frac{1}{2}} I_{\ell+\frac{1}{2}}(k_1 a)} - 4\frac{I_{\ell+\frac{1}{2}}(k_1(a-h))}{(a-h)^{\frac{1}{2}} I_{\ell+\frac{1}{2}}(k_1 a)}\right] \\
&-\frac{\mathcal{D}_2 a^{\frac{1}{2}}}{3(\mathcal{D}_1+\mathcal{D}_2)}\left[\frac{I_{\ell+\frac{1}{2}}(k_2 A)K_{\ell+\frac{1}{2}}(k_2(a+2h)) - K_{\ell+\frac{1}{2}}(k_2 A)I_{\ell+\frac{1}{2}}(k_2(a+2h))}{(a+2h)^{\frac{1}{2}}(I_{\ell+\frac{1}{2}}(k_2 A)K_{\ell+\frac{1}{2}}(k_2 a) - K_{\ell+\frac{1}{2}}(k_2 A)I_{\ell+\frac{1}{2}}(k_2 a))}\right. \\
&\left.-4\frac{I_{\ell+\frac{1}{2}}(k_2 A)K_{\ell+\frac{1}{2}}(k_2(a+h)) - K_{\ell+\frac{1}{2}}(k_2 A)I_{\ell+\frac{1}{2}}(k_2(a+h))}{(a+h)^{\frac{1}{2}}(I_{\ell+\frac{1}{2}}(k_2 A)K_{\ell+\frac{1}{2}}(k_2 a) - K_{\ell+\frac{1}{2}}(k_2 A)I_{\ell+\frac{1}{2}}(k_2 a))}\right] \qquad (44)
\end{aligned}
$$

and in the limit $h \to 0$ it becomes

$$
\rho(\mathbf{A}) \approx 1 + \frac{2h}{3(\mathcal{D}_1+\mathcal{D}_2)}\left[\frac{(\mathcal{D}_2-\mathcal{D}_1)\ell}{a} - \frac{\mathcal{D}_1 k_1 I_{\ell+\frac{3}{2}}(ak_1)}{I_{\ell+\frac{1}{2}}(ak_1)} - \frac{\mathcal{D}_2 k_2 K_{\ell+\frac{3}{2}}(ak_2)}{K_{\ell+\frac{1}{2}}(ak_2)}\right] + \mathcal{O}(h^2). \qquad (45)
$$

If we take also the limit $\Delta t \to \infty$ we approach Poisson behavior:

$$
\rho(\mathbf{A}) \approx 1 - \frac{2h}{3a(\mathcal{D}_1+\mathcal{D}_2)}[\mathcal{D}_1\ell + \mathcal{D}_2(\ell+1)] - \frac{2ah}{3(\mathcal{D}_1+\mathcal{D}_2)\Delta t(3+2\ell)} + \mathcal{O}(h^2) + \mathcal{O}(h/\Delta t^2). \qquad (46)
$$

When $\Delta t \to \infty$, the attenuation of the $\ell = 0$ mode implied by (46) is equal to the attenuation of that mode implied by (41). However, the attenuation of the $\ell = 0$ mode in (46) increases as Δt decreases, indicating that the heat problem converges

