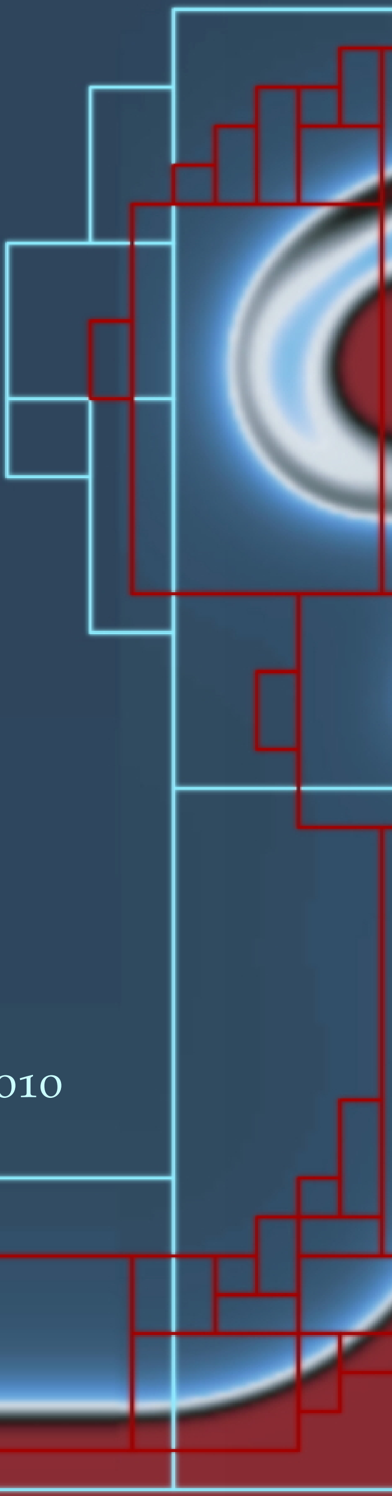
Communications in Applied Mathematics and Computational Science

vol. 5 no. 1 2010



mathematical sciences publishers

Communications in Applied Mathematics and Computational Science

pjm.math.berkeley.edu/camcos

See inside back cover or pjm.math.berkeley.edu/camcos for submission instructions.

The subscription price for 2010 is US $70/year for the electronic version, and $100/year for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscribers address should be sent to Mathematical Sciences Publishers, Department of Mathematics, University of California, Berkeley, CA 94720-3840, USA.

Communications in Applied Mathematics and Computational Science, at Mathematical Sciences Publishers, Department of Mathematics, University of California, Berkeley, CA 94720-3840 is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

CAMCoS peer-review and production is managed by EditFLOW$^{\text{TM}}$ from Mathematical Sciences Publishers.

PUBLISHED BY
mathematical sciences publishers
http://www.mathscipub.org
A NON-PROFIT CORPORATION
Typeset in LaTeX

COMM. APP. MATH. AND COMP. SCI.
Vol. 5, No. 1, 2010

FETI AND BDD PRECONDITIONERS FOR STOKES–MORTAR–DARCY SYSTEMS

JUAN GALVIS AND MARCUS SARKIS

We consider the coupling across an interface of a fluid flow and a porous media flow. The differential equations involve Stokes equations in the fluid region, Darcy equations in the porous region, plus a coupling through an interface with Beaver–Joseph–Saffman transmission conditions. The discretization consists of P2/P1 triangular Taylor–Hood finite elements in the fluid region, the lowest order triangular Raviart–Thomas finite elements in the porous region, and the mortar piecewise constant Lagrange multipliers on the interface. We allow for nonmatching meshes across the interface. Due to the small values of the permeability parameter κ of the porous medium, the resulting discrete symmetric saddle point system is very ill conditioned. We design and analyze preconditioners based on the finite element by tearing and interconnecting (FETI) and balancing domain decomposition (BDD) methods and derive a condition number estimate of order $C_1(1+(1/\kappa))$ for the preconditioned operator. In case the fluid discretization is finer than the porous side discretization, we derive a better estimate of order $C_2((\kappa+1)/(\kappa+(h^p)^2))$ for the FETI preconditioner. Here h^p is the mesh size of the porous side triangulation. The constants C_1 and C_2 are independent of the permeability κ, the fluid viscosity ν, and the mesh ratio across the interface. Numerical experiments confirm the sharpness of the theoretical estimates.

1. Introduction

We consider the coupling across an interface of a fluid flow and a porous media flow. The model consists of Stokes equations in the fluid region, Darcy equations for the filtration velocity in the porous medium, and an adequate transmission condition for coupling of these equations through an interface. Such problems appear in several applications such as well-reservoir coupling in petroleum engineering, transport of substances across groundwater and surface water, and (bio)fluid-organ interactions. There are works that address numerical analysis issues of this model. For $\inf-\sup$ conditions and approximation results associated to the continuous and

MSC2000: 35Q30, 65N22, 65N30, 65N55, 76D07.

Keywords: Stokes–Darcy coupling, mortar, balancing domain decomposition, FETI, saddle point problems, nonmatching grids, discontinuous coefficients, mortar elements.

discrete formulations for Stokes–Laplacian systems we refer [15; 12], for Stokes–Darcy systems we refer [31; 39; 2; 22], for Stokes–Mortar–Darcy systems, see [41; 26], and for DG discretizations [11; 41]. For studies on preconditioning analysis for Stokes-Laplacian systems, see [13; 14; 16; 17], and for Stokes-Darcy systems [3]. In this paper, we are interested in balancing domain decomposition (BDD) and finite element by tearing and interconnecting (FETI) preconditioned conjugate gradient methods for *Stokes–Mortar–Darcy* systems. For general references on BDD and FETI type methods, see [18; 19; 23; 24; 30; 33; 34; 35; 36; 40; 42; 43; 44].

In this paper we both extend some preliminary results contained in [25] and introduce and analyze new methods. We note that the BDD-I preconditioner introduced in [25] is not effective for small permeabilities (in real applications permeabilities are very small) while the preconditioner BDD-II in [25] requires constructing interface base functions which are orthogonal in the Stokes inner product (this construction is very expensive and impractical because it requires, as a precomputational step, solving many Stokes problems). Here in this paper we circumvent these issues by introducing a dual formulation and considering FETI-based methods. We propose and analyze FETI methods and present numerical experiments in order to verify the theory. We note that the analysis of the FETI algorithms for Stokes–Mortar–Darcy problems is very challenging due to the following issues:

(i) the mortar map from the Stokes to the Darcy side has a large kernel since the Stokes velocity space is in general richer than the Darcy velocity space on the interface;

(ii) the trace space of the Stokes velocity ($H^{1/2}$) is more regular than the trace space of the Darcy flux ($H^{-1/2}$), and due to a priori error estimates [31; 41; 26], the Stokes side must be chosen as the master side;

(iii) the energy associated to the Darcy region is much larger than the energy associated to the Stokes region due to the small value of the permeability.

Such issues imply that the master side must be chosen on the Stokes side and where the energy is smaller and velocity space is richer. The mathematical analysis under this choice is very hard to analyze even for simpler problems such as for transmission problems with discontinuous coefficients using Mortar or DG discretizations [19; 20; 21]. For problems where both the smallest coefficient and the finest mesh are placed on the master side, as far as we know, there are no optimal preconditioners developed in the literature for transmission problems, and typically there is a condition to rule out such a choice.

The rest of the paper is organized as follows: in Section 2 we present the Stokes–Darcy coupling model. In Section 3 we describe the weak formulation of this model. In Section 4 we introduce a finite element discretization. In Section 5 we study the primal and dual formulation of the discrete problem. Section 6 presents a complete analysis of the BDD-I preconditioner introduced in [25]. In Section 7, we design and analyze the FETI preconditioner; see Lemma 3 and Theorem 4. In particular we obtain the condition number estimate of order $C_1(1+(1)/(\kappa))$ for this preconditioner and also prove Theorem 7, which gives a better estimate of order $C_2((\kappa+1)/(\kappa+(h^p)^2))$ for the FETI preconditioner in case the fluid discretization is finer than the porous side discretization; the case where the Stokes mesh is not a refinement of the Darcy mesh is also discussed (see Remark 8). In Section 7 we also consider more general fluid bilinear forms by allowing the presence of a tangential interface fluid velocity energy (Remark 10), and also translate the FETI results to analyze certain BDD methods (Remark 9). In Section 8 we present the numerical results, and in Section 9 we discuss the multisubdomain case.

Here h^p is the mesh size of the porous side triangulation. The constants C_1 and C_2 are independent of the permeability κ, the fluid viscosity ν, and the mesh ratio across the interface. In Section 8 we present numerical results that confirm the theoretical estimates concerning the BDD and the FETI preconditioners.

2. Problem setting

Let $\Omega^f, \Omega^p \subset \mathbb{R}^n$ be polyhedral subdomains, define $\Omega := \operatorname{int}(\overline{\Omega}^f \cup \overline{\Omega}^p)$ and $\Gamma := \partial\Omega^f \cap \partial\Omega^p$, with outward unit normal vectors η^i on $\partial\Omega^i$, $i = f, p$. The tangent vectors on Γ are denoted by τ_1 ($n = 2$), or τ_l, $l = 1, 2$ ($n = 3$). The exterior boundaries are $\Gamma^i := \partial\Omega^i \setminus \Gamma$, $i = f, p$. Fluid velocities are denoted by $\boldsymbol{u}^i : \Omega^i \to \mathbb{R}^n$, $i = f, p$, and pressures by $p^i : \Omega^i \to \mathbb{R}$, $i = f, p$.

We consider Stokes equations in the fluid region Ω^f and Darcy equations for the filtration velocity in the porous medium Ω^p. More precisely, we have the following systems of equations in each subdomain:

$$\begin{array}{cc}
\text{Stokes equations} & \text{Darcy equations} \\
\left\{\begin{aligned} -\nabla\cdot T(\boldsymbol{u}^f, p^f) &= \boldsymbol{f}^f && \text{in } \Omega^f, \\ \nabla\cdot\boldsymbol{u}^f &= g^f && \text{in } \Omega^f, \\ \boldsymbol{u}^f &= \boldsymbol{h}^f && \text{on } \Gamma^f, \end{aligned}\right. &
\left\{\begin{aligned} \boldsymbol{u}^p &= -\tfrac{\kappa}{\nu}\nabla p^p && \text{in } \Omega^p, \\ \nabla\cdot\boldsymbol{u}^p &= g^p && \text{in } \Omega^p, \\ \boldsymbol{u}^p\cdot\eta^p &= h^p && \text{on } \Gamma^p. \end{aligned}\right.
\end{array} \tag{1}$$

Here $T(\boldsymbol{v}, p) := -pI + 2\nu\boldsymbol{D}\boldsymbol{v}$, where ν is the fluid viscosity, $\boldsymbol{D}\boldsymbol{v} := \frac{1}{2}(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T)$ is the linearized strain tensor and κ denotes the rock permeability. For simplicity on the analysis, we assume that κ is a real positive constant. We impose the following conditions:

(1) Interface matching conditions across Γ; see [15; 12; 16; 31] and references therein.
   (a) Conservation of mass across Γ: $\boldsymbol{u}^f\cdot\eta^f + \boldsymbol{u}^p\cdot\eta^p = 0$ on Γ.
   (b) Balance of normal forces across Γ: $p^f - 2\nu\eta^{fT}\boldsymbol{D}(\boldsymbol{u}^f)\eta^f = p^p$ on Γ.

(c) Beavers–Joseph–Saffman condition: this condition is a kind of empirical law that gives an expression for the component of the Cauchy stress tensor in the tangential direction of Γ; see [4] and [29]. It is expressed by

$$\boldsymbol{u}^f \cdot \tau_l = -\frac{\sqrt{\kappa}}{\alpha^f} 2\eta^{fT} \boldsymbol{D}(\boldsymbol{u}^f)\tau_l, \quad l = 1, n-1, \quad \text{on } \Gamma.$$

(2) Compatibility condition: the divergence and boundary data satisfy (see [26])

$$\langle g^f, 1\rangle_{\Omega^f} + \langle g^p, 1\rangle_{\Omega^p} - \langle \boldsymbol{h}^f \cdot \eta^f, 1\rangle_{\Gamma^f} - \langle h^p, 1\rangle_{\Gamma^p} = 0.$$

3. Weak formulation

In this section we present the weak version of the coupled system of partial differential equations introduced above. Without loss of generality, we consider $\boldsymbol{h}^f = \boldsymbol{\Gamma}$, $g^f = 0$, $h^p = 0$ and $g^p = 0$ in (1); see [26].

The problem can be formulated as: *Find* $(\boldsymbol{u}, p, \lambda) \in \boldsymbol{X} \times M_0 \times \Lambda$ *such that for all* $(\boldsymbol{v}, q, \mu) \in \boldsymbol{X} \times M_0 \times \Lambda$

$$\begin{cases} a(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) + b_\Gamma(\boldsymbol{v}, \lambda) & = f(\boldsymbol{v}), \\ b(\boldsymbol{u}, q) & = 0, \\ b_\Gamma(\boldsymbol{u}, \mu) & = 0, \end{cases} \tag{2}$$

where

$$\boldsymbol{X} = \boldsymbol{X}^f \times \boldsymbol{X}^p := H_0^1(\Omega^f, \Gamma^f)^n \times \boldsymbol{H}_0(\text{div}, \Omega^p, \Gamma^p)$$

and M_0 is the subset of $M := L^2(\Omega^f) \times L^2(\Omega^p) \equiv L^2(\Omega)$ of pressures with a zero average value in Ω. Here $H_0^1(\Omega^f, \Gamma^f)$ denotes the subspace of $H^1(\Omega^f)$ of functions that vanish on Γ^f. The space $\boldsymbol{H}_0(\text{div}, \Omega^p, \Gamma^p)$ consists of functions in $\boldsymbol{H}(\text{div}, \Omega^p)$ with zero normal trace on Γ^p, where

$$\boldsymbol{H}(\text{div}, \Omega^p) := \left\{\boldsymbol{v} \in L^2(\Omega^p)^n : \text{div } \boldsymbol{v} \in L^2(\Omega^p)\right\}.$$

For the Lagrange multiplier space we consider $\Lambda := H^{1/2}(\Gamma)$. See [26] for a discussion on the choice of the Lagrange multipliers space Λ and how to derive the weak formulation (2) and other equivalent weak formulations; see also [31].

The global bilinear forms are

$$a(\boldsymbol{u}, \boldsymbol{v}) := a^f_{\alpha^f}(\boldsymbol{u}^f, \boldsymbol{v}^f) + a^p(\boldsymbol{u}^p, \boldsymbol{v}^p),$$
$$b(\boldsymbol{v}, p) := b^f(\boldsymbol{v}^f, p^f) + b^p(\boldsymbol{v}^p, p^p),$$

with local bilinear forms $a^f_{\alpha^f}$, b^f and b^p defined by

$$a^f_{\alpha^f}(\boldsymbol{u}^f, \boldsymbol{v}^f) := 2\nu(\boldsymbol{D}\boldsymbol{u}^f, \boldsymbol{D}\boldsymbol{v}^f)_{\Omega^f} + \sum_{\ell=1}^{n-1} \frac{\nu\alpha^f}{\sqrt{\kappa}} \langle \boldsymbol{u}^f \cdot \tau_\ell, \boldsymbol{v}^f \cdot \tau_\ell\rangle_\Gamma, \ \boldsymbol{u}^f, \boldsymbol{v}^f \in \boldsymbol{X}^f, \tag{3}$$

$$a^p(\boldsymbol{u}^p, \boldsymbol{v}^p) := ((\nu/\kappa)\boldsymbol{u}^p, \boldsymbol{v}^p)_{\Omega^p}, \quad \boldsymbol{u}^p, \boldsymbol{v}^p \in \boldsymbol{X}^p, \tag{4}$$

$$b^f(\boldsymbol{v}^f, q^f) := -(q^f, \nabla\cdot\boldsymbol{v}^f)_{\Omega^f}, \quad \boldsymbol{v}^f \in \boldsymbol{X}^f,\ q^f \in M^f, \tag{5}$$

$$b^p(\boldsymbol{v}^p, p^p) := -(p^p, \nabla\cdot\boldsymbol{v}^p)_{\Omega^p}, \quad \boldsymbol{v}^p \in \boldsymbol{X}^p,\ p^p \in M^p, \tag{6}$$

and with weak conservation of mass bilinear form defined by

$$b_\Gamma(\boldsymbol{v}, \mu) := \langle \boldsymbol{v}^f\cdot\eta^f, \mu\rangle_\Gamma + \langle \boldsymbol{v}^p\cdot\eta^p, \mu\rangle_\Gamma,\ \boldsymbol{v} = (\boldsymbol{v}^f, \boldsymbol{v}^p) \in \boldsymbol{X}, \mu \in \Lambda. \tag{7}$$

The second duality pairing of (7) is interpreted as $\langle \boldsymbol{v}^p\cdot\eta^p, E_{\eta^p}(\mu)\rangle_{\partial\Omega^p}$. Here E_{η^p} is any continuous lift-in operator from $H^{1/2}(\Gamma)$ to $H^{1/2}(\partial\boldsymbol{\Omega}^p)$; recall that $\Gamma \subset \partial\boldsymbol{\Omega}^p$ and $\boldsymbol{v} \in \boldsymbol{H}_0(\mathrm{div}, \Omega^p, \Gamma^p)$. It easy to see that this duality pairing is independent of the lift-in operator E_{η^p}. In particular, one example of such a lift-in operator can be constructed by taking the trace on $\partial\boldsymbol{\Omega}^p$ of the harmonic extension with Dirichlet data μ on Γ and homogeneous Neumann data on Γ^p; see [26].

The functional f in the right side of (2) is defined by

$$f(\boldsymbol{v}) := f^f(\boldsymbol{v}^f) + f^p(\boldsymbol{v}^p), \quad \text{for all } \boldsymbol{v} = (\boldsymbol{v}^f, \boldsymbol{v}^p) \in \boldsymbol{X},$$

where $f^i(\boldsymbol{v}^i) := (\boldsymbol{f}^i, \boldsymbol{v}^i)_{L^2(\Omega^i)}$ for all $\boldsymbol{v}^i \in \boldsymbol{X}^i$, $i = f, p$.

The bilinear forms $a^f_{\alpha^f}, b^f$ are associated to Stokes equations, and the bilinear forms a^p, b^p to Darcy law. The bilinear form $a^f_{\alpha^f}$ includes interface matching conditions $1.b$ and $1.c$ above. The bilinear form b_Γ is used to impose the weak version of the interface matching condition $1.a$ above. We have the following lemma that addresses the well-posedness of the problem.

Lemma 1 (See [26; 31]). *There exists $\beta > 0$ such that*

$$\inf_{\substack{(q,\mu)\in M_0\times\Lambda\\ (q,\mu)\neq 0}} \sup_{\substack{\boldsymbol{v}\in\boldsymbol{X}\\ \boldsymbol{v}\neq 0}} \frac{b(\boldsymbol{v}, q) + b_\Gamma(\boldsymbol{v}, \mu)}{\|\boldsymbol{v}\|_X\,(\|p\|_M + \|\mu\|_\Lambda)} \geq \beta > 0. \tag{8}$$

where

$$\|\boldsymbol{v}\|^2_{\boldsymbol{X}} := \|\boldsymbol{v}^f\|^2_{H^1_0(\Omega_f)^2} + \|\boldsymbol{v}^p\|^2_{\boldsymbol{H}(\mathrm{div},\Omega_p)}.$$

This inf-sup condition, together with the fact that $a^f_{\alpha^f}$ is $\boldsymbol{X}^f \times \boldsymbol{H}(\mathrm{div}^0, \Omega^p)$-elliptic and $a^f_{\alpha^f}$, b and b_Γ are bounded, guarantees the well-posedness of the problem (2).

4. Discretization

From now on we consider only the two-dimensional case. We note that the ideas developed in the following can be easily extended to case of three-dimensional subdomains.

We assume that $\boldsymbol{\Omega}^i$, $i = f, p$, are *two-dimensional* polygonal subdomains. Let $\mathcal{T}^i_{h^i}(\boldsymbol{\Omega}^i)$ be a geometrically conforming shape regular and quasiuniform triangulation of $\boldsymbol{\Omega}^i$ with mesh size parameter h^i, $i = f, p$. We do not assume that these two triangulations match at the interface Γ. For the fluid region, let $\boldsymbol{X}^f_{h^f}$ and $M^f_{h^f}$ be P2/P1 triangular Taylor–Hood finite elements; see [7; 8; 10]. More precisely,

$$\boldsymbol{X}^f_{h^f} := \left\{ \boldsymbol{u} \in \boldsymbol{X}^f \ : \ \begin{array}{l} \boldsymbol{u}_K = \hat{\boldsymbol{u}}_K \circ F_K^{-1} \text{ and } \hat{\boldsymbol{u}}_K \in P_2(\hat{K})^2 \\ \text{for all } K \in \mathcal{T}^f_{h^f}(\Omega_f) \end{array} \right\} \cap C^0(\overline{\Omega}^f)^2, \tag{9}$$

where $\boldsymbol{u}_K := \boldsymbol{u}|_K$ and

$$M^f_{h^f} := \left\{ p \in L^2(\Omega^f) \ : \ \begin{array}{l} p_K = \hat{p}_K \circ F_K^{-1} \text{ and } \hat{p}_K \in P_1(\hat{K}) \\ \text{for all } K \in \mathcal{T}^f_{h^f}(\Omega_f), \end{array} \right\} \cap C^0(\overline{\Omega}^f).$$

Denote by $\mathring{M}^f_{h^f} \subset M^f_{h^f}$ the discrete fluid pressures with zero average value in Ω^f. For the porous region, let $\boldsymbol{X}^p_{h^p} \subset \boldsymbol{X}^p$ and $M^p_{h^p} \subset L^2(\Omega^p)$ be the lowest order Raviart–Thomas finite elements based on triangles; see [7; 10]. Let $\mathring{M}^p_{h^p} \subset M^p_{h^p}$ be the subset of pressures in $M^p_{h^p}$ with zero average value in Ω^p.

Define $\boldsymbol{X}_h := \boldsymbol{X}^f_{h^f} \times \boldsymbol{X}^p_{h^p} \subset \boldsymbol{X}$ and $M_h := M^f_{h^f} \times M^p_{h^p} \subset L^2(\Omega^f) \times L^2(\Omega^p)$. Note that in the definition of the discrete velocities we assume that the boundary conditions are included, that is, for $\boldsymbol{v}^f_{h^f} \in \boldsymbol{X}^f_{h^f}$, we have $\boldsymbol{v}^f_{h^f} = \boldsymbol{\Gamma}$ on Γ^f and for $\boldsymbol{v}^p_{h^p} \in \boldsymbol{X}^p_{h^p}$ we have that $\boldsymbol{v}^p_h \cdot \eta^p = 0$ on Γ^p.

Let $\mathcal{T}^p_{h^p}(\Gamma)$ be the restriction to Γ of the porous side triangulation $\mathcal{T}^p_{h^p}(\Omega^p)$. For the Lagrange multipliers space we choose piecewise constant functions on Γ with respect to the triangulation $\mathcal{T}^p_{h^p}(\Gamma)$:

$$\Lambda_{h^p} := \left\{ \lambda \ : \ \lambda|_{e^p_j} = \lambda_{e^p_j} \text{ is constant in each edge } e^p_j \text{ of } \mathcal{T}^p_{h^p}(\Gamma) \right\}, \tag{10}$$

that is, the *master* is on the fluid region side and the *slave* is on the porous region side; see [5; 6; 19; 45]. The choice of piecewise constant Lagrange multipliers leads to a nonconforming approximation on Λ_{h^p} since piecewise constant functions do not belong to $H^{1/2}(\Gamma)$. For the analysis of this nonconforming discretization and a priori error estimates we refer to [26].

5. Primal and dual formulations

In order to simplify the notation and since there is no danger of confusion, we will denote the finite element functions and the corresponding vector representation by the same symbol, that is, when writing finite element functions we will drop the indices h^i. Recall that we have the pair of spaces $(\boldsymbol{X}_h, M_h)$ associated to the coupled problem, and spaces associated to each subproblem: $(\boldsymbol{X}^f_{h^f}, M^f_{h^f})$ and $(\boldsymbol{X}^p_{h^p}, M^p_{h^p})$. We will keep the subscript h^i, $i = f, p$, in the notation for local subspaces $\boldsymbol{X}^f_{h^f}$, $M^f_{h^f}$, $\boldsymbol{X}^p_{h^p}$ and $M^p_{h^p}$.

Since we are interested in preconditioning issues we assume $\alpha^f = 0$ in the definition of the fluid side local bilinear form $a^f_{\alpha^f}$ in (3). We denote $a^f = a^f_0$. See Remark 10 for the case $\alpha^f > 0$.

With the discretization chosen in Section 4 we obtain the following symmetric saddle point linear system

$$\left[\begin{array}{cc|cc|c} A^f & B^{fT} & 0 & 0 & C^{fT} \\ B^f & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & A^p & B^{pT} & -C^{pT} \\ 0 & 0 & B^p & 0 & 0 \\ \hline C^f & 0 & -C^p & 0 & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{u}^f \\ p^f \\ \hline \boldsymbol{u}^p \\ p^p \\ \hline \lambda \end{array}\right] = \left[\begin{array}{c} \boldsymbol{f}^f \\ g^f \\ \hline \boldsymbol{f}^p \\ g^p \\ \hline 0 \end{array}\right], \tag{11}$$

with matrices A^i, B^i, C^i and columns vectors $\boldsymbol{f}^i, g^i, i = f, p$, defined by

$$\begin{array}{lcl} a^i(\boldsymbol{u}^i, \boldsymbol{v}^i) & = & \boldsymbol{v}^{iT} A^i \boldsymbol{u}^i, \\ b^i(\boldsymbol{u}^i, q^i) & = & q^{iT} B^i \boldsymbol{u}^i, \\ (\boldsymbol{u}^i \cdot \eta^f, \mu)_\Gamma & = & \mu^T C^i \boldsymbol{u}^i, \\ f^i(\boldsymbol{v}^i) & = & \boldsymbol{v}^{iT} \boldsymbol{f}^i, \\ g^i(q^i) & = & q^{iT} g^i. \end{array} \tag{12}$$

The matrix A^f corresponds to ν times the discrete version of the linearized stress tensor on Ω^f. Note that in the case $\alpha^f > 0$, the bilinear form $a^f_{\alpha^f}$ in (3) includes a boundary term; see Remark 10. The matrix A^p corresponds to ν/κ times a discrete L^2-norm on Ω^p. Matrix $-B^i$ is the discrete divergence in Ω^i, $i = f, p$, and matrices C^f and C^p correspond to the matrix form of the discrete conservation of mass on Γ. Note that ν can be viewed as a scaling factor since it appears in both matrices A^f and A^p. Therefore, it is not relevant for preconditioning issues.

Consider the following partition of the degrees of freedom: for $i = f, p$, let

$$\left[\begin{array}{c} \boldsymbol{u}^i_I \\ p^i_I \\ u^i_\Gamma \\ \bar{p}^i \end{array}\right] \begin{array}{l} \text{interior displacements + tangential velocities on } \Gamma, \\ \text{interior pressures with zero average in } \Omega^i, \\ \text{interface outward } \textit{normal velocities} \text{ on } \Gamma, \\ \text{constant pressure in } \Omega^i. \end{array}$$

For $i = f, p$, we have the block structure

$$A^i = \begin{bmatrix} A^i_{II} & A^{iT}_{\Gamma I} \\ A^i_{\Gamma I} & A^i_{\Gamma\Gamma} \end{bmatrix}, \quad B^i = \begin{bmatrix} B^i_{II} & B^{iT}_{\Gamma I} \\ 0 & \bar{B}^{iT} \end{bmatrix} \quad \text{and } C^i = \begin{bmatrix} 0 & 0 & \tilde{C}^i & 0 \end{bmatrix}.$$

Note that the (2, 1) entry of B^i corresponds to integrating an *interior* velocity against a constant pressure, then it vanishes due to the divergence theorem. We have the following matrix representation of the coupled problem in (11):

$$\left[\begin{array}{cccc|cccc|c}
A^f_{II} & B^{fT}_{II} & A^{fT}_{\Gamma I} & 0 & 0 & 0 & 0 & 0 & 0\\
B^f_{II} & 0 & B^{fT}_{\Gamma I} & 0 & 0 & 0 & 0 & 0 & 0\\
A^f_{\Gamma I} & B^{fT}_{I\Gamma} & A^f_{\Gamma\Gamma} & \bar{B}^{fT} & 0 & 0 & 0 & 0 & \tilde{C}^{fT}\\
0 & 0 & \bar{B}^f & 0 & 0 & 0 & 0 & 0 & 0\\
\hline
0 & 0 & 0 & 0 & A^p_{II} & B^{pT}_{II} & A^{pT}_{\Gamma I} & 0 & 0\\
0 & 0 & 0 & 0 & B^p_{II} & 0 & B^p_{I\Gamma} & 0 & 0\\
0 & 0 & 0 & 0 & A^p_{\Gamma I} & B^{pT}_{I\Gamma} & A^p_{\Gamma\Gamma} & \bar{B}^{pT} & -\tilde{C}^{pT}\\
0 & 0 & 0 & 0 & 0 & 0 & \bar{B}^p & 0 & 0\\
\hline
0 & 0 & \tilde{C}^f & 0 & 0 & 0 & -\tilde{C}^p & 0 & 0
\end{array}\right]
\left[\begin{array}{c}
\boldsymbol{u}^f_I\\ p^f_I\\ u^f_\Gamma\\ \bar{p}^f\\ \hline \boldsymbol{u}^p_I\\ p^p_I\\ u^p_\Gamma\\ \bar{p}^p\\ \hline \lambda
\end{array}\right]
=
\left[\begin{array}{c}
\boldsymbol{f}^f_I\\ g^f_I\\ f^f_\Gamma\\ \bar{g}^f\\ \hline \boldsymbol{f}^p_I\\ g^p_I\\ f^p_\Gamma\\ \bar{g}^p\\ \hline 0
\end{array}\right]. \tag{13}$$

Following [19; 40], we choose the following matrix representation in each subdomain Ω^i, $i = f, p$:

$$\left[\begin{array}{cc|cc}
A^i_{II} & B^{iT}_{II} & A^{iT}_{\Gamma I} & 0\\
B^i_{II} & 0 & B^i_{I\Gamma} & 0\\
\hline
A^i_{\Gamma I} & B^{iT}_{I\Gamma} & A^i_{\Gamma\Gamma} & \bar{B}^{iT}\\
0 & 0 & \bar{B}^i & 0
\end{array}\right]
=
\left[\begin{array}{c|c}
K^i_{II} & K^{iT}_{\Gamma I}\\
\hline
K^i_{\Gamma I} & K^i_{\Gamma\Gamma}
\end{array}\right]. \tag{14}$$

5.1. ***The primal formulation.*** From the last equation in (13) we see that the mortar condition on Γ (using the Darcy side as the slave side) can be imposed as $u^p_\Gamma = (\tilde{C}^p)^{-1}\tilde{C}^f u^f_\Gamma = \Pi u^f_\Gamma$, where Π is the $L^2(\Gamma)$ projection on the space of piecewise constant functions on each subinterval $e^p \in \mathcal{T}^p_{h^p}(\Gamma)$. We note that $\tilde{C}^p$ is a diagonal matrix for the lowest order Raviart–Thomas elements.

Now we eliminate $\boldsymbol{u}^i_I$, p^i_I, $i = f, p$, and λ, to obtain the following (saddle point) Schur complement:

$$S\begin{bmatrix} u^f_\Gamma\\ \bar{p}^f\\ \bar{p}^p \end{bmatrix} = \begin{bmatrix} b_\Gamma\\ \bar{b}^f\\ \bar{b}^p \end{bmatrix}. \tag{15}$$

Here S is given by

$$S := \begin{bmatrix} S^f_\Gamma & \bar{B}^{fT} & 0\\ \bar{B}^f & 0 & 0\\ 0 & 0 & 0 \end{bmatrix} + \tilde{\Pi}^T \begin{bmatrix} S^p_\Gamma & 0 & \bar{B}^{pT}\\ 0 & 0 & 0\\ \bar{B}^p & 0 & 0 \end{bmatrix} \tilde{\Pi} = \tilde{S}^f + \tilde{S}^p$$

$$= \left[\begin{array}{c|cc}
S^f_\Gamma + \Pi^T S^p_\Gamma \Pi & \bar{B}^{fT} & \Pi^T \bar{B}^{pT}\\
\hline
\bar{B}^f & 0 & 0\\
\bar{B}^p \Pi & 0 & 0
\end{array}\right] = \begin{bmatrix} S_\Gamma & \bar{B}^T\\ \bar{B} & 0 \end{bmatrix},$$

where

$$\tilde{\Pi} := \begin{bmatrix} \Pi & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{and } \bar{B}^T := [\bar{B}^{fT} \; \Pi^T \bar{B}^{pT}]. \tag{16}$$

Here, we have introduced

$$\tilde{S}^f := \begin{bmatrix} S^f_\Gamma & \bar{B}^{fT} & 0 \\ \bar{B}^f & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad \tilde{S}^p := \tilde{\Pi}^T \begin{bmatrix} S^p_\Gamma & 0 & \bar{B}^{pT} \\ 0 & 0 & 0 \\ \bar{B}^p & 0 & 0 \end{bmatrix} \tilde{\Pi} \tag{17}$$

and

$$S_\Gamma := S^f_\Gamma + \Pi^T S^p_\Gamma \Pi. \tag{18}$$

The local matrices S^i_Γ and $\bar{B}^i$ and the local Schur complement S^i are given by

$$S^i = \begin{bmatrix} S^i_\Gamma & \bar{B}^{iT} \\ \bar{B}^i & 0 \end{bmatrix} := K^i_{\Gamma\Gamma} - K^i_{\Gamma I} \big(K^i_{II}\big)^{-1} K^{iT}_{\Gamma I}, \; i = p, f. \tag{19}$$

The right side of (15) is given by

$$\begin{aligned} \begin{bmatrix} b_\Gamma \\ \bar{b}^f \\ \bar{b}^p \end{bmatrix} = & \left\{ \begin{bmatrix} f^f_\Gamma \\ \bar{g}^f \\ 0 \end{bmatrix} - \begin{bmatrix} K^f_{\Gamma I} \big(K^f_{II}\big)^{-1} \begin{bmatrix} \boldsymbol{f}^f_I \\ g^f_I \end{bmatrix} \\ 0 \end{bmatrix} \right\} \\ & + \left\{ \begin{bmatrix} \Pi^T f^p_\Gamma \\ 0 \\ \bar{g}^p \end{bmatrix} - \tilde{\Pi}^T \begin{bmatrix} K^p_{\Gamma I} \big(K^p_{II}\big)^{-1} \begin{bmatrix} \boldsymbol{f}^p_I \\ g^p_I \end{bmatrix} \\ 0 \end{bmatrix} \right\}. \end{aligned}$$

We note that the reduced system (15), as well as the original system (13), is solvable when $\bar{b}^f + \bar{b}^p = 0$, and the solution is unique when we restrict to pressures with zero average value on Ω.

From now on we only work with functions defined on Γ and extended inside the subdomain using the discrete Stokes and Darcy problems. It is convenient to define the space

$$V_\Gamma := \left\{ v_\Gamma = ({v^f}_\Gamma, v^p_\Gamma) : {v^f}_\Gamma = \mathcal{SH}(\boldsymbol{v}^f \cdot \eta^f|_\Gamma) \text{ and } v^p_\Gamma = \mathcal{DH}(\boldsymbol{v}^p \cdot \eta^p|_\Gamma))\right\} \tag{20}$$

and

$$\begin{aligned} M^h_0 := \Big\{ q \in M^h : q^i = \text{piecewise constant in } \Omega^i \text{ for } i = f, p, \\ \text{and } \int_{\Omega^f} q^f + \int_{\Omega^p} q^p = 0 \Big\}. \end{aligned} \tag{21}$$

Here $\mathscr{SH}$ ($\mathscr{DH}$) is the velocity component of the discrete Stokes (Darcy) harmonic extension operator that maps discrete interface normal velocity $u_\Gamma^f \in H_{00}^{1/2}(\Gamma)$ (respectively $u_\Gamma^p \in (H^{1/2}(\Gamma))'$) to the solution of following problem: Find $\boldsymbol{u}^i \in \boldsymbol{X}_{h^i}^i$ and $p^i \in \mathring{M}_{h^i}^i$ such that for all $\boldsymbol{v}^i \in \boldsymbol{X}_{h^i}^i$ and $q^i \in \mathring{M}_{h^i}^i$, $i = f, p$, we have

$$\left\{\begin{array}{rll} a^f(\mathscr{SH}u^f, \boldsymbol{v}^f) + b^f(\boldsymbol{v}^f, p^f) & = 0, & \\ b^f(\mathscr{SH}u^f, q^f) & = 0, & \\ \mathscr{SH}u^f \cdot \eta^f & = u_\Gamma^f & \text{on } \Gamma, \\ \mathscr{SH}u^f & = \mathbf{\Gamma} & \text{on } \Gamma^f, \end{array}\right. \tag{22}$$

and

$$\left\{\begin{array}{rll} a^p(\mathscr{DH}u^p, \boldsymbol{v}^p) + b^p(\boldsymbol{v}^p, p^p) & = 0, & \\ b^p(\mathscr{DH}u^p, q^p) & = 0, & \\ \mathscr{DH}u^p \cdot \eta^p & = u_\Gamma^p & \text{on } \Gamma, \\ \mathscr{DH}u^p \cdot \eta^p & = 0 & \text{on } \Gamma^p. \end{array}\right. \tag{23}$$

The degrees of freedom associated with $\mathscr{SH}u^f \cdot \tau^f$ on Γ are free. This corresponds to imposing the natural boundary condition $\tau^T \boldsymbol{D}(\mathscr{SH}u^f)\eta_f = 0$ on Γ which is the expression for interface condition of Beavers–Joseph–Saffman with $\alpha^f = 0$.

For $i = f, p$, define the normal trace component of $\boldsymbol{X}_{h^i}^i$ by

$$Z_{h^i}^i = \left\{\boldsymbol{v}^i \cdot \eta^i|_\Gamma \ : \ \boldsymbol{v}^i \in \boldsymbol{X}_{h^i}^i\right\}. \tag{24}$$

Associated with the coupled problem (13) we introduce the *balanced subspace*:

$$V_{\Gamma,\bar{B}} := \left\{ v^f{}_\Gamma \in Z_{h^f}^f \ : \ (v^f{}_\Gamma, \Pi v_\Gamma^f) \in V_\Gamma \text{ and } \int_\Gamma v^f{}_\Gamma \cdot \eta_f = 0 \right\}, \tag{25}$$

with V_Γ defined in (20); see [40]. Observe that $V_{\Gamma,\bar{B}} = \operatorname{Ker}\bar{B}$, where $\bar{B}$ is defined in (16) and (19). Then for $v_\Gamma^f \in V_{\Gamma,\bar{B}}$ we have $\bar{B}v_\Gamma^f = 0$. We will refer to functions $v_\Gamma^f \in V_{\Gamma,\bar{B}}$ as *balanced functions*. If $v_\Gamma^p = \Pi v_\Gamma^f$ and v_Γ^f is a balanced function, then we also say that v_Γ^p is a balanced function or the pair $(v_\Gamma^f, \Pi v^f{}_\Gamma)$ is balanced.