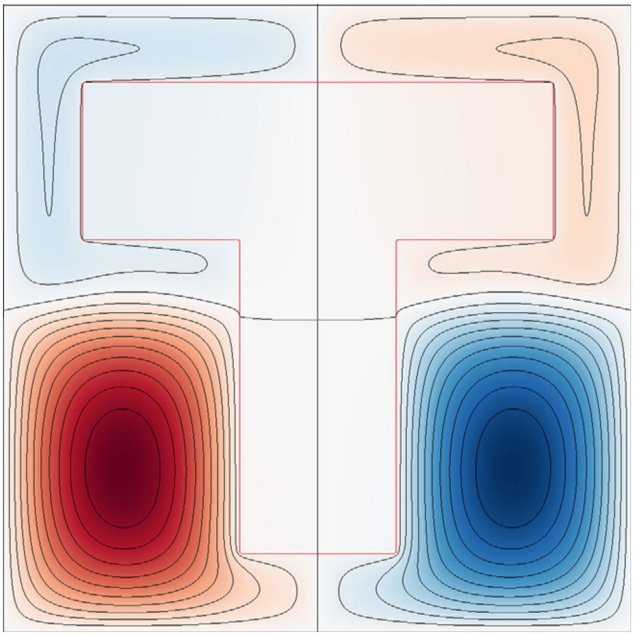

Fig. 6. Test Problem 2 from Tatebe [32] solved as a two-domain Poisson problem. The potential ϕ ranges from -1.104 (blue) to $+1.104$ (red); the colormap is blue for all negative values and red for all positive values. The contours are drawn at 21 equally spaced levels from $\phi = -1$ to $\phi = 1$ inclusive. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
The L_1 and L_∞ norm of the solution error of (47) estimated by Richardson extrapolation as a function of the grid size h together with the rate at which this difference goes to zero.

$1/h$	Iterations	L_1	Rate	L_∞	Rate
64	14				
128	15	2.485×10^{-3}		4.551×10^{-2}	
256	20	7.303×10^{-4}	1.77	2.596×10^{-2}	0.81
512	16	2.334×10^{-4}	1.65	1.551×10^{-2}	0.74
1024	22	8.001×10^{-5}	1.54	9.484×10^{-3}	0.71

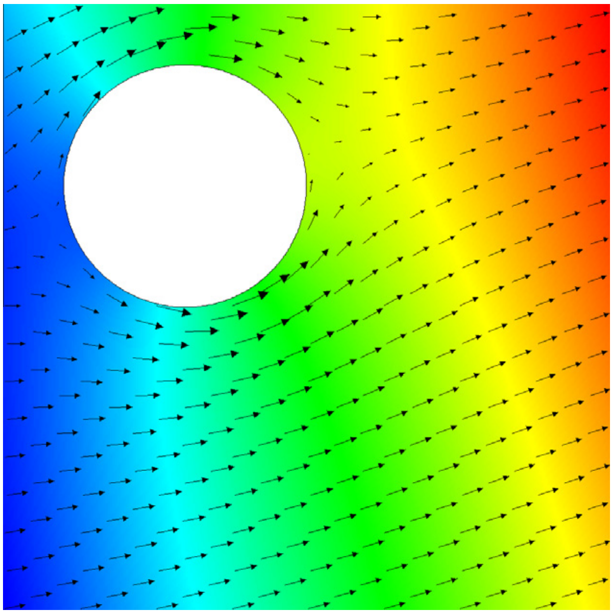

Fig. 7. Potential flow Problem 1: center at (0.30,0.70) relative to lower left corner. The potential ϕ ranges from -0.5011 (blue) to 0.8036 (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

more rapidly than the Poisson problem for comparable parameters. As with the Poisson problem, when $\mathcal{D}_2 > \mathcal{D}_1$ convergence will be faster. However, for any $\mathcal{D}$ and any Δt the simple iterative method scales like $h^{-(D+1)}$.

5. The Performance of NNIS on two standard Poisson problems

5.1. The solution of a discontinuous coefficient Poisson problem

The two-phase Poisson problem can be used to resolve variable coefficient Poisson problems with discontinuous coefficients where the jump in the coefficients occurs on the boundary $\partial\Omega_{12}$ between two domains Ω_1 and Ω_2. For example, here we compute the solution to Problem 2 in Tatebe [32]:

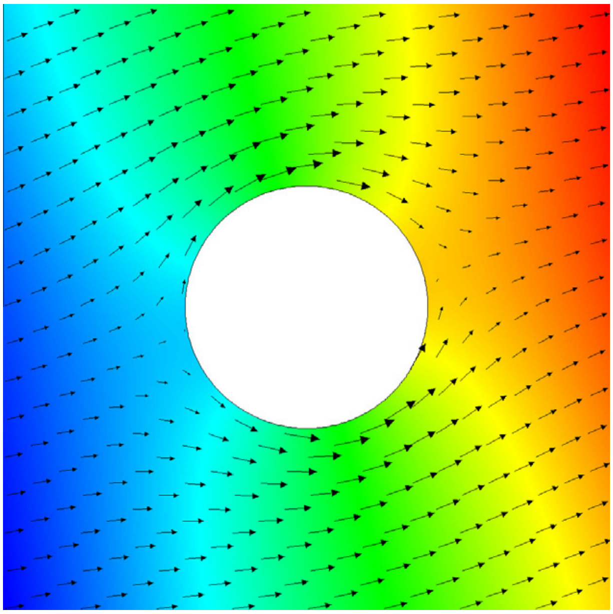

Fig. 8. Potential flow Problem 2: center at (0.50,0.50) relative to lower left corner. ϕ ranges from -0.6592 (blue) to 0.6592 (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

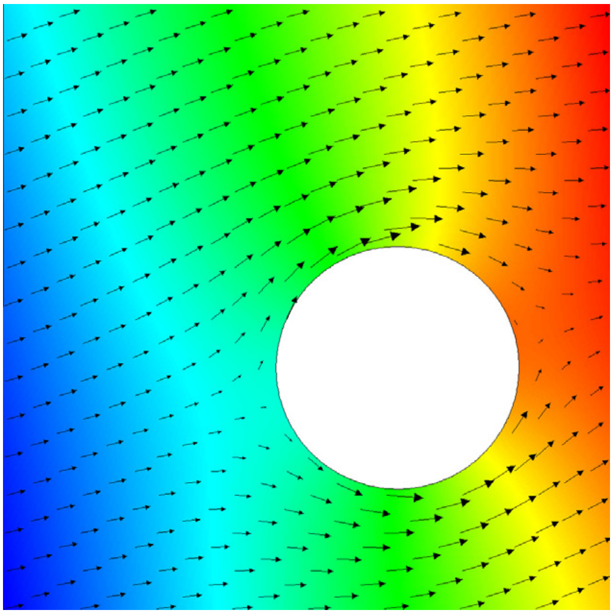

Fig. 9. Potential flow Problem 3: center at (0.65,0.40) relative to lower left corner. ϕ ranges from -0.7794 (blue) to 0.5320 (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
The L_1 and L_∞ norms of the difference between the true and computed values of ϕ as a function of the grid size h for potential flow problem (51)–(53), together with the rate at which this difference goes to zero.

$1/h$	Iterations	L_1	Rate	L_∞	Rate
Centering 1					
32	3	9.541×10^{-5}		5.967×10^{-4}	
64	4	2.381×10^{-5}	2.00	1.127×10^{-4}	2.40
128	5	5.990×10^{-6}	1.99	2.243×10^{-5}	2.32
256	6	1.509×10^{-6}	1.99	5.490×10^{-6}	2.03
Centering 2					
32	2	6.809×10^{-5}		7.394×10^{-4}	
64	3	1.699×10^{-5}	2.00	9.687×10^{-5}	2.93
128	3	4.299×10^{-6}	1.98	1.668×10^{-5}	2.54
256	4	1.082×10^{-6}	1.99	4.164×10^{-6}	2.00
Centering 3					
32	3	6.248×10^{-5}		6.516×10^{-4}	
64	4	1.582×10^{-5}	1.98	8.223×10^{-5}	2.99
128	5	3.996×10^{-6}	1.99	1.754×10^{-5}	2.23
256	8	1.007×10^{-6}	1.99	4.454×10^{-6}	1.98

$$\begin{aligned} &\nabla\cdot(k\nabla\phi)=f \quad \text{on} \quad \Omega=[0,1]\times[0,1],\\ &\phi=0 \quad \text{on} \quad \partial\Omega, \end{aligned} \tag{47}$$