5.2. *Dual formulation.* In the system (13), we first eliminate the unknowns $\boldsymbol{u}_I^f$, p_I^f and $\boldsymbol{u}_I^p$, p_I^p. We obtain

$$\left[\begin{array}{cc|cc|c} S_\Gamma^f & \bar{B}^{fT} & 0 & 0 & \tilde{C}^{fT} \\ \bar{B}^f & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & S_\Gamma^p & \bar{B}^{pT} & -\tilde{C}^{pT} \\ 0 & 0 & \bar{B}^p & 0 & 0 \\ \hline \tilde{C}^f & 0 & -\tilde{C}^p & 0 & 0 \end{array}\right] \left[\begin{array}{c} u_\Gamma^f \\ \bar{p}^f \\ \hline u_\Gamma^p \\ \bar{p}^p \\ \hline \lambda \end{array}\right] = \left[\begin{array}{c} \tilde{b}^f \\ \hline \tilde{b}^p \\ \hline 0 \end{array}\right], \tag{26}$$

where the right side of (26) is given by

$$\left[\begin{array}{c}\tilde{b}^f\\ \hline \tilde{b}^p\\ \hline 0\end{array}\right]=\left[\begin{array}{c}\begin{bmatrix}f_\Gamma^f\\ \bar{g}^f\end{bmatrix}-K_{\Gamma I}^f\left(K_{II}^f\right)^{-1}\begin{bmatrix}f_I^f\\ g_I^f\end{bmatrix}\\ \hline \begin{bmatrix}f_\Gamma^p\\ \bar{g}^p\end{bmatrix}-K_{\Gamma I}^p\left(K_{II}^p\right)^{-1}\begin{bmatrix}f_I^p\\ g_I^p\end{bmatrix}\\ \hline 0\end{array}\right].$$

Here S_Γ^i, K_{II}^i and $K_{I\Gamma}^i$, $i=f,p$, are defined in (19) and (14).

Let $N_i:=\begin{bmatrix}\tilde{C}^i & 0\end{bmatrix}$ and consider S^i, $i=f,p$, defined in (19). Then the matrix in the left side of (26) can be rewritten as

$$\left[\begin{array}{c|c|c}S^f & 0 & N^{fT}\\ \hline 0 & S^p & -N^{pT}\\ \hline N^f & -N^p & 0\end{array}\right].$$

Now we eliminate the unknowns u_Γ^f, $\bar{p}^f$ and u_Γ^p, $\bar{p}^p$. We end up with the reduced system

$$F\lambda=c, \tag{27}$$

where the operator F is defined by

$$F:=N^f(S^f)^{-1}N^{fT}+N^p(S^p)^{-1}N^{pT}, \tag{28}$$

and the right side c is given by

$$c=N^f(S^f)^{-1}\left\{\begin{bmatrix}f_\Gamma^f\\ \bar{g}^f\end{bmatrix}-K_{\Gamma I}^f(K_{II}^f)^{-1}\begin{bmatrix}f_I^f\\ g_I^f\end{bmatrix}\right\}\\ -N^p(S^p)^{-1}\left\{\begin{bmatrix}f_\Gamma^p\\ \bar{g}^p\end{bmatrix}-K_{\Gamma I}^p(K_{II}^p)^{-1}\begin{bmatrix}f_I^f\\ g_I^p\end{bmatrix}\right\}.$$

Note that F is positive semidefinite and since a discrete Lagrange multiplier in Λ_{h^p} does not have necessarily zero mean average value on Γ, the operator F has one simple zero eigenvalue corresponding to a constant Lagrange multiplier. The linear system above, as well as the original linear system (13), is solvable for zero mean right side, that is, for $c^T\cdot(1,\dots,1)=0$.

6. BDD preconditioner

In this section we design and analyze a BDD type preconditioner for the Schur complement system (15); see [9; 19; 42] and also [1; 21; 35; 40; 43]. For the sake of simplicity on the analysis we assume that $\Gamma = \{1\} \times (0,1)$, $\Omega^f = (1,2) \times (0,1)$ and $\Omega^p = (0,1) \times (0,1)$. We introduce the velocity coarse space on Γ as the span of the normal velocity $v_0 = y(1-y)$ (with v_0 also denoting its vector representation). Define

$$R_0 := \begin{bmatrix} v_0^T & 0 \\ 0 & I_{2\times 2} \end{bmatrix}, \quad S_0 := R_0 S R_0^T \quad \text{and} \quad Q_0 := R_0^T S_0^{\dagger} R_0. \tag{29}$$

The system (15) is solvable when the right side satisfies $\bar{b}^f + \bar{b}^p = 0$ with uniqueness of the solution in the space of vectors with pressure component having zero average value on Ω. Then S_0 is invertible restricted to vectors with pressure component in M_0^h defined in (21). The low dimensionality of the coarse space (which is spanned by v_0 and a constant pressure per subdomain Ω^i, $i = f, p$) and the fact that the function v_0 is independent of the triangulation parameters imply stable discrete inf-sup condition for the coarse problem.

Denote $\tilde{S}_0 := v_0^T S_\Gamma v_0$ and $\tilde{S} := \bar{B} v_0 \tilde{S}_0^{-1} v_0^T \bar{B}^T$. We can write, see (18) and (29),

$$S_0 = \begin{bmatrix} \tilde{S}_0 & (\bar{B}v_0)^T \\ \bar{B}v_0 & 0 \end{bmatrix}.$$

A simple calculation using the formula for the inverse of a saddle point matrix gives

$$Q_0 = \begin{bmatrix} v_0\tilde{S}_0^{-1}v_0^T - v_0\tilde{S}_0^{-1}v_0^T\bar{B}^T\tilde{S}^{-1}\bar{B}v_0\tilde{S}_0^{-1}v_0^T & v_0\tilde{S}_0^{-1}v_0^T\bar{B}^T\tilde{S}^{-1} \\ \tilde{S}^{-1}\bar{B}v_0\tilde{S}_0^{-1}v_0^T & \tilde{S}^{-1} \end{bmatrix},$$

and using (18) we obtain

$$Q_0 S = \begin{bmatrix} v_0\tilde{S}_0^{-1}v_0^T S_\Gamma - v_0\tilde{S}_0^{-1}v_0^T\bar{B}^T\tilde{S}^{-1}\bar{B}v_0\tilde{S}_0^{-1}v_0^T S_\Gamma + v_0\tilde{S}_0^{-1}v_0^T\bar{B}^T\tilde{S}^{-1}\bar{B} & 0 \\ \tilde{S}^{-1}\bar{B}v_0\tilde{S}_0^{-1}v_0^T S_\Gamma - \tilde{S}^{-1}\bar{B} & I \end{bmatrix},$$

or

$$Q_0 S = \begin{bmatrix} \mathcal{P} & 0 \\ \mathcal{G} & I \end{bmatrix},$$

where we have defined

$$\mathcal{P} := \left(v_0\tilde{S}_0^{-1}v_0^T S_\Gamma - v_0\tilde{S}_0^{-1}v_0^T\bar{B}^T\tilde{S}^{-1}\bar{B}v_0\tilde{S}_0^{-1}v_0^T S_\Gamma\right) + v_0\tilde{S}_0^{-1}v_0^T\bar{B}^T\tilde{S}^{-1}\bar{B},$$
$$\mathcal{G} := \tilde{S}^{-1}\bar{B} - \tilde{S}^{-1}\bar{B}v_0\tilde{S}_0^{-1}v_0^T S_\Gamma.$$

With this notation we have that

$$I - Q_0 S = \begin{bmatrix} I - \mathcal{P} & 0 \\ \mathcal{G} & 0 \end{bmatrix}.$$

Elementary calculations show that $\mathcal{P}^2 = \mathcal{P}$ and $\bar{B}(I-\mathcal{P}) = 0$, hence $I-\mathcal{P}$ is a projection and its image is contained on the balanced subspace defined in (25); see also [40].

Given a residual $r = \begin{bmatrix} f_\Gamma^T & \bar{g}^T \end{bmatrix}^T$, the coarse problem $Q_0 r$, with Q_0 defined in (29), is the solution of the coupled problem (13) with one velocity degree of freedom (v_0), and a constant pressure per subdomain Ω^i, $i = f, p$, with mean zero in $\Omega = \text{int}(\overline{\Omega}^f \cup \overline{\Omega^p})$. Note that the matrix S_0 defined in (29) can be computed easily, and in order to ensure zero mean pressure on Ω we can use a Lagrange multiplier.

For balanced functions v_Γ^f and u_Γ^f, the S_Γ-inner product (see (18)) is defined by

$$\langle u_\Gamma^f, v_\Gamma^f \rangle_{S_\Gamma} := \langle S_\Gamma u_\Gamma^f, v_\Gamma^f \rangle = u_\Gamma^{fT} S_\Gamma v_\Gamma^f.$$

Recall that $\bar{B} u_\Gamma^f = 0$ when u_Γ^f is balanced. Then, on this subspace of balanced functions, the S_Γ inner product coincides with the S-inner product defined by

$$\left\langle \begin{bmatrix} v_\Gamma^f \\ \bar{q}^f \\ \bar{q}^p \end{bmatrix}, \begin{bmatrix} u_\Gamma^f \\ \bar{p}^f \\ \bar{p}^p \end{bmatrix} \right\rangle_S := \begin{bmatrix} v_\Gamma^f \\ \bar{q}^f \\ \bar{q}^p \end{bmatrix}^T S \begin{bmatrix} u_\Gamma^f \\ \bar{p}^f \\ \bar{p}^p \end{bmatrix} = \begin{bmatrix} v_\Gamma^f \\ \bar{q} \end{bmatrix}^T \begin{bmatrix} S_\Gamma & \bar{B}^T \\ \bar{B} & 0 \end{bmatrix} \begin{bmatrix} u_\Gamma^f \\ \bar{p} \end{bmatrix},$$

where $\bar{p}^T = \begin{bmatrix} \bar{p}^p & \bar{p}^p \end{bmatrix}^T$. Consider the BDD preconditioner operator given by

$$S_N^{-1} := Q_0 + (I - Q_0 S)\,(\tilde{S}^f)^\dagger\,(I - S Q_0)\,, \tag{30}$$

where $\tilde{S}^f$ is defined in (17); see [19; 40]. The notation $(\tilde{S}^f)^\dagger$ stands for the pseudo-inverse of $\tilde{S}^f$, that is,

$$(\tilde{S}^f)^\dagger = \begin{bmatrix} (S^f)^{-1} & 0 \\ 0 & 0 \end{bmatrix},$$

with S^f defined in (19). The preconditioned operator is given by

$$\begin{aligned} S_N^{-1} S &= Q_0 S + (I - Q_0 S)\,(\tilde{S}^f)^\dagger S\,(I - Q_0 S) \\ &= \begin{bmatrix} \mathcal{P} & 0 \\ \mathcal{G} & I \end{bmatrix} + \begin{bmatrix} I-\mathcal{P} & 0 \\ \mathcal{G} & 0 \end{bmatrix} (\tilde{S}^f)^\dagger \begin{bmatrix} S_\Gamma & \bar{B}^T \\ \bar{B} & 0 \end{bmatrix} \begin{bmatrix} I-\mathcal{P} & 0 \\ \mathcal{G} & 0 \end{bmatrix}. \end{aligned} \tag{31}$$

Note that applying $(S^f)^{-1}$ to a vector

$$\begin{bmatrix} u_\Gamma^f \\ \bar{p} \end{bmatrix}$$

is equivalent to solving the linear system

$$\begin{bmatrix} A_{II}^{f} & B_{II}^{fT} & A_{\Gamma I}^{fT} & 0 \\ B_{II}^{f} & 0 & B_{I\Gamma}^{f} & 0 \\ A_{\Gamma I}^{f} & B_{I\Gamma}^{fT} & A_{\Gamma\Gamma}^{f} & \bar{B}^{fT} \\ 0 & 0 & \bar{B}^{f} & 0 \end{bmatrix} \begin{bmatrix} w_I^f \\ s_I^f \\ w_\Gamma^f \\ \bar{s}^f \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ u_\Gamma^f \\ \bar{p}^f \end{bmatrix}.$$

If u_Γ^f is balanced, so is the velocity component of

$$(S^f)^{-1} \begin{bmatrix} u_\Gamma^f \\ \bar{p}^f \end{bmatrix}.$$

Using elementary calculations with the matrices in (31) we obtain

$$\left\langle S_N^{-1} S \left[\begin{array}{c} u_\Gamma \\ \bar{p} \end{array} \right], \left[\begin{array}{c} v_\Gamma \\ \bar{q} \end{array} \right] \right\rangle_S = \langle (S_\Gamma^f)^{-1} S_\Gamma u_\Gamma, v_\Gamma \rangle_{S_\Gamma},$$

for $u_\Gamma, v_\Gamma \in \text{Range}(I - \mathcal{P})$. In order to bound the condition number of the preconditioned operator $S_N^{-1} S$, we need only analyze the condition of the operator $(S_\Gamma^f)^{-1} S_\Gamma$. Note that

$$c \langle u_\Gamma^f, u_\Gamma^f \rangle_{S_\Gamma} \leq \left\langle \left(S^f\right)^{-1} S_\Gamma u_\Gamma^f, u_\Gamma^f \right\rangle_{S_\Gamma} \leq C \langle u_\Gamma^f, u_\Gamma^f \rangle_{S_\Gamma}$$

is equivalent to

$$c \langle S^f u_\Gamma^f, u_\Gamma^f \rangle \leq \langle S_\Gamma u_\Gamma^f, u_\Gamma^f \rangle \leq C \langle S^f u_\Gamma^f, u_\Gamma^f \rangle. \tag{32}$$

The next theorem shows that the condition number estimate for the BDD method introduced in (30) is of order $O(1 + (1/\kappa))$, where κ is the permeability of the porous medium; see (1).

Theorem 2. *If u_Γ^f is a balanced function then*

$$\langle S_\Gamma^f u_\Gamma^f, u_\Gamma^f \rangle \leq \langle S_\Gamma u_\Gamma^f, u_\Gamma^f \rangle \prec \left(1 + \frac{1}{\kappa}\right) \langle S_\Gamma^f u_\Gamma^f, u_\Gamma^f \rangle.$$

Proof. The lower bound follows trivially from $\tilde{S}_\Gamma^f$ and $\tilde{S}_\Gamma^p$ being positive on the subspace of balanced functions. Next we concentrate on the upper bound.

Let v_Γ^f be a balanced function and $v_\Gamma^p = \Pi v_\Gamma^f$. Define $\boldsymbol{v}^p = \mathcal{DH} v_\Gamma^p$; see (23). Using properties of the discrete operator $\mathcal{DH}$ [38] we obtain

$$\langle S_\Gamma^p v_\Gamma^p, v_\Gamma^p \rangle = a^p(\boldsymbol{v}^p, \boldsymbol{v}^p) \asymp \frac{\nu}{\kappa} \| v_\Gamma^p \|^2_{(H^{1/2})'(\Gamma)}.$$

Using the L_2-stability property of mortar projection Π, we have

$$\| v_\Gamma^p \|^2_{(H^{1/2})'(\Gamma)} \prec \| v_\Gamma^p \|^2_{L^2(\Gamma)} = \| v_\Gamma^f \|^2_{L^2(\Gamma)} \prec \| v_\Gamma^f \|^2_{H_{00}^{1/2}(\Gamma)}.$$

With $\mathscr{SH}$ defined in (22), define $\boldsymbol{v}^f = \mathscr{SH} v_\Gamma^f$. Using properties of $\mathscr{SH}$ [40], we have

$$\nu \| v_\Gamma^f \|^2_{H_{00}^{1/2}(\Gamma)} \asymp a^f(\boldsymbol{v}^f, \boldsymbol{v}^f)$$

and then

$$\langle S_\Gamma^p v_\Gamma^p, v_\Gamma^p \rangle \prec \frac{1}{\kappa} \langle S^f u_\Gamma^f, u_\Gamma^f \rangle. \tag{33}$$

This gives the upper bound and finishes the proof. □

Recall that we consider the preconditioned projected conjugate gradient method applied to the Schur complement problem (15). Here is the algorithm:

(1) Initialize

$$\begin{aligned} x^{(0)} &= Q_0 b + w \\ d^{(0)} &= b - S x^{(0)} \end{aligned}$$

with $w \in \mathrm{Range}(I - Q_0 S)$. Recall that all vectors have three components, for instance,

$$x = \begin{bmatrix} x_\Gamma \\ \bar{x}^f \\ \bar{x}^p \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} b_\Gamma \\ \bar{b}^f \\ \bar{b}^p \end{bmatrix}.$$

(2) Iterate $k = 1, 2, \ldots$ until convergence

$$\begin{aligned} \text{Precondition: } z^{(k-1)} &= (\tilde{S}^f)^\dagger d^{(k-1)}, \\ \text{Project: } y^{(k-1)} &= (I - Q_0 S) z^{(k-1)} \\ \beta^k &= \langle y^{(k-1)}, d^{(k-1)} \rangle / \langle y^{(k-2)}, d^{(k-1)} \rangle \ \ [\beta^{(1)} = 0], \\ r^{(k)} &= y^{(k-1)} + \beta^{(k)} r^{(k)} \ [r^{(1)} = y^{(0)}], \\ \alpha^{(k)} &= \langle y^{(k-1)}, d^{(k-1)} \rangle / \langle d^{(k)}, S r^{(k)} \rangle, \\ x^{(k)} &= x^{(k-1)} + \alpha^{(k)} r^{(k)}, \\ d^{(k)} &= d^{(k-1)} - \alpha^{(k)} S r^{(k)}. \end{aligned}$$

Implementation of the projected preconditioned conjugate gradient algorithm for the system (15) *involving the BDD preconditioner* (30).

7. FETI preconditioner

In this section we analyze a FETI preconditioner for the reduced linear system (27); see [9; 19; 42; 24; 30; 37]. Recall the definition of F in (28). We propose the following preconditioner

$$(N^p)^\dagger (S^p) (N^p)^{\dagger T}, \tag{34}$$

where $(N^p)^\dagger$ is the pseudo-inverse $(N^p)^\dagger = [(\tilde{C}^p)^{-1} \ 0]$.

Note that after computing the action of $(S^f)^{-1}$ and $(S^p)^{-1}$ in the application of F to a zero average Lagrange multiplier, we end up with balanced functions. Therefore, to apply the preconditioned operator $(N^p)^{\dagger}(S^p)(N^p)^{\dagger T}F$ to a zero mean Lagrange multiplier, we do not need to solve a coarse problem at the beginning of the CG, nor inside of the CG iteration.

The FETI preconditioner in (34) can be considered as the dual preconditioner of the BDD preconditioner defined in (30); see the proof of Lemma 3 below.

Recall the definition of S^i, $i = f, p$, in (19) and the definition of space of balanced functions $V_\Gamma = V_\Gamma^f \times V_\Gamma^p$ in (25) and (24). We prove the following result.

Lemma 3. *Let $\lambda \in \Lambda_{h^p} \cap L_0^2(\Gamma)$ be a zero mean Lagrange multiplier. Then*

$$\langle N^f (S^f)^{-1} N^{fT}\lambda, \lambda\rangle \prec \frac{1}{\kappa}\langle N^p (S^p)^{-1} N^{pT}\lambda, \lambda\rangle.$$

Proof. Consider a zero mean Lagrange multiplier λ. Define $t = (S_\Gamma^p)^{-1/2}\tilde{C}^{pT}\lambda$ and $w^f = \tilde{C}^{fT}\lambda$. Then it is enough to prove that

$$\|(S_\Gamma^f)^{-1/2}w^f\|^2 \prec \|t\|^2.$$

Since w^f is balanced, that is, $w^f \in V_\Gamma^f$, we have that

$$\begin{aligned}\|(S_\Gamma^f)^{-1/2}w^f\|^2 &= \sup_{z^f \in Z_{h_f}^f} \frac{\langle (S_\Gamma^f)^{-1/2}w^f, z^f\rangle^2}{\|z^f\|^2} = \sup_{v^f \text{ balanced}} \frac{\langle w^f, v^f\rangle^2}{\|(S_\Gamma^f)^{1/2}v^f\|^2}\\ &= \sup_{v^f \text{ balanced}} \frac{\langle \lambda, N^f v^f\rangle^2}{\|(S_\Gamma^f)^{1/2}v^f\|^2}\\ &= \sup_{v^f \text{ balanced}} \frac{\langle (S_\Gamma^p)^{-1/2}\tilde{C}^p\lambda, (S_\Gamma^p)^{1/2}(\tilde{C}^p)^{-1}\tilde{C}^f v^f\rangle^2}{\|(S_\Gamma^f)^{1/2}v^f\|^2}.\end{aligned}$$

Then using the Cauchy–Schwarz inequality and (33) in the proof of Theorem 2, we have

$$\begin{aligned}\|(S_\Gamma^f)^{-1/2}w^f\|^2 &= \sup_{v^f \text{balanced}} \frac{\langle t, (S_\Gamma^p)^{1/2}(\tilde{C}^p)^{-1}\tilde{C}^f v^f\rangle^2}{\|(S_\Gamma^f)^{1/2}v^f\|^2}\\ &\le \|t\|^2 \sup_{v^f \text{balanced}} \frac{\|(S_\Gamma^p)^{1/2}(\tilde{C}^p)^{-1}\tilde{C}^f v^f\|^2}{\|(S_\Gamma^f)^{1/2}v^f\|^2} \prec \frac{1}{\kappa}\|t\|^2.\end{aligned}$$ □

Using Lemma 3 we can derive the following estimate for the condition number of the FETI preconditioner defined in (34).

Theorem 4. *Let λ be a zero mean Lagrange multiplier. Then*

$$\langle N^p (S^p)^{-1} N_p^T \lambda, \lambda\rangle \prec \langle F\lambda, \lambda\rangle \prec \left(1 + \frac{1}{\kappa}\right)\langle N^p (S^p)^{-1} N^{pT}\lambda, \lambda\rangle.$$

The condition number estimate $O((\kappa+1)/\kappa)$ can be improved in the case where the fluid side triangulation is finer than the porous side triangulation. This case has some advantages when κ is small. In order to fix ideas and simplify notation we analyze in detail the case where the triangulation of the fluid side is a *refinement* of the porous side triangulation. In particular, in Theorem 7, we will prove that the condition of the FETI preconditioned operator is of order $O((\kappa+1)/(\kappa+(h^p)^2))$ in this simpler situation. The analysis that we will present to prove Theorem 7 can be extended easily for the case where the fluid side triangulation is finer than (and not necessarily a refinement of) the porous side triangulation; see Remark 8.

We assume that the fluid side discretization on Γ, $\mathcal{T}^f_{h^f}(\Omega^f)|_\Gamma$, is a refinement of the corresponding porous side discretization, $\mathcal{T}^p_{h^p}(\Omega^p)|_\Gamma$. That is, assume that $h^p = rh^f$ for some positive integer r. We will refer to this assumption as the *nested refinement assumption*. For $j = 1, \dots, m^p$, we introduce the normal fluid velocity ϕ^f_j as the P2 bubble function defined on $\mathcal{T}^p_{h^p}(\Omega^p)|_\Gamma$ and with support on the interval $e^p_j = \{0\} \times [(j-1)h^p, jh^p]$. Recall that we are using P2/P1 Taylor–Hood discretization on the fluid side. Under the nested refinement assumption we have $\phi^f_j \in Z^f_{h^f}$ with $Z^f_{h^f}$ defined in (24). Denote by $Z^f_{h^f,b}$ the subspace of $Z^f_{h^f}$ spanned by all ϕ^f_j, $j = 1, \dots, m^p$, and by $Z^f_{h^f,0}$ the subspace of $Z^f_{h^f}$ spanned by functions with zero average on all edges e^p_j, $j = 1, \dots, m^p$. Note that $Z^f_{h^f,b}$ and $Z^f_{h^f,0}$ form a direct sum for $Z^f_{h^f}$ and the image $\Pi Z^f_{h^f,0}$ is the zero vector.

Before deriving the condition number estimate of the FETI preconditioner under the nested refinement assumption we first prove a preliminary lemma.

Lemma 5. *Assume that $h^p = rh^f$, where r is a positive integer. If $v^f_{\Gamma,b} \in Z^f_{h^f,b}$ is a balanced function, then*

$$\langle S^f_\Gamma v^f_{\Gamma,b}, v^f_{\Gamma,b}\rangle \prec \frac{\kappa}{(h^p)^2}\langle S^p_\Gamma \Pi v^f_{\Gamma,b}, \Pi v^f_{\Gamma,b}\rangle.$$

Proof. Let

$$v^f_{\Gamma,b} = \sum_{j=1}^{m^p} \beta_j \phi^f_j \in Z^f_{h^f,b} \subset Z^f_{h^f},$$

and note that since the basis functions ϕ^f_j, $j = 1, \dots, m^p$, do not overlap each other on Γ, they are orthogonal in $L^2(\Gamma)$ and also in $H^1_0(\Gamma)$. Then

$$\|v^f_{\Gamma,b}\|^2_{L^2(\Gamma)} = \sum_{j=1}^{m^p} \beta_j^2 \|\phi^f_j\|^2_{L^2(\Gamma)} \asymp h^p \sum_{j=1}^{m^p} \beta_j^2, \tag{35}$$

$$|v^f_{\Gamma,b}|^2_{H^1(\Gamma)} = \sum_{j=1}^{m^p} \beta_j^2 |\phi^f_j|^2_{H^1_0(e^p_j)} \asymp \frac{1}{h^p} \sum_{j=1}^{m^p} \beta_j^2. \tag{36}$$

Using (35), (36) and a interpolation estimate we see that

$$\|v^f_{\Gamma,b}\|^2_{H^{1/2}_{00}(\Gamma)} \asymp \sum_{j=1}^{m^p} \beta_j^2 \asymp \frac{1}{h^p}\|v^f_{\Gamma,b}\|^2_{L^2(\Gamma)}.$$

Note also that $\langle S^f v^f_{\Gamma,b}, v^f_{\Gamma,b}\rangle \leq a^f(\mathcal{SH}v^f_{\Gamma,b}, \mathcal{SH}v^f_{\Gamma,b}) \asymp \nu \|v^f_{\Gamma,b}\|^2_{H^{1/2}_{00}(\Gamma)}$.

Denote by

$$z^p_{\Gamma,b} = \sum_{j=1}^{m^p} \rho_j \chi_{e^p_j}$$

the unique piecewise constant function such that $\Pi v^f_{\Gamma,b} = z^p_{\Gamma,b}$. Note that $|\rho_j| \asymp |\beta_j|$, $j = 1, \ldots, m^p$. We obtain

$$\langle S^f_\Gamma v^f_{\Gamma,b}, v^f_{\Gamma,b}\rangle \prec \frac{\nu}{h^p}\|v^f_{\Gamma,b}\|^2_{L^2(\Gamma)} \asymp \frac{\nu}{h^p}\|z^p_{\Gamma,b}\|^2_{L^2(\Gamma)} \tag{37}$$

$$\prec \frac{\nu}{(h^p)^2}\|z^p_{\Gamma,b}\|^2_{(H^{1/2})'(\Gamma)} \asymp \frac{\kappa}{(h^p)^2}\langle S^p_\Gamma z^p_{\Gamma,b}, z^p_{\Gamma,b}\rangle, \tag{38}$$

where we have used an inverse inequality for piecewise constant functions. □

We now translate Lemma 5 in a result concerning the dual preconditioner.

Lemma 6. *Assume that $h^p = rh^f$, where r is a positive integer and let λ be a zero mean Lagrange multiplier. Then*

$$\frac{(h^p)^2}{\kappa}\langle N^p(S^p)^{-1}N^{pT}\lambda, \lambda\rangle \prec \langle N^f(S^f)^{-1}N^{fT}\lambda, \lambda\rangle.$$

Proof. We proceed as before. Let $t = (S^f_\Gamma)^{-\frac{1}{2}}\tilde{C}^{fT}\lambda$ and $w = \tilde{C}^p\lambda$. Then

$$\begin{aligned}
\|(S^p_\Gamma)^{-\frac{1}{2}}w\|^2 &= \sup_{z^p\in Z^p_{h^p}} \frac{\langle (S^p_\Gamma)^{-\frac{1}{2}}w, z^p\rangle^2}{\|z^p\|^2} = \sup_{v^p\text{balanced}} \frac{\langle w, v^p\rangle^2}{\|(S^p_\Gamma)^{\frac{1}{2}}v^p\|^2} \\
&= \sup_{v^p\text{balanced}} \frac{\langle \lambda, N^p v^p\rangle^2}{\|(S^p_\Gamma)^{\frac{1}{2}}v^p\|^2} = \sup_{v^f_b\text{balanced}} \frac{\langle \lambda, \tilde{C}^f v^f_b\rangle^2}{\|(S^p_\Gamma)^{\frac{1}{2}}(\tilde{C}^p)^{-1}N^f v^f_b\|^2} \\
&= \sup_{v^f_b\text{balanced}} \frac{\langle (S^f_\Gamma)^{-\frac{1}{2}}\tilde{C}^{fT}\lambda, (S^f_\Gamma)^{\frac{1}{2}}v^f_b\rangle^2}{\|(S^p_\Gamma)^{\frac{1}{2}}(\tilde{C}^p)^{-1}\tilde{C}^f v^f_b\|^2} \\
&\leq \|t\|^2 \sup_{v^f_b\text{balanced}} \frac{\|(S^f_\Gamma)^{\frac{1}{2}}v^f_b\|^2}{\|(S^p_\Gamma)^{\frac{1}{2}}(\tilde{C}^p)^{-1}\tilde{C}^f v^f_b\|^2} \prec \frac{\kappa}{(h^p)^2}\|t\|^2,
\end{aligned}$$

where the last step follows from Lemma 5. □

From Lemmas 3 and 6, the next theorem follows.

Theorem 7. *Assume that $h^p = rh^f$, where r is a positive integer. Let λ be a zero mean Lagrange multiplier*, *then*

$$\Big(1+\frac{(h^p)^2}{\kappa}\Big)\langle N^p(S^p)^{-1}N^{pT}\lambda,\lambda\rangle \prec \langle F\lambda,\lambda\rangle \prec \Big(1+\frac{1}{\kappa}\Big)\langle N^p(S^p)^{-1}N^{pT}\lambda,\lambda\rangle.$$

We solve the system (27) using preconditioned conjugate gradient. Here is the algorithm:

```
(1) Initialize:
        $x^{(0)} = 0$  (no coarse problem)
        $\lambda^{(0)} = c$
(2) Iterate $k = 1, 2, \dots$ until convergence:
        Precondition: $y^{(k-1)} = (N^p)^\dagger (S^p)(N^{pT})^\dagger d^{(k-1)}$,
        $\beta^k = \langle y^{(k-1)}, d^{(k-1)}\rangle / \langle y^{(k-2)}, d^{(k-1)}\rangle \ [\beta^{(1)} = 0]$ ,
        $r^{(k)} = y^{(k-1)} + \beta^{(k)} r^{(k)} \ [r^{(1)} = y^{(0)}]$ ,
        $\alpha^{(k)} = \langle y^{(k-1)}, d^{(k-1)}\rangle / \langle d^{(k)}, Fr^{(k)}\rangle$,
        $x^{(k)} = x^{(k-1)} + \alpha^{(k)} r^{(k)}$,
        $d^{(k)} = d^{(k-1)} - \alpha^{(k)} F r^{(k)}$.
```

Implementation of the preconditioned conjugate gradient algorithm for the system (27) *involving the FETI preconditioner* (34).

Remark 8. Theorem 7 can be extended for the case where $h^f \le 2h^p$. We only need to extend the argument given in the proof of Lemma 5. The basic idea in the proof of Lemma 5 is to associate a bubble function $\phi_j^f \in Z_{h^f}^f$ to each porous side element e_j^p, $j = 1, \dots, m^p$, in such a way that we can construct a one to one and continuous map $v_{\Gamma,b}^f \mapsto z_{\Gamma,b}^p$. The bubble functions ϕ_j^f, $j = 1, \dots, m^p$, can be chosen orthogonal in $L^2(\Gamma)$ and in $H_0^1(\Gamma)$. This can also be done when $h^f \le h^p$. The smaller the h^f, the closer is the size of the support of the bubble ϕ_j^f to the size of the element e_j^p since more and more elements e^f can be associated to only one element e^p. This construction can also be carried out in the case $h^p < h^f \le 2h^p$ where nonorthogonal Taylor–Hood basis functions must be used. This last situation leads to the appearance of an additional constant that depends on the nonorthogonality; see Section 8.

Remark 9. We note that Lemma 5 can be used directly to obtain a bound for the balancing domain decomposition preconditioner similar to the one presented in Section 6 but with $\tilde{S}^p$ instead of $\tilde{S}^f$ in (30); see Proposition 2 of [25]. In this case an additional variable elimination is needed. We have to eliminate the component of the normal fluid velocity in the space $Z^f_{h^f,0}$ and work with the Schur complement with respect to the space $Z^f_{h^f,b}$. This is rather difficult to implement (we can use Lagrange multipliers in this case). Then passing to the dual preconditioner permits us to take advantage of the case where the fluid side discretization on Γ is a refinement of the corresponding porous side discretization.

Remark 10. Theorems 2, 4 and 7 are also valid for the case $\alpha^f > 0$ in (3). To see this we need to compare, for different values of α^f, the energy of discrete extensions for a given normal velocity defined on Γ. Given the outward normal velocity v^f_Γ on Γ, let $\mathcal{S}\mathcal{H}_{\alpha^f} v^f_\Gamma$ denote the discrete harmonic extension in the sense of $(a^f_{\alpha^f}, b^f)$, that is, the solution of problem (22) with a^f replaced by $a^f_{\alpha^f}$. Recall that $a^f = a^f_0$, where $a^f_0 = a_{\alpha^f}$ when $\alpha^f = 0$, and therefore, $\mathcal{S}\mathcal{H} v^f_\Gamma = \mathcal{S}\mathcal{H}_0 v^f_\Gamma$. Note that in (22) we have imposed the *natural* boundary condition $\tau^T \boldsymbol{D}(\mathcal{S}\mathcal{H} u^f)\eta_f = 0$ on Γ. Now we define another extension denoted by $\widehat{\mathcal{S}\mathcal{H}} v^f_\Gamma$. Given the outward normal velocity v^f_Γ on Γ, let $\widehat{\mathcal{S}\mathcal{H}} v^f_\Gamma$ be the (a^f, b^f)-discrete harmonic extension given by the solution of (22) with the boundary condition $\widehat{\mathcal{S}\mathcal{H}} v^f_\Gamma \cdot \tau = 0$. For both $\mathcal{S}\mathcal{H}$ and $\widehat{\mathcal{S}\mathcal{H}}$ are imposed essential boundary condition v^f_Γ for the normal component on Γ. The difference between them is in how the boundary condition is imposed for the tangential component on Γ: For the $\mathcal{S}\mathcal{H}$, is imposed homogeneous *natural* boundary condition, while for $\widehat{\mathcal{S}\mathcal{H}}$, is imposed homogeneous *essential* boundary condition.

Both extensions $\mathcal{S}\mathcal{H}_{\alpha^f}$ and $\widehat{\mathcal{S}\mathcal{H}}$ satisfy the zero discrete divergence and boundary conditions in (22). Using this fact and the minimization property of the $(a^f_{\alpha^f}, b^f)$-discrete harmonic extension $\mathcal{S}\mathcal{H}_{\alpha^f}$ and the (a^f, b^f)-discrete harmonic extension $\widehat{\mathcal{S}\mathcal{H}}$, we get

$$
\begin{aligned}
&a^f(\mathcal{S}\mathcal{H} v^f_\Gamma, \mathcal{S}\mathcal{H} v^f_\Gamma)\\
&\qquad = a^f_0(\mathcal{S}\mathcal{H} v^f_\Gamma, \mathcal{S}\mathcal{H} v^f_\Gamma) \ \text{(by definition)}\\
&\qquad \le a^f_0(\mathcal{S}\mathcal{H}_{\alpha^f} v^f_\Gamma, \mathcal{S}\mathcal{H}_{\alpha^f} v^f_\Gamma) \ \text{(by the minimization property of } \mathcal{S}\mathcal{H}\text{)}\\
&\qquad \le a^f_{\alpha^f}(\mathcal{S}\mathcal{H}_{\alpha^f} v^f_\Gamma, \mathcal{S}\mathcal{H}_{\alpha^f} v^f_\Gamma) \ (\alpha^f > 0)\\
&\qquad \le a^f_{\alpha^f}(\widehat{\mathcal{S}\mathcal{H}} v^f_\Gamma, \widehat{\mathcal{S}\mathcal{H}} v^f_\Gamma) \ \text{(by the minimization property of } \mathcal{S}\mathcal{H}_{\alpha^f}\text{)}\\
&\qquad = a^f_0(\widehat{\mathcal{S}\mathcal{H}_0} v^f_\Gamma, \widehat{\mathcal{S}\mathcal{H}_0} v^f_\Gamma) \ \text{(because } \widehat{\mathcal{S}\mathcal{H}} u^f \cdot \tau^f = 0 \text{ on } \Gamma\text{)}\\
&\qquad \asymp \nu \|v^f_\Gamma\|^2_{H^{1/2}_{00}(\Gamma)}\\
&\qquad \asymp a^f(\mathcal{S}\mathcal{H}_0 v^f_\Gamma, \mathcal{S}\mathcal{H} v^f_\Gamma).
\end{aligned}
$$

The last two equivalences follow from properties of the (a^f, b)-discrete harmonic extensions $\mathcal{S}\mathcal{H}$ and $\widehat{\mathcal{S}\mathcal{H}}$ (which coincides with the discrete Stokes harmonic extension) [28; 40]. The two equivalences appearing above are independent of the permeability, fluid viscosity and mesh sizes. Then, the energy of the $(a^f_{\alpha^f}, b)$-discrete harmonic extensions is equivalent to the energy of the (a^f, b)-discrete harmonic extension, that is, the discrete Stokes harmonic extension. This equivalence guarantees the extensions of Theorems 2, 4 and 7 to the case $\alpha^f > 0$.