where the functions k and f are given by

$$k=\begin{cases} 100 & \Omega'=\left(\left[\frac{3}{8},\frac{5}{8}\right]\times\left[\frac{1}{8},\frac{7}{8}\right]\right)\cup\left(\left[\frac{1}{8},\frac{7}{8}\right]\times\left[\frac{5}{8},\frac{7}{8}\right]\right)\\ 1 & \Omega\setminus\Omega', \end{cases} \tag{48}$$

$$f=\begin{cases} 80 & \Omega''=\left(\left[0,\frac{1}{2}\right]\times\left[0,\frac{1}{2}\right]\right)\cup\left(\left[\frac{1}{2},1\right]\times\left[\frac{1}{2},1\right]\right)\\ -80 & \Omega\setminus\Omega''. \end{cases} \tag{49}$$

We partition the domain so that the coefficient k is constant on each region $\Omega_1,\ \Omega_2$ where Ω_1 and Ω_2 are defined by

$$\begin{aligned}
&\Omega_1 = \Omega', \\
&\Omega_2 = \Omega \setminus \Omega_1, \\
&\Delta\phi = f/100 \quad \text{on} \quad \Omega_1, \\
&\Delta\phi = f \quad \text{on} \quad \Omega_2, \\
&\phi = 0 \quad \text{on} \quad \partial\Omega, \\
&[\![\phi]\!] = 0 \quad \text{on} \quad \partial\Omega_{12}, \\
&\left[\!\!\left[\frac{1}{\rho}(\mathbf{n}_1 \cdot \nabla)\phi\right]\!\!\right] = 0 \quad \text{on} \quad \partial\Omega_{12}.
\end{aligned} \tag{50}$$

(The region Ω_1 is the 'T' shaped region displayed in Fig. 6, while Ω_2 is the complement of this region.) We set $\rho_1 = 1/100$ and $\rho_2 = 1$ and solve the resulting system using NNIS with weights $w_1 = w_2 = 0.25$.

As shown in Table 3 the computed solution converges to the true solution as a function of grid size h at a superlinear, but less than second-order, rate in the L_1 norm. This is to be expected, since there is singular behavior at the corners of the 'T' shaped region on which $k = 100$ and we are resolving this region exactly (e.g., see p. 278 of [29]). Although the number of iterations of preconditioned Bi-CGSTAB is not constant, they increase by less than a factor of 2 as the problem size is increased by a factor of 2^4. The number of iterations of our method is larger than the number of iterations reported in [32] by roughly a factor of two and each iteration of our method required on the order of 25 multigrid V-cycles, which is more work than the computation in [32]. However we resolve the jump in k exactly along the boundary of the 'T' shaped domain Ω_1 as shown in Fig. 6.

5.2. The computation of potential flow around a circle

In our final test problem, we compute the potential flow $\mathbf{v} = (v_x, v_y) = \nabla\phi$ around a circle of radius a centered at the point $\mathbf{x}_0$ with boundary conditions at infinity given by

$$\mathbf{v} = U\begin{pmatrix}\cos\theta \\ \sin\theta\end{pmatrix} \quad \text{as} \quad (x, y) \to (\pm\infty, \pm\infty) \tag{51}$$

for some constant U and orientation θ. The flow is divergence free

$$\nabla \cdot \mathbf{v} = 0 \tag{52}$$

and satisfies the 'no-flow' boundary condition on the circle,

$$\mathbf{v} \cdot \mathbf{n} = 0 \quad \text{on} \quad \|\mathbf{x} - \mathbf{x}_0\|_2 = a. \tag{53}$$

Table 5
The L_1 and L_∞ norms of the difference between the true and computed values of the x component of the velocity $v_x = \partial\phi/\partial x$ as a function of the grid size h for the potential flow problem (51)–(53), together with the rate at which this difference goes to zero.