8. Numerical results

In this section we present numerical tests in order to verify the estimates in Theorems 2, 4 and 7. We consider $\Omega^f = (1, 2) \times (0, 1)$ and $\Omega^p = (0, 1) \times (0, 1)$. See [11] and [26] for examples of exact solutions and compatible divergence and boundary data. Note that the reduced systems (15) and (27) involve only degrees of freedom on the interface Γ. To solve both reduced systems (15) and (27) we can use the PCG algorithms described on pages 15 and 19. Recall that the original system (11) is a "three times" saddle point problem. Note that since the finite element basis of $M^f_{h^f} \times M^p_{h^p}$ and Λ^{h^p} have no zero mean, the finite element matrix in (13) has the kernel composed by constant pressures in $\Omega = \mathrm{int}(\overline{\Omega^f} \cup \overline{\Omega^p})$ and constant Lagrange multipliers on Γ. The corresponding system is solved up to a constant pressure and a constant Lagrange multiplier. These constants can be recovered when imposing the zero average pressure constraint [26].

In our test problems we compute the eigenvalues of the preconditioned operators. We also run PCG until the initial residual is reduced by a factor of 10^{-6}.

8.1. ***BDD preconditioner.*** In the case of the BDD preconditioner (30) for (15), we solve a coarse problem before reducing the system to ensure balanced velocities at the beginning of the CG iterations.

We consider $\alpha^f = 0$ and $\nu = 1$, and different values of h^f and h^p with nonmatching grids across the interface Γ. Table 1 shows results for $\kappa = 1$, Table 2 for $\kappa = 10^{-3}$ and Table 3 for $\kappa = 10^{-5}$. These three tables reveal growth of order $O(1 + (1/\kappa))$ in κ and hence, verify the sharpness of the estimate in Theorem 2.

$h^f \downarrow \quad h^p \rightarrow$	$3^{-1} * 2^{-0}$	$3^{-1} * 2^{-1}$	$3^{-1} * 2^{-2}$	$3^{-1} * 2^{-3}$	$3^{-1} * 2^{-4}$
$2^{-1} * 2^{-0}$	1, 1.0189(3)	1, 1.0198(3)	1, 1.0194(3)	1, 1.0193(3)	1, 1.0193(3)
$2^{-1} * 2^{-1}$	1, 1.0209(3)	1, 1.0200(3)	1, 1.0197(3)	1, 1.0196(3)	1, 1.0196(3)
$2^{-1} * 2^{-2}$	1, 1.0217(3)	1, 1.0205(3)	1, 1.0202(3)	1, 1.0201(3)	1, 1.0201(3)
$2^{-1} * 2^{-3}$	1, 1.0220(3)	1, 1.0208(3)	1, 1.0204(3)	1, 1.0203(3)	1, 1.0203(3)
$2^{-1} * 2^{-4}$	1, 1.0221(3)	1, 1.0209(3)	1, 1.0205(3)	1, 1.0204(3)	1, 1.0204(3)

Table 1. Minimum and maximum eigenvalues (and number of PCG iterations) for the BDD preconditioned operator. Here $\kappa = 1$ and $\alpha^f = 0$.

$h^f\downarrow$ $h^p\rightarrow$	$3^{-1}*2^{-1}$	$3^{-1}*2^{-2}$	$3^{-1}*2^{-3}$	$3^{-1}*2^{-4}$
$2^{-1}*2^{-0}$	1, 21.0147(3)	1, 20.6035(3)	1, 20.3686(3)	1, 20.2893(3)
$2^{-1}*2^{-1}$	1, 21.3303(6)	1, 20.8549(7)	1, 20.6550(7)	1, 20.5836(7)
$2^{-1}*2^{-2}$	1, 22.0017(6)	1, 21.3392(9)	1, 21.1424(10)	1, 21.0735(10)
$2^{-1}*2^{-3}$	1, 22.2367(6)	1, 21.6045(10)	1, 21.3626(9)	1, 21.2955(10)
$2^{-1}*2^{-4}$	1, 22.3479(6)	1, 21.7006(10)	1, 21.4666(11)	1, 21.3929(9)

Table 2. Minimum and maximum eigenvalues (and number of PCG iterations) for the BDD preconditioned operator. Here $\kappa = 10^{-3}$ and $\alpha^f = 0$.

$h^f\downarrow$ $h^p\rightarrow$	$3^{-1}*2^{-1}$	$3^{-1}*2^{-2}$	$3^{-1}*2^{-3}$	$3^{-1}*2^{-4}$
$2^{-1}*2^{-0}$	1, 1977.08(3)	1, 1945.05(3)	1, 1932.10(3)	1, 1928.32(3)
$2^{-1}*2^{-1}$	1, 1997.27(6)	1, 1972.77(7)	1, 1961.34(7)	1, 1957.88(7)
$2^{-1}*2^{-2}$	1, 2053.57(6)	1, 2021.03(13)	1, 2010.27(17)	1, 2006.90(17)
$2^{-1}*2^{-3}$	1, 2079.68(6)	1, 2044.05(13)	1, 2032.42(21)	1, 2029.13(31)
$2^{-1}*2^{-4}$	1, 2090.10(6)	1, 2054.33(13)	1, 2042.26(22)	1, 2038.90(28)

Table 3. Minimum and maximum eigenvalues (and number of PCG iterations) for the BDD preconditioned operator. Here $\kappa = 10^{-5}$ and $\alpha^f = 0$.

8.2. *FETI preconditioner.* In the case of the FETI preconditioner (34), we solve the reduced system (27) up to a constant Lagrange multiplier and a constant pressure. These constants are recovered after enforcing zero mean pressure on $\Omega = \text{int}\,(\overline{\Omega}^f \cup \overline{\Omega}^p)$ [26]. We recall that the FETI method can be viewed as the dual preconditioner counterpart of the BDD preconditioner. We repeat the same experiments mentioned above for the latter preconditioner.

$h^f\downarrow$ $h^p\rightarrow$	$3^{-1}*2^{-1}$	$3^{-1}*2^{-2}$	$3^{-1}*2^{-3}$	$3^{-1}*2^{-4}$
$2^{-1}*2^{-0}$	1.0000, 1.0208(3)	1.0000, 1.0194(3)	1.0000, 1.0193(3)	1.0000, 1.0193(3)
$2^{-1}*2^{-1}$	1.0017, 1.0200(3)	1.0000, 1.0197(3)	1.0000, 1.0196(3)	1.0000, 1.0196(3)
$2^{-1}*2^{-2}$	1.0026, 1.0205(3)	1.0004, 1.0202(3)	1.0000, 1.0200(3)	1.0000, 1.0201(3)
$2^{-1}*2^{-3}$	1.0027, 1.0208(3)	1.0007, 1.0204(3)	1.0001, 1.0203(3)	1.0000, 1.0203(3)
$2^{-1}*2^{-4}$	1.0028, 1.0209(2)	1.0007, 1.0205(3)	1.0002, 1.0204(3)	1.0000, 1.0204(3)
$2^{-1}*2^{-5}$	1.0028, 1.0209(2)	1.0007, 1.0206(3)	1.0002, 1.0205(3)	1.0000, 1.0204(3)

Table 4. Minimum and maximum eigenvalues (and number of PCG iterations) of the FETI preconditioned operator. Here $\kappa = 1$ and $\alpha^f = 0$.

$h_f \downarrow$ $h_p \rightarrow$	$3^{-1}*2^{-1}$	$3^{-1}*2^{-2}$	$3^{-1}*2^{-3}$	$3^{-1}*2^{-4}$
$2^{-1}*2^{-0}$	1.000, 20.7608(3)	1.000, 20.4405(3)	1.000, 20.3110(3)	1.000, 20.2732(3)
$2^{-1}*2^{-1}$	2.707, 20.9627(5)	1.000, 20.7177(7)	1.000, 20.6034(7)	1.000, 20.5688(7)
$2^{-1}*2^{-2}$	3.634, 21.5257(5)	1.425, 21.2003(10)	1.000, 21.0927(12)	1.000, 21.0590(12)
$2^{-1}*2^{-3}$	3.714, 21.7868(5)	1.651, 21.4305(9)	1.106, 21.3142(11)	1.000, 21.2813(12)
$2^{-1}*2^{-4}$	3.760, 21.891 (5)	1.663, 21.5333(9)	1.162, 21.4126(11)	1.026, 21.3790(12)
$2^{-1}*2^{-5}$	3.771, 21.937 (5)	1.673, 21.5768(9)	1.164, 21.4561(11)	1.040, 21.4220(12)

Table 5. Minimum and maximum eigenvalues (and number of PCG iterations) for the FETI preconditioned operator. Here $\kappa = 10^{-3}$ and $\alpha^f = 0$.

$h^f \downarrow$ $h^p \rightarrow$	$3^{-1}*2^{-2}$	$3^{-1}*2^{-3}$	$3^{-1}*2^{-4}$
$2^{-1}*2^{-0}$	1.00, 1945.05(3)	1.00, 1932.10(3)	1.00, 1928.32(3)
$2^{-1}*2^{-1}$	1.00, 1972.77(7)	1.00, 1961.34(7)	1.00, 1957.88(7)
$2^{-1}*2^{-2}$	43.45, 2021.03(11)	1.00, 2010.27(17)	1.00, 2006.90(17)
$2^{-1}*2^{-3}$	66.10, 2044.05(11)	11.58, 2032.42(20)	1.00, 2029.13(37)
$2^{-1}*2^{-4}$	67.29, 2054.33(10)	17.20, 2042.26(19)	3.64, 2038.90(35)
$2^{-1}*2^{-5}$	68.32, 2058.68(10)	17.42, 2046.61(10)	5.04, 2043.20(36)

Table 6. Minimum and maximum eigenvalues (and number of PCG iterations) for the FETI preconditioned operator. Here $\kappa = 10^{-5}$ and $\alpha^f = 0$.

We consider $\alpha^f = 0$, $\nu = 1$ and different values of h^f and h^p with nonmatching grids across the interface Γ; see Table 4 on the previous page for the results when $\kappa = 1$, Table 5 for $\kappa = 10^{-3}$ and Table 6 for the case $\kappa = 10^{-5}$. Note that in Tables 4–6 the minimum eigenvalues are strictly greater than one when $h^f \leq 2h^p$, and the value of the minimum eigenvalues seem to stabilize very quickly for smaller h^f with fixed h^p. This confirms the extension of Theorem 7 for the case where $h^f \leq 2h^p$ (Remark 8). In Table 7 we present the numerical results where one of the meshes on the interface is a refinement of the other side triangulation on the interface. We observe a behavior similar to the behavior of Table 6 with a bigger value for the minimum eigenvalue when $h_f \leq h_p$. This verifies the estimates of Theorem 7. This shows that the FETI preconditioner is scalable for the parameters faced in practice, that is, when the fluid side mesh is finer than the porous side mesh, and the permeability κ is very small. We conclude that the numerical experiments concerning the FETI preconditioner reveal the sharpness of the results obtained in Theorems 4 and 7 and Remark 8.

Recall that we have assumed $\alpha^f = 0$. Now consider $\alpha^f > 0$. Numerical experiment were performed with $\alpha^f > 0$ revealing results similar to the ones presented

$h^f \downarrow$ $h^p \rightarrow$	$2^{-1} * 2^{-2}$	$2^{-1} * 2^{-3}$	$2^{-1} * 2^{-4}$
$2^{-1} * 2^{-0}$	1.00, 1961.35(3)	1.00, 1937.86(3)	1.00, 1929.93(3)
$2^{-1} * 2^{-1}$	1.00, 1986.49(7)	1.00, 1966.50(7)	1.00, 1959.36(7)
$2^{-1} * 2^{-2}$	176.56, 2034.92(7)	1.00, 2015.24(18)	1.00, 2008.35(17)
$2^{-1} * 2^{-3}$	151.62, 2061.45(7)	44.91, 2037.26(13)	1.00, 2030.55(45)
$2^{-1} * 2^{-4}$	154.45, 2071.06(7)	38.04, 2047.66(13)	11.98, 2040.29(21)
$2^{-1} * 2^{-5}$	154.86, 2075.43(7)	38.73, 2051.91(13)	10.20, 2044.66(24)

Table 7. Minimum and maximum eigenvalues (and number of PCG iterations) for the FETI preconditioned operator. Here $\kappa = 10^{-5}$ and $\alpha^f = 0$. The refinement condition of Theorem 7 is satisfied under the diagonal.

$h^f \downarrow$ $h^p \rightarrow$	$3^{-1}\ 2^{-2}$	$3^{-1}\ 2^{-3}$	$3^{-1}\ 2^{-4}$
$2^{-1}\ 2^{-0}$	1.00, 1678.07(3)	1.00, 1666.84(3)	1.00, 1663.55(3)
$2^{-1}\ 2^{-1}$	1.00, 1787.53(7)	1.00, 1776.50(7)	1.00, 1773.22(7)
$2^{-1}\ 2^{-2}$	41.65, 1812.69(17)	1.00, 1801.61(17)	1.00, 1798.29(17)
$2^{-1}\ 2^{-3}$	63.63, 1816.43(17)	11.24, 1804.66(13)	1.00, 1801.34(43)
$2^{-1}\ 2^{-4}$	66.82, 1817.38(17)	16.75, 1805.30(13)	3.58, 1801.91(23)
$2^{-1}\ 2^{-5}$	67.99, 1817.68(17)	17.37, 1805.57(13)	4.97, 1802.14(24)

Table 8. Minimum and maximum eigenvalues (and number of PCG iterations) for the FETI preconditioned operator. Here $\kappa = 10^{-5}$ and $\alpha^f = 1$.

above for the case $\alpha^f = 0$. We only include Table 8 which shows the extreme eigenvalues of the FETI preconditioned operator for the case $\alpha^f = 1$, $\nu = 1$ and $\kappa = 10^{-5}$. This table presents a similar behavior to the one with $\alpha^f = 0$ in Table 6 and hence confirms Remark 10, which says that the parameter α^f does not play much of a role for preconditioning.

9. The multisubdomain case

The methods introduced in the previous sections considered only the two-subdomain cases where discrete Stokes and Darcy indefinite subproblems are solved exactly in each subdomain and in each CG iteration. These methods might be very costly for large subproblems since direct or accurate iterative local solvers for the indefinite systems have to be used. In this section we show that the methodology developed for the two-subdomain cases can be developed also for the multisubdomain case. The analysis (using tools developed in Section 7) and numerical experiments for the multisubdomain case will be presented elsewhere.

We now extend the FETI method of Section 7 for many subdomains when the triangulations $\mathcal{T}^f_{h^f}$ and $\mathcal{T}^p_{h^p}$ coincide on the interface Γ. Let $\{\Omega^i_j\}_{j=1}^{n^i}$ be a geometrically conforming substructures of Ω^i, $i = f, p$. We also assume that $\{\Omega^f_j\}_{j=1}^{n^f} \cup \{\Omega^p_j\}_{j=1}^{n^p}$ form a geometrically conforming decomposition of Ω; hence, the two decompositions are aligned on the interface Γ. We define the local inner interfaces as $\Gamma^i_j = \partial\Omega^i_j \setminus \partial\Omega^i$, $j = 1, \ldots, n^i$, $i = f, p$. We also define

$$\widetilde{\Gamma} = \bigcup_{j=1}^{n^f} \Gamma^f_j \cup \bigcup_{j=1}^{n^p} \Gamma^p_j \cup \Gamma.$$

See Figure 1. In order to simplify the presentation, we assume that for the fluid region, the spaces $\boldsymbol{X}^f_{h^f}$ and $M^f_{h^f}$ are the P2/P0 triangular finite elements, while for the porous region, the spaces $\boldsymbol{X}^p_{h^p} \subset \boldsymbol{X}^p$ and $M^p_{h^p} \subset L^2(\Omega^p)$ are the lowest order Raviart–Thomas finite elements based on triangles. Similar as in the previous sections, and using the FETI-DP framework [42], we decompose the velocity and pressure spaces as follows:

$\boldsymbol{X}^f_I$: interior velocities in the subdomains $\{\Omega^f_j\}_{j=1}^{nf}$

$\boldsymbol{X}^f_{\widetilde{\Gamma}}$: interface velocities on $\widetilde{\Gamma} \cap \overline{\Omega}^f$

$\boldsymbol{X}^p_I$: interior velocities in the subdomains $\{\Omega^p_j\}_{j=1}^{n^p}$

$\boldsymbol{X}^p_{\widetilde{\Gamma}}$: interface velocities on $\widetilde{\Gamma} \cap \overline{\Omega}^p$

M^i_I, $(i = p, f)$: interior zero mean pressure in each subdomain $\{\Omega^i_j\}_{j=1}^{n^i}$, $i = f, p$

M^i_0, $(i = p, f)$: constant pressure in each subdomain $\{\Omega^i_j\}_{j=1}^{n^i}$, $i = f, p$

$M_I = M^f_I \times M^p_I$

$\boldsymbol{X}_I = \boldsymbol{X}^f_I \times \boldsymbol{X}^p_I$, $\boldsymbol{X}_{\widetilde{\Gamma}} = \boldsymbol{X}^f_{\widetilde{\Gamma}} \times \boldsymbol{X}^p_{\widetilde{\Gamma}}$, $M_I = M^f_I \times M^p_I$ and $M_0 = M^f_0 \times M^p_0$

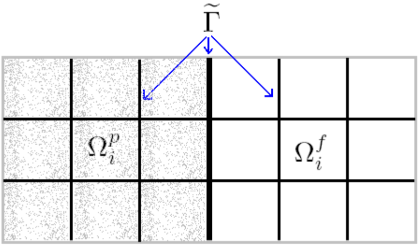


Figure 1. Global interface $\widetilde{\Gamma}$ that includes all local interfaces and the Stokes–Darcy interface Γ.

After imposing the mortar condition as in Section 4 we can reduce (11) to a Schur complement system on the interface $\widetilde{\Gamma}$,

$$\widetilde{S}_{\widetilde{\Gamma}}\boldsymbol{u}_{\widetilde{\Gamma}} = \widetilde{b}_{\widetilde{\Gamma}} \tag{39}$$

which is the multisubdomain generalization of the reduced system (15).

The $\widetilde{\Gamma}$-interface velocity space $\boldsymbol{X}_{\widetilde{\Gamma}}$ can be decomposed in primal and dual degrees of freedom, that is, $\boldsymbol{X}_{\widetilde{\Gamma}} = \boldsymbol{X}_C \oplus \boldsymbol{X}_\Delta$ where $\boldsymbol{X}_C$ consists of functions which are continuous with respect to the primal degrees of freedom. The primal variables for the fluid velocity field satisfy the continuity of the fluid velocities at the substructure corners and the continuity of the mean normal and mean tangential component on each face of the substructures $\{\Omega_j^f\}_{j=1}^{n^f}$. For the porous side, the primal variables satisfy the continuity of the mean normal flux on the each face of the subsubstructures $\{\Omega_j^p\}_{j=1}^{n^p}$ [27; 32; 33; 34; 43]. For faces of the subdomains on Γ, only the continuity of the mean fluxes is satisfied. The space $\boldsymbol{X}_\Delta$ includes the remaining fluid side velocity degrees of freedom and the remaining porous media velocity degrees of freedom.

Functions in $\boldsymbol{X}_\Delta$ do not satisfy the continuity requirements on $\widetilde{\Gamma}$. The continuity requirement can be enforced using Lagrange multipliers $\widetilde{\lambda}$ on $\widetilde{\Gamma}$ and represented by the equation

$$B_\Delta \boldsymbol{v}_\Delta = 0.$$

We ensure that this condition coincides with the last equation of (13) that corresponds to the flux continuity across the Stokes–Darcy interface Γ. On that interface we use the same Lagrange multipliers of the dual formulation (27). Proceeding as in [32] we can obtain a reduced system of the form

$$\widetilde{F}\widetilde{\lambda} = \widetilde{b},$$

which corresponds to the multisubdomain version of (27). The preconditioner operator is of the form

$$B_\Delta \widetilde{S}_{\widetilde{\Gamma}} B_\Delta^T,$$

where $\widetilde{S}_{\widetilde{\Gamma}}$ was introduced in (39). See [27] for a more detailed discussion and numerical experiments for the FETI method in the multisubdomain case.

10. Conclusions and final comments

We consider the problem of coupling fluid flows with porous media flows with Beavers–Joseph–Saffman condition on the interface. We choose a discretization consisting of Taylor–Hood finite elements of order two on the free fluid side and the lowest order Raviart–Thomas finite element on the porous fluid side. The meshes are allowed to be nonmatching across the interface.

We design and analyze two preconditioners for the resulting symmetric linear system. We note that the original linear system is symmetric indefinite and involves three Lagrange multipliers: one for each subdomain pressure and a third one to impose the weak conservation of mass across the interface Γ; see Section 1.

One preconditioner is based on BDD methods and the other one is based on FETI methods. In the case of the BDD preconditioner, the energy is controlled by the Stokes side, while in the FETI preconditioner, the energy is controlled by the Darcy system; see Theorems 2 and 4. In both cases a bound $C_1((\kappa+1)/\kappa)$ is derived. Furthermore, under the assumption that the fluid side mesh on the interface is finer than the corresponding porous side mesh, we derive the better bound $C_2((\kappa+1)/(\kappa+(h^p)^2))$ for the FETI preconditioner; see Theorem 7 and Remark 8. This better bound also shows that the FETI preconditioner is more scalable for parameters faced in practice, for example, problems with small permeability κ and where the fluid side mesh is finer than the porous side mesh. The constants C_1 and C_2 above are independent of the fluid viscosity ν, the mesh ratio across the interface and the permeability κ.

References

[1] Y. Achdou, Y. Maday, and O. B. Widlund, *Iterative substructuring preconditioners for mortar element methods in two dimensions*, SIAM J. Numer. Anal. **36** (1999), no. 2, 551–580. MR 99m:65233 Zbl 0931.65110

[2] T. Arbogast and D. S. Brunson, *A computational method for approximating a Darcy–Stokes system governing a vuggy porous medium*, Comput. Geosci. **11** (2007), no. 3, 207–218. MR 2009b:76155 Zbl 05200264

[3] T. Arbogast and M. S. M. Gomez, *A discretization and multigrid solvers for a Darcy–Stokes system of three dimensional vuggy porous media*, Comput. Geosci. **13** (2009), no. 3, 331–348.

[4] G. S. Beavers and D. D. Joseph, *Boundary conditions at a naturally permeable wall*, J. Fluid Mech. **30** (1967), 197–207.

[5] F. Ben Belgacem and Y. Maday, *The mortar element method for three-dimensional finite elements*, RAIRO Modél. Math. Anal. Numér. **31** (1997), no. 2, 289–302. MR 98e:65107 Zbl 0868.65082

[6] C. Bernardi, Y. Maday, and A. T. Patera, *A new nonconforming approach to domain decomposition: the mortar element method*, Nonlinear partial differential equations and their applications. Collège de France Seminar, Vol. XI (H. Brezis et al., eds.), Pitman Res. Notes Math. Ser., no. 299, Longman Sci. Tech., Harlow, 1994, pp. 13–51. MR 95a:65201 Zbl 0797.65094

[7] D. Braess, *Finite elements: Theory, fast solvers, and applications in solid mechanics*, 2nd ed., Cambridge University Press, Cambridge, 2001. MR 2001k:65002 Zbl 0976.65099

[8] S. C. Brenner and L. R. Scott, *The mathematical theory of finite element methods*, Texts in Applied Mathematics, no. 15, Springer, New York, 1994. MR 95f:65001 Zbl 0804.65101

[9] S. C. Brenner and L.-Y. Sung, *BDDC and FETI-DP without matrices or vectors*, Comput. Methods Appl. Mech. Engrg. **196** (2007), no. 8, 1429–1435. MR 2007k:65208 Zbl 1173.65363

[10] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, Springer Series in Computational Mathematics, no. 15, Springer, New York, 1991. MR 92d:65187 Zbl 0788.73002

[11] E. Burman and P. Hansbo, *A unified stabilized method for Stokes' and Darcy's equations*, J. Comput. Appl. Math. **198** (2007), no. 1, 35–51. MR 2007i:65076

[12] M. Discacciati and A. Quarteroni, *Analysis of a domain decomposition method for the coupling of Stokes and Darcy equations*, Numerical mathematics and advanced applications (F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, eds.), Springer Italia, Milan, 2003, pp. 3–20. MR 2008i:65288 Zbl 02064881

[13] M. Discacciati, *Domain decomposition methods for the coupling of surface and groundwater flows*, Ph.D. thesis, Ecole Polytechnique Fédérale, Lausanne (Switzerland), 2004.

[14] M. Discacciati, *Iterative methods for Stokes/Darcy coupling*, Domain decomposition methods in science and engineering (R. Kornhuber et al., eds.), Lect. Notes Comput. Sci. Eng., no. 40, Springer, Berlin, 2005, pp. 563–570. MR 2236665 Zbl 02143589

[15] M. Discacciati, E. Miglio, and A. Quarteroni, *Mathematical and numerical models for coupling surface and groundwater flows*, Appl. Numer. Math. **43** (2002), no. 1-2, 57–74. MR 2003h:76087 Zbl 1023.76048

[16] M. Discacciati and A. Quarteroni, *Convergence analysis of a subdomain iterative method for the finite element approximation of the coupling of Stokes and Darcy equations*, Comput. Vis. Sci. **6** (2004), no. 2-3, 93–103. MR 2005e:65142 Zbl 02132413

[17] M. Discacciati, A. Quarteroni, and A. Valli, *Robin–Robin domain decomposition methods for the Stokes–Darcy coupling*, SIAM J. Numer. Anal. **45** (2007), no. 3, 1246–1268. MR 20083:65390 Zbl 1139.76030

[18] C. R. Dohrmann, *A preconditioner for substructuring based on constrained energy minimization*, SIAM J. Sci. Comput. **25** (2003), no. 1, 246–258. MR 2004k:74099 Zbl 1038.65039

[19] M. Dryja and W. Proskurowski, *On preconditioners for mortar discretization of elliptic problems*, Numer. Linear Algebra Appl. **10** (2003), no. 1-2, 65–82. MR 2004b:65201 Zbl 1071.65530

[20] M. Dryja, J. Galvis, and M. Sarkis, *BDDC methods for discontinuous Galerkin discretization of elliptic problems*, J. Complexity **23** (2007), no. 4-6, 715–739. MR 2008m:65316 Zbl 1133.65097

[21] M. Dryja, H. H. Kim, and O. B. Widlund, *A BDDC algorithm for problems with mortar discretization*, Tech. Report TR2005-873, Courant Institute of Mathematical Sciences, Computer Science Department, 2005.

[22] V. J. Ervin, E. W. Jenkins, and S. Sun, *Coupled generalized nonlinear Stokes flow with flow through a porous medium*, SIAM J. Numer. Anal. **47** (2009), no. 2, 929–952. MR 2485439

[23] C. Farhat, M. Lesoinne, and K. Pierson, *A scalable dual-primal domain decomposition method*, Numer. Linear Algebra Appl. **7** (2000), no. 7-8, 687–714. MR 2001k:65085 Zbl 1051.65119

[24] C. Farhat and F.-X. Roux, *A method of finite element tearing and interconnecting and its parallel solution algorithm*, Internat. J. Numer. Methods Engrg. **32** (1991), 1205–1227. Zbl 0758.65075

[25] J. Galvis and M. Sarkis, *Balancing domain decomposition methods for mortar coupling Stokes–Darcy systems*, Domain decomposition methods in science and engineering XVI (O. Widlund and D. Keyes, eds.), Lect. Notes Comput. Sci. Eng., no. 55, Springer, Berlin, 2007, pp. 373–380. MR 2334125

[26] ______, *Non-matching mortar discretization analysis for the coupling Stokes–Darcy equations*, Electron. Trans. Numer. Anal. **26** (2007), 350–384. MR 2009a:76120 Zbl 1170.76024

[27] J. Galvis and M. Sarkis, *FETI-DP for Stokes–Mortar–Darcy systems*, 2009, Submitted to the proceedings of the 19th International Conference on Domain Decomposition Methods.

[28] V. Girault and P.-A. Raviart, *Finite element methods for Navier–Stokes equations*, Springer Series in Computational Mathematics: theory and algorithms, no. 5, Springer, Berlin, 1986. MR 88b:65129 Zbl 0585.65077

[29] W. Jäger and A. Mikelić, *On the interface boundary condition of Beavers, Joseph, and Saffman*, SIAM J. Appl. Math. **60** (2000), no. 4, 1111–1127. MR 2001e:76122

[30] A. Klawonn and O. B. Widlund, *FETI and Neumann–Neumann iterative substructuring methods: connections and new results*, Comm. Pure Appl. Math. **54** (2001), no. 1, 57–90. MR 2001i:65131 Zbl 1023.65120

[31] W. J. Layton, F. Schieweck, and I. Yotov, *Coupling fluid flow with porous media flow*, SIAM J. Numer. Anal. **40** (2002), no. 6, 2195–2218. MR 2004c:76048 Zbl 1037.76014

[32] J. Li, *A dual-primal FETI method for incompressible Stokes equations*, Numer. Math. **102** (2005), no. 2, 257–275. MR 2007e:65123 Zbl 02245459

[33] J. Li and O. Widlund, *BDDC algorithms for incompressible Stokes equations*, SIAM J. Numer. Anal. **44** (2006), no. 6, 2432–2455. MR 2008f:65218 Zbl 05223840

[34] ———, *A BDDC preconditioner for saddle point problems*, Domain decomposition methods in science and engineering XVI (O. Widlund and D. Keyes, eds.), Lect. Notes Comput. Sci. Eng., no. 55, Springer, Berlin, 2007, pp. 413–420. MR 2334130

[35] J. Mandel, *Balancing domain decomposition*, Comm. Numer. Methods Engrg. **9** (1993), no. 3, 233–241. MR 94b:65158 Zbl 0796.65126

[36] J. Mandel and M. Brezina, *Balancing domain decomposition for problems with large jumps in coefficients*, Math. Comp. **65** (1996), no. 216, 1387–1401. MR 97a:65109 Zbl 0853.65129

[37] J. Mandel and R. Tezaur, *Convergence of a substructuring method with Lagrange multipliers*, Numer. Math. **73** (1996), no. 4, 473–487. MR 97h:65142 Zbl 0880.65087

[38] T. P. Mathew, *Domain decomposition and iterative refinement methods for mixed finite element discretizations of elliptic problems*, Ph.D. thesis, Courant Institute of Mathematical Sciences, 1989.

[39] M. Mu and J. Xu, *A two-grid method of a mixed Stokes–Darcy model for coupling fluid flow with porous media flow*, SIAM J. Numer. Anal. **45** (2007), no. 5, 1801–1813. MR 2008i:65264 Zbl 1146.76031

[40] L. F. Pavarino and O. B. Widlund, *Balancing Neumann–Neumann methods for incompressible Stokes equations*, Comm. Pure Appl. Math. **55** (2002), no. 3, 302–335. MR 2002h:76048 Zbl 1024.76025

[41] B. Rivière and I. Yotov, *Locally conservative coupling of Stokes and Darcy flows*, SIAM J. Numer. Anal. **42** (2005), no. 5, 1959–1977. MR 2006a:76035 Zbl 1084.35063

[42] A. Toselli and O. Widlund, *Domain decomposition methods—algorithms and theory*, Springer Series in Computational Mathematics, no. 34, Springer, Berlin, 2005. MR 2005g:65006 Zbl 1069.65138

[43] X. Tu, *A BDDC algorithm for a mixed formulation of flow in porous media*, Electron. Trans. Numer. Anal. **20** (2005), 164–179. MR 2006g:76078 Zbl 1160.76368

[44] ———, *A BDDC algorithm for flow in porous media with a hybrid finite element discretization*, Electron. Trans. Numer. Anal. **26** (2007), 146–160. MR 2008k:76086 Zbl 1170.76034

[45] B. I. Wohlmuth, *A mortar finite element method using dual spaces for the Lagrange multiplier*, SIAM J. Numer. Anal. **38** (2000), no. 3, 989–1012. MR 2001h:65132 Zbl 0974.65105

Received November 22, 2008. Revised November 17, 2009.

JUAN GALVIS: jugal@math.tamu.edu
Department of Mathematics, Texas A&M University, College Station, TX 77843-3368, United States
www.math.tamu.edu/∼jugal

MARCUS SARKIS: msarkis@wpi.edu
Worcester Polytechnic Institute, Mathematical Sciences Department, 100 Institute Road, Worcester, MA 01609, United States

and

Instituto Nacional de Matemática Pura e Aplicada, Estrada Dona Castorina 110, 22460-320 Rio de Janeiro, RJ, Brazil
www.wpi.edu/∼msarkis

COMM. APP. MATH. AND COMP. SCI.
Vol. 5, No. 1, 2010

A CUT-CELL METHOD FOR SIMULATING SPATIAL MODELS OF BIOCHEMICAL REACTION NETWORKS IN ARBITRARY GEOMETRIES

WANDA STRYCHALSKI, DAVID ADALSTEINSSON AND TIMOTHY ELSTON

Cells use signaling networks consisting of multiple interacting proteins to respond to changes in their environment. In many situations, such as chemotaxis, spatial and temporal information must be transmitted through the network. Recent computational studies have emphasized the importance of cellular geometry in signal transduction, but have been limited in their ability to accurately represent complex cell morphologies. We present a finite volume method that addresses this problem. Our method uses Cartesian-cut cells in a differential algebraic formulation to handle the complex boundary dynamics encountered in biological systems. The method is second-order in space and time. Several models of signaling systems are simulated in realistic cell morphologies obtained from live cell images. We then examine the effects of geometry on signal transduction.

1. Introduction

Cells must be able to sense and respond to external environmental cues. Information about external signals, such as hormones or growth factors, is transmitted by signaling pathways to the cellular machinery required to generate the appropriate response. Defects in these pathways can lead to diseases, such as cancer, diabetes, and heart disease. Therefore, understanding how intracellular signaling pathways function is not only a fundamental problem in cell biology, but also important for developing therapeutic strategies for treating disease.

In many pathways, proper signal transduction requires that both the spatial and temporal dynamics of the system be tightly regulated [10]. For example, recent experiments have revealed spatial gradients of protein activation in migrating cells [19]. Mathematical models can be used to elucidate the control mechanisms used to regulate the spatiotemporal dynamics of signaling pathways, and recent computational studies emphasize the importance of cellular geometry in signaling networks [16; 17; 23]. For computational simplicity, many of these investigations

MSC2000: 92-08, 65M06.

Keywords: systems biology, numerical methods, reaction-diffusion equation.

assume idealized cell geometries [12; 16], whereas others approximate irregularly shaped cells using a "staircase" representation of the cell membrane [22].

Both finite element and finite volume methods have been used to simulate spatial models of biochemical reaction networks [16; 22; 23; 31]. The most common finite volume algorithm to simulate reaction networks in two and three dimensions is the virtual cell algorithm [22]. Cellular geometries are represented by staircase curves. The authors note that the approximation of fluxes across membranes leads to a decrease in the spatial accuracy of the numerical method to first-order. The temporal accuracy of algorithm in [22] is also limited to first-order. For finite element methods, which typically require a triangulation of the computational domain, grid generation can be a challenge. This becomes especially true if the boundaries of the computational domain are moving.

To overcome the issues of accurate boundary representation and grid generation, we developed a finite volume method that utilizes a Cartesian grid. Our numerical scheme is based on a cut-cell method that accurately represents the cell boundary using a piecewise-linear approximation. The method presented here extends the results on embedded boundary methods to systems of nonlinear reaction diffusion equations with arbitrary boundary conditions. Embedded boundary methods [4; 5; 9; 13; 15; 25] have been used to solve Poisson's equation [9] and the heat equation [15; 25] with homogeneous Dirichlet and Neumann boundary conditions as well as hyperbolic conservation laws [5]. Surface diffusion of one species in three dimensions was simulated with an embedded boundary discretization in [24]. We also offer an alternative formulation to embedded boundary methods for handling the temporal update. In our formulation, the boundary conditions form a system of nonlinear algebraic equations that can be solved with existing differential algebraic equation solvers. We provide a novel use of DASPK (Differential Algebraic Solver Pack) [2] as a time integrator for the finite volume method. The embedded boundary spatial discretization combined with the differential algebraic formulation allows us to achieve second-order accuracy in space and time. Our method also provides an appropriate framework for addressing moving boundary problems using level set methods [18; 26].

The remainder of the article is organized as follows. In Section 2, we describe the mathematical formulation and governing equations. In Section 3, we describe the numerical scheme, the flux based formulation, and coupling reactions terms on the interior and boundary with spatial terms to form one interconnected system. We also outline how the system is adapted for the DASPK numerical solver [2]. In Section 4, we verify the numerical method. The computed solution is compared to a known solution on a circular domain. Additionally, we perform grid refinements of the computed solution on a well resolved grid to show convergence in the absence of an exact solution. The numerical method is then demonstrated on a more physically relevant domain with an irregular domain. Finally, we simulate a biologically relevant reaction-diffusion model on a very irregular domain.

2. Mathematical formulation

Spatial models of biochemical reaction networks are typically represented using partial differential equations consisting of reaction and diffusion terms. Active transport, driven by molecular motors, also occurs within cells. This effect can be included in our numerical scheme by the use of advection terms and will be addressed in future work. For simplicity we restrict ourselves to two spatial dimensions x and y. For a given chemical species, the reaction terms encompass processes such as activation, degradation, protein modifications and the formation of molecular complexes. These reactions typically include nonlinear terms, such as those arising from Michaelis–Menten kinetics. In a system consisting of n chemical species, the concentration of the ith species c_i evolves in space and time according to the equation

$$\frac{\partial c_i}{\partial t} = -\nabla \cdot \mathbf{J} + f_i(\boldsymbol{c}), \tag{1}$$

where $\mathbf{J} = -D_i \nabla c_i$ is the flux density, D_i is the diffusion coefficient, and the function $f_i(\boldsymbol{c})$ models the reactions within the cell that affect c_i. The elements of the vector $\boldsymbol{c}$ are the concentrations of the n chemical species. Reactions also may occur on the cell membrane yielding nonlinear conditions on the boundary $\partial\Omega$:

$$-D\vec{n} \cdot \nabla c_i|_{\partial\Omega} + g(\boldsymbol{c})|_{\partial\Omega} = 0. \tag{2}$$

Equations (1) and (2) are solved subject to appropriate initial conditions $c_i(x, y, 0)$ for each species in the system.

3. Numerical methods

Our goal is to develop a simulation tool that can accurately and efficiently solve spatial models of signaling and regulatory pathways in realistic cellular geometries. We obtain the computational domain from live-cell images. The model equations are solved on a Cartesian grid by discretizing the Laplacian operator, which models molecular diffusion, with a finite volume method.