$1/h$	Iterations	L_1	Rate	L_∞	Rate
Centering 1					
32	3	1.390×10^{-3}		1.604×10^{-2}	
64	4	3.740×10^{-4}	1.89	4.990×10^{-3}	1.68
128	5	9.851×10^{-5}	1.92	1.338×10^{-3}	1.90
256	6	2.540×10^{-5}	1.96	3.932×10^{-4}	1.77
Centering 2					
32	2	1.524×10^{-3}		1.214×10^{-2}	
64	3	4.158×10^{-4}	1.87	4.489×10^{-3}	1.44
128	3	1.073×10^{-4}	1.95	1.291×10^{-3}	1.80
256	4	2.742×10^{-5}	1.97	3.418×10^{-4}	1.92
Centering 3					
32	3	1.441×10^{-3}		1.223×10^{-2}	
64	4	3.868×10^{-4}	1.87	3.901×10^{-3}	1.65
128	5	1.028×10^{-4}	1.91	1.305×10^{-3}	1.58
256	8	2.639×10^{-5}	1.96	3.628×10^{-4}	1.85

Table 6
The L_1 and L_∞ norms of the difference between the true and computed values of the y component of the velocity $v_y = \partial\phi/\partial x$ as a function of the grid size h for the potential flow problem (51)–(53), together with the rate at which this difference goes to zero.

$1/h$	Iterations	L_1	Rate	L_∞	Rate
Centering 1					
32	3	1.379×10^{-3}		1.332×10^{-2}	
64	4	3.684×10^{-4}	1.90	4.595×10^{-3}	1.54
128	5	9.783×10^{-5}	1.91	1.337×10^{-3}	1.78
256	6	2.512×10^{-5}	1.96	3.751×10^{-4}	1.83
Centering 2					
32	2	1.406×10^{-3}		1.152×10^{-2}	
64	3	3.957×10^{-4}	1.83	4.184×10^{-3}	1.43
128	3	1.027×10^{-4}	1.95	1.264×10^{-3}	1.73
256	4	2.637×10^{-5}	1.96	3.440×10^{-4}	1.88
Centering 3					
32	3	1.438×10^{-3}		1.440×10^{-2}	
64	4	3.936×10^{-4}	1.87	4.581×10^{-3}	1.65
128	5	1.035×10^{-4}	1.93	1.328×10^{-3}	1.79
256	8	2.658×10^{-5}	1.96	3.559×10^{-4}	1.90

In our implementation of this problem we write it as a two-phase potential problem

$$\begin{aligned} &\Delta\phi = 0 && \text{on} \quad \Omega_1 \text{ and } \Omega_2, \\ &[\![\phi]\!] = 0 && \text{on} \quad \partial\Omega_{12}, \\ &\left[\!\!\left[\frac{1}{\rho}(\mathbf{n}_1\cdot\nabla)\phi\right]\!\!\right] = 0 && \text{on} \quad \partial\Omega_{12}, \\ &\mathbf{v} = \nabla\phi && \text{on} \quad \Omega \equiv \Omega_1 \cup \Omega_2, \end{aligned} \tag{54}$$

with boundary conditions on the exterior boundary $\partial\Omega$ given by the true solution (e.g., see [5]). The problem in (54) is equivalent to (51)–(53) in the limit $\rho_1/\rho_2 \to \infty$, where Ω_1 is the interior of the circle of radius a centered at $\mathbf{x}_0$. We compute approximate solutions of (54) on the unit square with $U = 1$ and $\theta = 15^\circ$. We approximate the density contrast by letting $\rho_2 = 1$ and $\rho_1 = 10^9$ and use weights $w_1 = 0$ and $w_2 = 1/2$.

In order to demonstrate the performance of our algorithm as a function of the location of the position of the circle in the computational domain, we perform three separate computations of this problem with the center of the circle at different positions in the computational domain. The computed results are displayed in Figs. 7–9. The black arrows represent the velocity, $\mathbf{v} = \nabla\phi$, which we compute using centered differences inside the domain and with third-order one-sided differences at domain boundaries.

For this problem we measure error relative to the true solution, which is known (e.g., see [5]). The difference between the true and computed values of ϕ goes to zero at a second-order rate in both the L^1 and L_∞ norms as shown in Table 4. As shown in Tables 5 and 6, the difference between the true and computed values of the velocity also goes to zero which tends toward second-order the in L^1 norm as the grid is resolved. Similarly the rate at which this difference goes to zero in the L_∞ norm tends toward second-order as the grid is resolved. All points in the domain are used to calculate the error norms, but in Figs. 7–9 only ≈400 representative velocity arrows are shown for clarity.