3.1. ***Computational domain.*** Figure 1 shows a gray-scale image of a mouse fibroblast [19]. Because the original image is noisy, the image was smoothed by convolving it twice with the standard five-point Gaussian smoothing filter. After smoothing, a suitable thresholding value was picked, and the front was computed by an iso-contour finder. A signed distance function is constructed with the smoothened boundary using fast marching methods [14]. The zero-level set of the signed distance

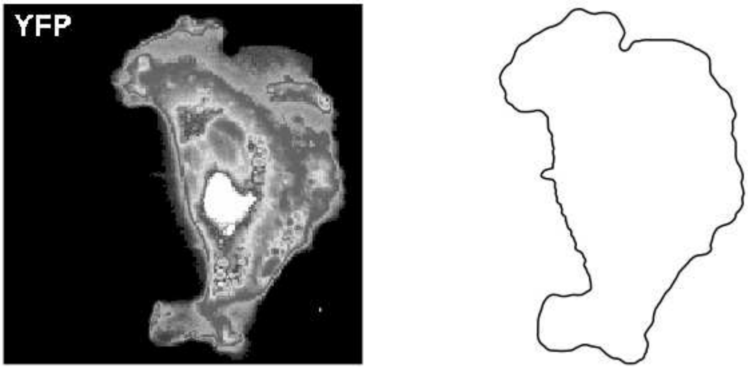


Figure 1. Grayscale image of a mouse fibroblast taken from supplemental data in [19] (left) and the smoothened boundary (right).

function yields piecewise linear segments used to define cut cells (Figure 2). Implicit representation of irregular boundaries has also been proposed in [4; 13].

3.2. ***Discretization of the spatial operator.*** We utilize a Cartesian grid-based, finite volume algorithm originally presented in [9] to discretize the diffusion operator arising from (1). Finite volume methods store the average value of the concentration over a computational grid cell at the location (i, j). That is,

$$\bar{c}_{i,j} = \frac{1}{V_{i,j}} \iint_{V_{i,j}} c(x, y)\, dV, \tag{3}$$

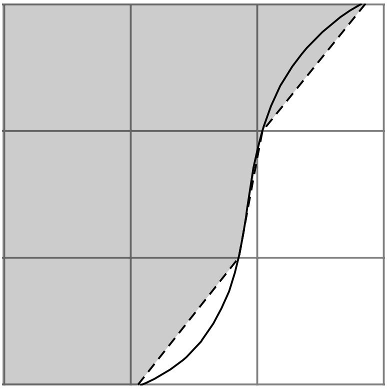


Figure 2. Computational boundary (dashed line) with an assumed higher-order representation of the cell boundary drawn as a solid line.

where $V_{i,j}$ is the volume of the (i, j) grid cell. Inserting (3) into (1) produces

$$\frac{\partial \bar{c}_{i,j}}{\partial t} - \overline{f(c)}_{i,j} = -\frac{1}{V_{i,j}} \iint_{V_{i,j}} \nabla \cdot \mathbf{J}\, dV. \tag{4}$$

The divergence theorem allows us to convert the above volume integral into a surface integral,

$$\frac{\partial \bar{c}_{i,j}}{\partial t} - \overline{f(c)}_{i,j} = -\frac{1}{V_{i,j}} \int_{\partial V_{i,j}} (\mathbf{J} \cdot \vec{n})\; dS. \tag{5}$$

For interior grid cells, we have

$$\begin{aligned}\frac{\partial \bar{c}_{i,j}}{\partial t} - \overline{f(c)}_{i,j} = -\frac{1}{V_{i,j}} \Bigg[&\int_{y_{j-1/2}}^{y_{j+1/2}} \big(J_x(x_{i+1/2}, y) - J_x(x_{i-1/2}, y)\big)\; dy \\ &+ \int_{x_{i-1/2}}^{x_{i+1/2}} \big(J_y(x, y_{j+1/2}) - J_y(x, y_{j-1/2})\big)\; dx \Bigg],\end{aligned} \tag{6}$$

where $J_x = -D(\partial c/\partial x)$ and $J_y = -D(\partial c/\partial y)$. Approximation of the integrals in (6) with the midpoint rule yields

$$\begin{aligned}\frac{\partial c_{i,j}}{\partial t} - f(c_{i,j}) \approx -\frac{1}{V_{i,j}} \big[&\Delta y (J_x(x_{i+1/2}, y_j) - J_x(x_{i-1/2}, y_j)) \\ &+ \Delta x \big(J_y(x_i, y_{j+1/2}) - J_y(x_i, y_{j-1/2})\big)\big].\end{aligned} \tag{7}$$

By approximating the gradient terms with centered differences, we arrive at the standard five-point Laplacian. Therefore in computational grid cells with volume $V_{i,j} = 1$, the finite volume stencil is the same as the five-point Laplacian approximation.

The cut-cell method generalizes as follows. The boundary of the computational domain is approximated as a piecewise linear segments (Figure 2, dashed line), and grid cells that the boundary passes through are referred to as *cut cells*. The volume of a cut cell is computed by recasting the volume integral as a boundary integral:

$$V_{i,j} = \iint_{V_{i,j}} dV = \iint_{V_{i,j}} \nabla \cdot \left(\frac{x}{2}, \frac{y}{2}\right) dV = \int_{\partial V_{i,j}} \left(\left(\frac{x}{2}, \frac{y}{2}\right) \cdot \vec{n}\right) dS, \tag{8}$$

where $\vec{n}$ is the normal vector to the surface. The integral on the right can be computed exactly for the polygon. Each segment is evaluated, then summed. The center of mass can also be computed using a boundary integral, for example:

$$\iint_{V_{ij}} x dV = \iint_{V_{ij}} \nabla \cdot \left(\frac{x^2}{2}, 0\right) dV = \int_{\partial V_{i,j}} \left(\left(\frac{x^2}{2}, 0\right) \cdot \vec{n}\right) dS. \tag{9}$$

We initialize cut cells with values computed at the centroid as in [15].

Next, we construct the integral on the right side of (5) for a cut cell. In general, there are up to five surface integrals to approximate. Let $a_{l,m} \in [0, 1]$ represent the fraction of each of the four cell edges covered by the cut cell and a_f be the length of the line segment representing the boundary. Then (7) becomes

$$\frac{\partial c_{i,j}}{\partial t} - f(c_{i,j}) \approx -\frac{1}{V_{i,j}}\Big[\Delta y\,\big(a_{i+1/2,j}J_x(x_{c_{i+1/2}}, y_j) - a_{i-1/2,j}J_x(x_{c_{i-1/2}}, y_j)\big) \\ + \Delta x\,\big(a_{i,j+1/2}J_y(x_i, y_{c_{j+1/2}}) - a_{i,j-1/2}J_y(x_i, y_{c_{j-1/2}})\big) + a^f J_f\Big]. \quad (10)$$

The notation $(x_{c_{i\pm1/2}}, y_j)$ indicates the midpoint of partially covered $(x_{i\pm1/2}, y_j)$ face. Let

$$F_{i\pm1/2,j} = -a_{i\pm1/2,j}\Delta y J_x(x_{c_{i\pm1/2}}, y_j), \qquad F_{i,j\pm1/2} = -a_{i,j\pm1/2}\Delta x J_y(x_i, y_{c_{j\pm1/2}}).$$

With this notation, we rewrite the previous equation as

$$\frac{\partial c_{i,j}}{\partial t} - f(c_{i,j}) \approx \frac{1}{V_{i,j}}\big(F_{i+1/2,j} - F_{i-1/2,j} + F_{i,j+1/2} - F_{i,j-1/2} - F^f_{i,j}\big). \quad (11)$$

We refer to the Fs as the surface fluxes (Figure 3). On a full edge with $a_{l,m} = 1$ the surface flux is calculated with centered differences. For example, in Figure 3, we have

$$F_{i-1/2,j+1} = D\Delta y\frac{c_{i,j+1} - c_{i-1,j+1}}{\Delta x}. \quad (12)$$

The flux gradient across a cut edge, for example $(x_{i-1/2}, y_j)$, is approximated by a linear interpolation of two gradients, which are computed by centered differences. A linear interpolation formula between two points y_1 and y_2 as a function of a

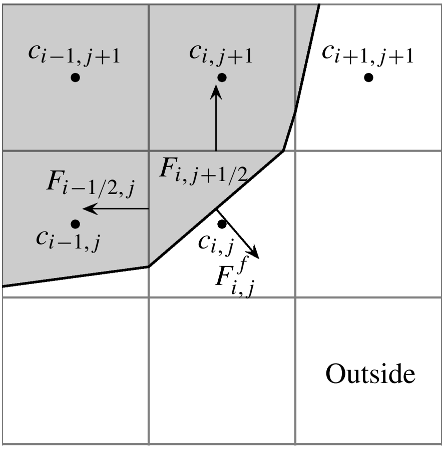


Figure 3. Diagram of fluxes for cut cells where shaded boxes indicate cells that are inside the boundary.

parameter $\mu \in [0, 1]$ is

$$y^I = (1 - \mu)y_1 + \mu y_2. \tag{13}$$

In the case of a cut-cell edge, $\mu = (1 + a_{l,m})/2$. For example, to construct $F_{i-1/2,j}$ in Figure 3, the gradient at $(x_{i-1/2}, y_j)$ and $(x_{i-1/2}, y_{j+1})$ is used:

$$F_{i-1/2,j} = Da_{i-1/2,j}\,\Delta y\Bigg[\frac{(1+a_{i-1/2,j})}{2}\frac{(c_{i,j} - c_{i-1,j})}{\Delta x} + \frac{(1-a_{i-1/2,j})}{2}\frac{(c_{i,j+1} - c_{i-1,j+1})}{\Delta x}\Bigg]. \tag{14}$$

To calculate the flux through a boundary, for example, $F^f_{i,j}$, we compute the gradient along a line normal to the boundary and centered at the boundary midpoint. To find function values on the normal line, we interpolate using three equally spaced cell-centered points (Figure 4). If the normal line is oriented with an angle of $\pi/4 < |\theta| < 3\pi/4$ relative to the horizontal grid lines, horizontal grid points are used to compute the values on the line. Otherwise vertical points are used.

The two points computed along the normal line and the value on the boundary are then used to construct a quadratic polynomial. The concentration gradient is calculated by differentiating the quadratic polynomial and evaluating the result at the boundary point c^f:

$$G^f = \frac{1}{d_2 - d_1}\left[\frac{d_2}{d_1}(c^f - c^I_1) - \frac{d_1}{d_2}(c^f - c^I_2)\right], \tag{15}$$

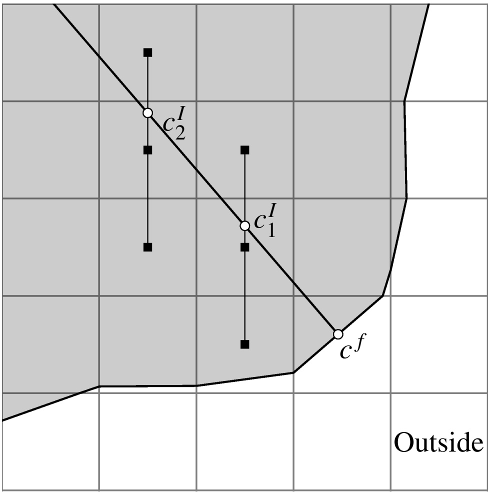


Figure 4. White circles indicate interpolated values that depend on the grid-based values.

where c_1^I and c_2^I are the interpolated values along the normal line and d_1 and d_2, respectively, are the distances of these two points from the boundary. The flux F_{ij}^f in (11) is calculated by multiplying G^f by the area of the cut-cell edge a_f and the diffusion coefficient D. The discretization of the boundary condition (2) becomes the algebraic equation

$$DG^f + g(\boldsymbol{c}^f) = 0. \tag{16}$$

Because all gradients are constructed with second-order methods, the overall discretization scheme is second-order in space. Further discussion on the accuracy of the spatial discretization scheme can be found in [9].

3.3. ***Time discretization.*** Spatial discretizations of (1) and (2) are treated as a differential-algebraic system of nonlinear equations (DAE). The general form for a differential-algebraic system is

$$F(t, \boldsymbol{C}, \boldsymbol{C}') = \boldsymbol{0}, \tag{17}$$

where $\boldsymbol{C}$ is an $(N_g + N_b) \times 1$ vector. The first N_g entries are associated with Cartesian grid based values in the differential-algebraic system from the discretization of (1) for the chemical species concentrations. These entries have an explicit time derivative term. The N_b remaining entries arise from discretizing the boundary conditions given in (2) that form algebraic constraints. As noted in [1], reformulating algebraic constraints in a nonlinear model as a system of ordinary differential equations may be time consuming or impossible. DAEs formed by reaction-diffusion equations described in Section 2 are semiexplicit, index-1 systems of the form

$$\begin{aligned} \boldsymbol{C}_1' &= F_1(\boldsymbol{C}_1, \boldsymbol{C}_2, t), \\ \boldsymbol{0} &= F_2(\boldsymbol{C}_1, \boldsymbol{C}_2, t). \end{aligned} \tag{18}$$

$\boldsymbol{C}_1$ represents the first N_g variables and $\boldsymbol{C}_2$ represents the remaining N_b variables. Equation (18) is an index-1 system if and only if $\partial F_2/\partial \boldsymbol{C}_2$ is nonsingular [1]. Ordinary differential equations are index-0.

We use the DASPK solver described in [2] as a time integrator for our differential algebraic system. In DAPSK, backward differentiation formulas (BDF) discretize the time derivative in (17). A basic implicit method with a backward Euler time discretization of (17) is given by,

$$F\Big(t^{n+1}, \boldsymbol{C}^{n+1}, \frac{\boldsymbol{C}^{n+1} - \boldsymbol{C}^n}{\Delta t}\Big) = \boldsymbol{0}, \tag{19}$$

where n is defined such that $t^n = n\Delta t$. Newton's method can be used to solve the resulting nonlinear equations for $\boldsymbol{C}^{n+1}$,

$$\boldsymbol{C}_{m+1}^{n+1} = \boldsymbol{C}_m^{n+1} - \Big(\frac{\partial F}{\partial \boldsymbol{C}} + \frac{1}{\Delta t}\frac{\partial F}{\partial \boldsymbol{C}'}\Big)\Big|_{\boldsymbol{C}_m^{n+1}}^{-1} F\Big(t^{n+1}, \boldsymbol{C}_m^{n+1}, \frac{\boldsymbol{C}_m^{n+1} - \boldsymbol{C}^n}{\Delta t}\Big), \tag{20}$$

where m is the index of the Newton iteration. In order to achieve higher-order temporal accuracy, a higher-order interpolating polynomial is used to approximate the time derivative.

In a k-step BDF, the time derivative is replaced by the derivative of an interpolating polynomial at $k+1$ times $t^{n+1}, t^n \ldots, t^{n+1-k}$ evaluated at t^{n+1}. If we approximate the derivative using a kth order stencil using k known values and the implicit value $\boldsymbol{C}^{n+1}$ we get

$$\boldsymbol{C}'^{\,n+1} \approx \frac{1}{\Delta t}\left(\alpha_0 \boldsymbol{C}^{n+1} + \sum_{i=1}^{k} \alpha_i \boldsymbol{C}^{n+1-i}\right). \tag{21}$$

The coefficients of the BDF are given by α_is. In DAPSK, these values are coefficients of the Newton divided difference interpolating polynomial [1]. The default order of the BDF method in DASPK is five.

The new implicit equation to be solved at each time step is

$$F\left(t^{n+1}, \boldsymbol{C}^{n+1}, \frac{1}{\Delta t}\left(\alpha_0 \boldsymbol{C}^{n+1} + \sum_{i=1}^{k} \alpha_i \boldsymbol{C}^{n+1-i}\right)\right) = \mathbf{0}. \tag{22}$$

This can be rewritten as

$$F\left(t^{n+1}, \boldsymbol{C}^{n+1}, \frac{\alpha_0}{\Delta t}\boldsymbol{C}^{n+1} + \mathbf{v}\right) = \mathbf{0}, \tag{23}$$

where $\mathbf{v}$ is a vector that depends on previously computed time values. Details of choosing step-size, starting selection and variable order strategies are found in [1]. The nonlinear system is solved with a modified Newton's method, given by

$$\boldsymbol{C}_{m+1}^{n+1} = \boldsymbol{C}_m^n - \zeta\left(\frac{\partial F}{\partial \boldsymbol{C}} + \frac{\alpha_0}{\Delta t}\frac{\partial F}{\partial \boldsymbol{C}'}\right)\Bigg|_{\boldsymbol{C}_m^{n+1}}^{-1} F\left(t^{n+1}, \boldsymbol{C}_m^{n+1}, \frac{\alpha_0}{\Delta t}\boldsymbol{C}_m^{n+1} + \mathbf{v}\right), \tag{24}$$

where ζ is a constant chosen to speed up convergence and m is the iteration index. Each step of the Newton iteration requires inverting the matrix

$$A = \frac{\partial F}{\partial \boldsymbol{C}} + \frac{\alpha_0}{\Delta t}\frac{\partial F}{\partial \boldsymbol{C}'}. \tag{25}$$

We store this matrix in sparse triple format, and use routines from SPARSKIT [20] to solve the linear system iteratively. The generalized minimal residual (GMRES) method [21] with an incomplete LU (ILU) preconditioner is used to solve the linear system.

4. Results

4.1. ***Convergence tests.*** To demonstrate the accuracy of our method on a domain containing all types of cut cells, the convergence of our method is compared against an exact solution on a circle. The exact solution to the diffusion equation with a zero Dirichlet boundary condition can be found in terms of Bessel functions. Let λ denote the first root of the Bessel function $J_0(x)$, and r be the radius of the circle centered at the point $(0.5, 0.5)$. Then the expression

$$f(x,y,t) = \exp\Big(-D\Big(\frac{\lambda}{r}\Big)^2 t\Big)\, J_0\bigg(\lambda \frac{\sqrt{(x-0.5)^2+(y-0.5)^2}}{r}\bigg) \tag{26}$$

is an exact solution to the diffusion equation.

For this example, the error is computed as the difference between computed solution values on a triangular grid subtracted from the exact solution. The grids for both two dimensional triangular meshes were the same. For purposes of generating the following convergence data, the spatial steps Δx and Δy are equal and set to $1/N$, where N is the grid size. The time step Δt is set to $\Delta x/4$ (that is, it is refined with the spatial step size). Because DASPK uses variable time steps, the output at the time step requested might be interpolated as described in [1]. A time series of the truncation error in the infinity norm over time is shown in Figure 5. Table 1 lists the truncation error at the simulation time $t = 0.4$. The convergence rate r is calculated as

$$r = \log \frac{e_1}{e_2} \Big/ \log \frac{\Delta x_1}{\Delta x_2}, \tag{27}$$

where e_1 and e_2 are errors computed in norms with grid spacing Δx_1 and Δx_2. A log-log plot of truncation error as a function of the spatial step is shown in Figure 6. The error was calculated with the computed and exact solutions at the time value of $t = 0.4$. The results of this analysis demonstrate global second-order accuracy of the numerical method.

Next we tested a nonlinear system in which a protein C can exist in two distinct chemical states: active and inactive. The reactions that convert the protein between the two states are assumed to follow Michaelis–Menten kinetics, which describes the kinetics of many enzymatic reactions including phosphorylation and dephosphorylation events [11]. The protein C is deactivated in the interior of the

Grid size	Time step	L^2 norm	r	L^1 norm	r	L^∞ norm	r
50×50	$5.00{\cdot}10^{-3}$	$2.95{\cdot}10^{-4}$	–	$2.61{\cdot}10^{-4}$	–	$5.46{\cdot}10^{-4}$	–
100×100	$2.50{\cdot}10^{-3}$	$4.94{\cdot}10^{-5}$	2.58	$4.32{\cdot}10^{-5}$	2.59	$9.28{\cdot}10^{-5}$	2.56
200×200	$1.25{\cdot}10^{-3}$	$1.05{\cdot}10^{-5}$	2.24	$9.20{\cdot}10^{-6}$	2.23	$2.09{\cdot}10^{-5}$	2.15
400×400	$6.25{\cdot}10^{-4}$	$2.42{\cdot}10^{-6}$	2.11	$2.13{\cdot}10^{-6}$	2.11	$5.42{\cdot}10^{-6}$	1.95

Table 1. The norms and convergence rates for the diffusion equation at the time value of 0.4.