6. Conclusions

We have presented a new method for computing Poisson problems with jumps at internal boundaries, such as occur when computing the pressure in multifluid problems with surface tension. Our approach is based on applying existing second-order accurate, finite volume embedded boundary solvers for single domain Poisson problems with irregular geometries to multiple domains using a domain decomposition approach; namely, Neumann–Neumann preconditioned iterative substructuring (NNIS). This method resolves the interface as a sharp boundary and resolves jumps in the pressure and its gradient with no smoothing or regularization near the boundary. We validate the method by solving several well-known variable coefficient Poisson problems, including potential flow around a circle for which an analytic solution is available. In addition, this problem demonstrates that the method works well for very large density ratios (e.g., 10^9).

We compare the performance of our method with the approach taken by Crockett et al. (CCG) in [6], in which the authors use a simple iterative procedure for solving the discretized system of equations. While both methods converge to the same solution, NNIS has superior performance and superior scaling. In particular the iterative procedure in CCG is very sensitive to model parameters and grid resolution.

We have developed an analytical model to theoretically explain these sensitivities. Both theoretically and heuristically we show the iterative procedure scales like $\mathcal{O}(h^{-(D+1)})$, where h is the cell size and $D = 2, 3$ is the dimensionality of the problem.

For a 2D computation of a water droplet in air with 64×64 grid cells the computational cost of the iterative procedure in CCG is about 8 times more than iterative substructuring. This performance difference increases with increasing grid resolution and is strongly dependent on the problem parameters. For example, the cost of computing a problem with the parameters corresponding to an air bubble in water rather than a water droplet in air with the simple iterative procedure increases by an additional factor of 740, and requires as many as 3×10^6 multigrid V-cycles for a problem in two space dimensions with 256×256 grid cells.

On the other hand, in the model computations with physical parameters corresponding to a water droplet in air and an air bubble in water the NNIS approach is only weakly sensitive to the material parameters, grid resolution, and dimensionality. For these problems NNIS requires fewer than 1.1×10^3 multigrid V-cycles to converge to the exact solution. If the convergence of our NNIS algorithm, which is a nonsymmetric Bi-CGSTAB system, is similar to the convergence of symmetric conjugate gradient Neumann–Neumann preconditioned systems, then our iterative substructuring approach should scale like $\mathcal{O}(h^{-D} \ln h^{-1})$.

Acknowledgments

This work was supported by the National Science Foundation under Grant No. DMS-0810939, and by the U.S. Department of Energy Office of Advanced Scientific Computing Research under Grant Nos. DE-SC0001981 and DE-FG02-03ER25579.

References

[1] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.

[2] H.S. Carslaw, J.C. Jaeger, Conduction of Heat in Solids, second ed., Oxford University Press, Oxford, 1959.

[3] T.F. Chan, T. Szeto, Composite step product methods for solving nonsymmetric linear systems, SIAM J. Sci. Comput. 17 (1996) 1491–1508.

[4] Y.C. Chang, T.Y. Hou, B. Merriman, S. Osher, A level set formulation of Eulerian interface capturing methods for incompressible fluid flows, J. Comput. Phys. 124 (1996) 449–464.

[5] A.J. Chorin, J.E. Marsden, A Mathematical Introduction to Fluid Mechanics, third ed., Springer, New York, 1993.

[6] R. Crockett, P. Colella, D. Graves, A Cartesian grid embedded boundary method for solving the Poisson and heat equations with discontinuous coefficients in three dimensions, J. Comput. Phys. 230 (2011) 2451–2469.

[7] S.C. Eisenstat, H.C. Elman, M.H. Schultz, Variational iterative methods for nonsymmetric systems of linear equations, SIAM J. Numer. Anal. 20 (1983) 345–357.

[8] M.J. Gander, Optimized Schwarz methods, SIAM J. Numer. Anal. 44 (2006) 699–731.

[9] M.J. Gander, L. Halpern, F. Magoulès, An optimized Schwarz method with two-sided Robin transmission conditions for the Helmholtz equation, Int. J. Numer. Methods Fluids 55 (2007) 163–175.

[10] L. Grasedyck, R. Kriemann, S. Le Borne, A sparse matrix arithmetic based on h-matrices. Part I: Introduction to h-matrices, Computing 62 (1999) 89–108.

[11] L. Grasedyck, R. Kriemann, S. Le Borne, H-matrices for convection–diffusion problems with constant convection, Computing 70 (2003) 261–274.

[12] L. Grasedyck, R. Kriemann, S. Le Borne, Domain decomposition based h-lu preconditioning, Numer. Math. 112 (2009) 565–600.

[13] W. Helfrich, Elastic properties of lipid bilayers: theory and possible experiments, Z. Naturforsch. 28c (1973) 693–703.

[14] H. Johansen, P. Colella, A Cartesian grid embedded boundary method for Poisson's equation in irregular domains, J. Comput. Phys. 147 (1998) 60–85.

[15] S. Kaniel, Estimates of some computational techniques in linear algebra, Math. Comput. 20 (1966) 369–378.

[16] P. Le Tallec, Domain decomposition methods in computational mechanics, Comput. Mech. Adv. 1 (1994) 121–220.

[17] P. Le Tallec, J. Mandel, M. Vidrascu, A Neumann–Neumann domain decomposition algorithm for solving plate and shell problems, SIAM J. Numer. Anal. 35 (1998) 836–867.

[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] D.F. Martin, K.L. Cartwright, Solving Poisson's equation using adaptive mesh refinement, U.C. Berkeley Electronics Research Laboratory Report No. UCB/ERL M96/66, October 19, 1996.

[20] V. Martin, An optimized Schwarz waveform relaxation method for the unsteady convection diffusion equation in two dimensions, Comput. Fluids 33 (2004) 829–837.

[21] G. Meinardus, Über eine verallgemeinerung einer ungleichung von L.V. Kantorowitsch, Numer. Math. 5 (1963) 14–23.

[22] G. Miller, An iterative boundary potential method for the infinite domain Poisson problem with interior Dirichlet boundaries, J. Comput. Phys. 227 (2008) 7917–7928.

[23] G. Miller, D. Trebotich, An embedded boundary method for the Navier–Stokes equations on a time-dependent domain, Commun. Appl. Math. Comput. Sci. 7 (2012) 1–31.

[24] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.

[25] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 1–39.

[26] F.-X. Roux, Spectral analysis of the interface operators associated with the preconditioned saddle-point principle domain decomposition method, in: D.E. Keyes, T.F. Chan, G. Meurant, J.S. Scroggs, R.G. Voigt (Eds.), Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, 1992, pp. 73–90.

[27] B. Smith, P. Bjørstad, W. Gropp, Domain Decomposition, Cambridge University Press, Cambridge, 1996.

[28] S.L. Sobolev, Partial Differential Equations of Mathematical Physics, Pergamon, Oxford, 1964.

[29] G. Strang, G.J. Fix, An Analysis of the Finite Element Method, Prentice-Hall, Englewood Cliffs, NJ, 1973.

[30] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two phase flows, J. Comput. Phys. 162 (2000) 301–337.

[31] M.M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1) (1994) 146–159.

[32] O. Tatebe, The multigrid preconditioned conjugate gradient method, in: Sixth Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO., April 4–9, 1993.

[33] A. Toselli, O. Widlund, Domain Decomposition Methods – Algorithms and Theory, Springer, New York, 2005.

[34] I.W. Turner, J.A. Belward, M.N. Oqielat, Error bounds for least squares gradient estimates, SIAM J. Sci. Comput. 32 (2010) 2146–2166.

[35] H. van der Vorst, BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (2) (1992) 631–644.