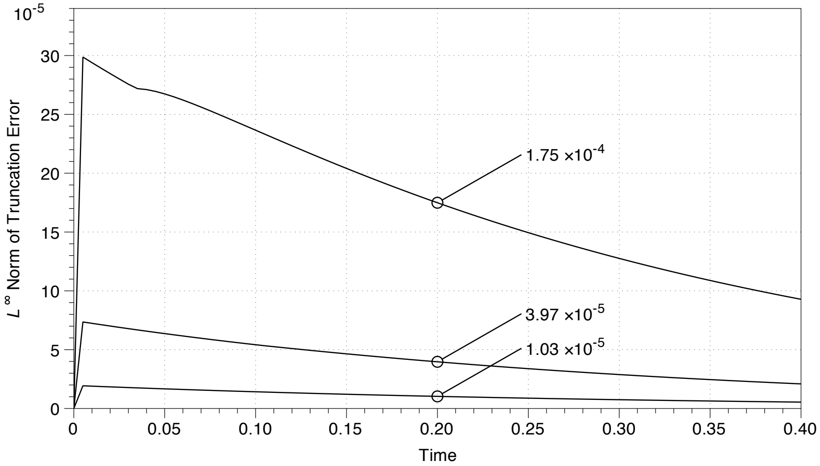


Figure 5. L^∞-norm truncation error for the diffusion equation, with size $N = 100, 200, 400$ (from top to bottom). The time step at each refinement was set to $1/(4N)$.

computational domain according to the following equations:

$$\frac{\partial C_i}{\partial t} = D\Delta C_i + \frac{k_2 C_a}{K_{m2}+C_a}, \qquad \frac{\partial C_a}{\partial t} = D\Delta C_a - \frac{k_2 C_a}{k_{m2}+C_a}, \tag{28}$$

where C_i and C_a are the concentrations of inactive and active protein, respectively, k_2 is the maximum deactivation rate, and K_{m2} is the Michaelis constant. Activation

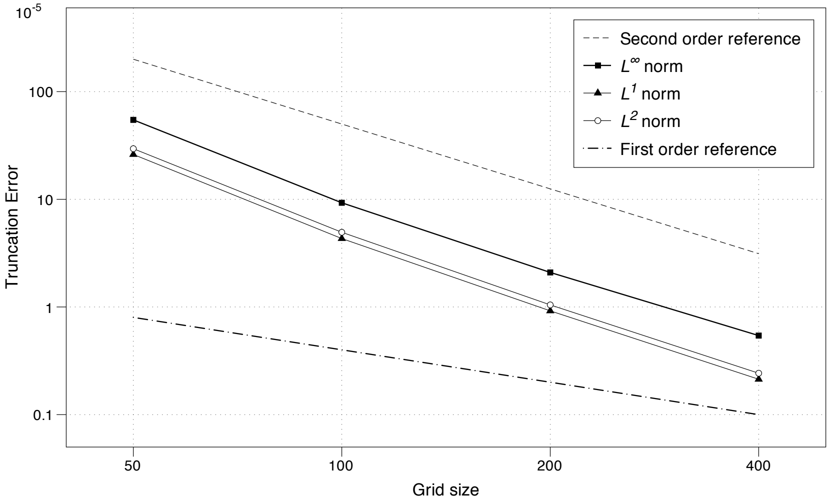


Figure 6. Truncation error for the diffusion equation at the time value of 0.4. The convergence data is the same as in Table 1.

occurs on the boundary, $\partial\Omega$, according to the following boundary conditions:

$$-D\vec{n}\cdot\nabla C_i|_{\partial\Omega} = \frac{k_1 S C_i}{K_{m1}+C_i}\bigg|_{\partial\Omega}, \qquad -D\vec{n}\cdot\nabla C_a|_{\partial\Omega} = -\frac{k_1 S C_i}{K_{m1}+C_i}\bigg|_{\partial\Omega}, \tag{29}$$

where k_1 is the maximum activation rate and K_{m1} is the Michaelis constant. The equations are solved in the domain

$$\Omega(r,\theta) = r \leq 0.3 - 0.09\sin(4\theta). \tag{30}$$

In our simulation, Ω is shifted to the center of the unit box. The initial concentration of inactive protein is assumed to be constant and equal to 1. There is initially no active protein. Figure 7 shows a plot of the active concentration at $t = 0.25$. For visualization purposes, the computational domain and boundary points are triangulated with Triangle [27]. The concentration of the active protein is shown as a cross-section of the two-dimensional geometry at several time values in Figure 7, bottom. The constants (see figure caption) were arbitrarily chosen to generate a gradient. Execution times for Mac Pro desktop computer with dual-core 2.66 GHz Intel Xeon processors for different grid sizes are listed in Table 2.

We compute the error as the difference between successive grid refinements as follows. The truncation error function $E(x, y, t)$ is defined on interior values of the coarser grid. Computed solution values located in coarse grid cut cells are excluded from the domain. This includes some values located in interior points for the more refined grid (Figure 8). The truncation error function is defined as

$$E(x, y, t) = c_{\Delta x}(x, y, t) - c_{\Delta x/2}(x, y, t). \tag{31}$$

The coarse grid values are located in the center of a box defined by four refined grid values. Four refined grid values are averaged and subtracted from one coarse value. Because the time integration is handled implicitly, a different convergence rate of the truncation error in cut cells and boundary values would affect the convergence

		Execution
Grid size	Time step	time
50 × 50	$5.000\cdot 10^{-3}$	1.76 s
100 × 100	$2.500\cdot 10^{-3}$	5.81 s
200 × 200	$1.250\cdot 10^{-3}$	32.15 s
400 × 400	$6.250\cdot 10^{-4}$	203.95 s
800 × 800	$3.125\cdot 10^{-4}$	1202.18 s

Table 2. Execution times for the two-species model. The end time of the simulation was $t = 0.5$.

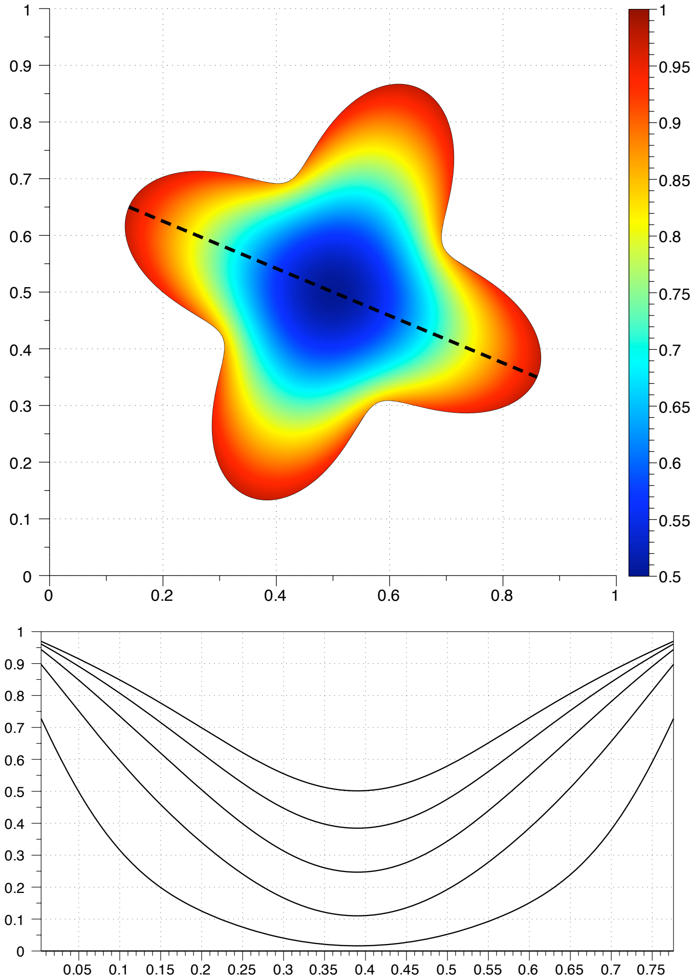


Figure 7. Concentration of the active species C_a at $t = 0.25$ (top) and at evenly spaced time values for $t \in [0, 0.25]$ (bottom) along the section shown with a dashed line in the top figure. Values chosen for the constants: $D = K_{m1} = k_{m2} = 0.2$, $S = k_1 = k_2 = 1.0$.

rate of the truncation error for interior cells. Therefore, by computing the error with interior cells, we are still able to draw conclusions about the order of the method.

Table 3 lists convergence data for the two-species system given by (28) and (29). The data used for calculating the error was taken from computed solutions at the simulation time value of $t = 0.5$. Note that the norms of truncation errors for both C_i and C_a are the same. The system is mass conservative, and the computed solution is also conservative to machine precision. Therefore we only show convergence

Species C_i							
Grid size	Time step	L^2 norm	r	L^1 norm	r	L^∞ norm	r
50×50	$5.000{\cdot}10^{-3}$	–	–	–	–	–	–
100×100	$2.500{\cdot}10^{-3}$	$7.59{\cdot}10^{-4}$	–	$1.67{\cdot}10^{-4}$	–	$1.49{\cdot}10^{-3}$	–
200×200	$1.250{\cdot}10^{-3}$	$1.92{\cdot}10^{-4}$	1.98	$4.44{\cdot}10^{-5}$	1.91	$5.01{\cdot}10^{-4}$	1.57
400×400	$6.250{\cdot}10^{-4}$	$4.57{\cdot}10^{-5}$	2.07	$1.08{\cdot}10^{-5}$	2.04	$1.25{\cdot}10^{-4}$	2.00
800×800	$3.125{\cdot}10^{-4}$	$1.09{\cdot}10^{-5}$	2.07	$2.61{\cdot}10^{-6}$	2.05	$3.12{\cdot}10^{-5}$	2.00
Species C_a							
Grid size	Time step	L^2 norm	r	L^1 norm	r	L^∞ norm	r
50×50	$5.000{\cdot}10^{-3}$	–	–	–	–	–	–
100×100	$2.500{\cdot}10^{-3}$	$7.59{\cdot}10^{-4}$	–	$1.67{\cdot}10^{-4}$	–	$1.49{\cdot}10^{-3}$	–
200×200	$1.250{\cdot}10^{-3}$	$1.92{\cdot}10^{-4}$	1.98	$4.44{\cdot}10^{-5}$	1.91	$5.01{\cdot}10^{-4}$	1.57
400×400	$6.250{\cdot}10^{-4}$	$4.57{\cdot}10^{-5}$	2.07	$1.08{\cdot}10^{-5}$	2.04	$12.5{\cdot}10^{-4}$	2.00
800×800	$3.125{\cdot}10^{-4}$	$1.09{\cdot}10^{-5}$	2.07	$2.61{\cdot}10^{-6}$	2.05	$3.12{\cdot}10^{-5}$	2.00

Table 3. Norms and convergence rates for the two-species model at the time value of 0.5.

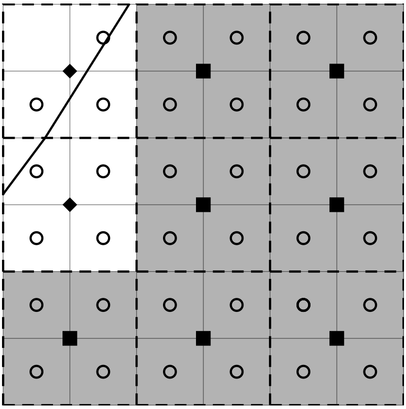


Figure 8. Interior grid cells on the coarser grid (dashed lines) are shaded. Square- and diamond-filled points indicate locations of cell-centered values on the coarse grid. Values associated with diamond-grid points represent cut cells for the coarser grid. Coarse and refined values in these cut cells are not used in the averaging scheme. The refined grid is indicated by solid lines. Circles mark the cell centers of the refined grid cells. Four-refined point values are averaged and compared to the square point on the coarse grid.

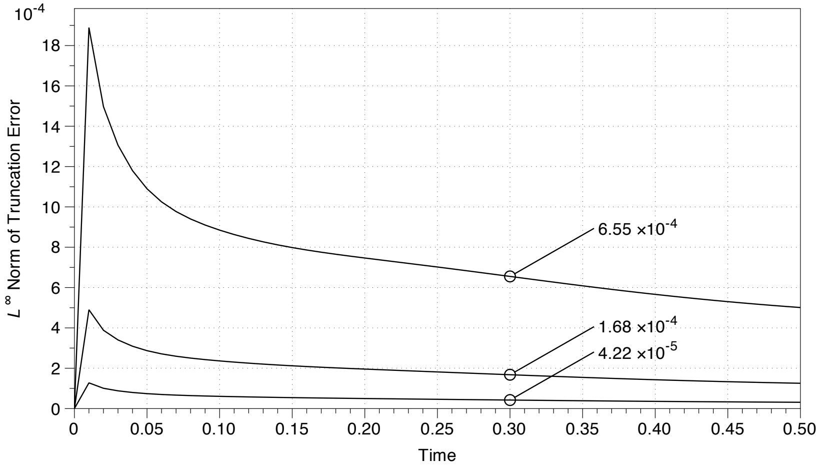


Figure 9. L^∞-norm truncation error for species C_i for the reaction-diffusion equation. The values for the top plot were computed by subtracting the solution at grid size $N = 200$ from the one at $N = 100$ (see text). The middle plot was calculated with $N = 200$ and $N = 400$, and the bottom one with $N = 400$ and $N = 800$.

figures for species C_i. The truncation error for species C_i computed in the infinity norm as a function of time is listed in Figure 9. A log-log plot of the truncation error as a function of the grid size is listed in Figure 10. From this analysis, we conclude second-order accuracy.

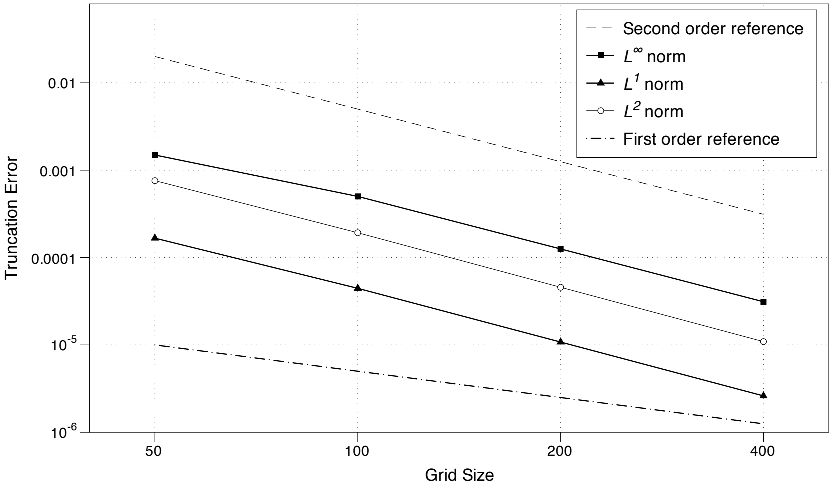


Figure 10. Truncation error for the C_i at the time value of 0.5. The convergence data is the same as that in Table 3.

4.2. ***A two-compartment model.*** In this model, we have two compartments: cytoplasm and nucleus. The cellular geometry was taken from a yeast cell undergoing chemotrophic growth in the direction of a pheromone gradient [8]. Proteins involved in the pheromone response pathway are known to localize on the plasma membrane, the nucleus, and in the cytosol [7]. The nucleus is modeled as a circle located toward the front of the cell. Because yeast cells are three dimensional, we model the top view of the cell as in [6], where membrane-bound species are located in the interior of the computational domain but are assumed to diffuse slower than cytosolic forms.

The model consists of two species, A and C, with inactive and active forms. Protein C is allowed to enter and exit the nucleus, whereas protein A is restricted to the cytoplasm (Figure 11, top).

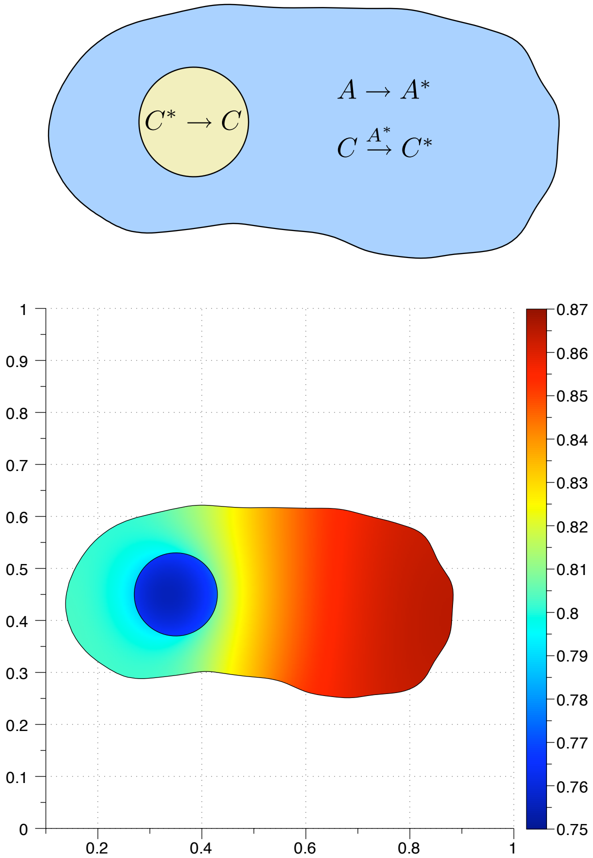


Figure 11. Two-compartment model. Top: reactions and species in the two-compartment model. Bottom: steady-state concentration values for active C species in the cytoplasm and nucleus.

Initially both A and C are in their inactive forms. At the beginning of the simulation, the reaction rate for the activation of A, k_0, is instantaneously increased from 0 to 1. This is meant to model the cell receiving an external signal. Once A is activated it is assumed to interact with the cell membrane, causing a reduction in the protein's diffusion coefficient [30]. The active form of A can then activate protein C. The active form of C is only deactivated within the nucleus. This simple model captures some of the signaling events that occur during the pheromone response of yeast [28]. If we denote the concentration of a chemical species with brackets, the equations for the cytoplasmic species are:

$$\begin{aligned} \frac{\partial[A_c]}{\partial t} &= D_1\Delta[A_c]-k_0[A_c], & \frac{\partial[A_c^*]}{\partial t} &= D_2\Delta[A_c^*]+k_0[A_c], \\ \frac{\partial[C_c]}{\partial t} &= D_1\Delta[C_c]-k_1[A_c^*][C_c], & \frac{\partial[C_c^*]}{\partial t} &= D_1\Delta[C_c^*]+k_1[A_c^*][C_c], \end{aligned} \tag{32}$$

where the asterisks denote the active form of the protein, D_1 is the diffusion coefficient in the cytoplasm, D_2 is diffusion coefficient in the membrane, and the ks represent the reaction rates. Subscripts indicate cytosolic and nuclear species. The boundary conditions at the cell membrane $\partial\mathbf{\Omega}_1$ are no flux for all chemical species. The nuclear boundary conditions for A species are also no flux, whereas C species are allowed to move through the nuclear membrane $\partial\mathbf{\Omega}_2$ and satisfy the conditions

$$\begin{aligned} -D_1(\vec{n}\cdot\nabla[C_c])|_{\partial\Omega_2} &= -k_2([C_n]-[C_c])|_{\partial\Omega_2}, \\ -D_1(\vec{n}\cdot\nabla[C_c^*])|_{\partial\Omega_2} &= -k_2([C_n^*]-[C_c^*])|_{\partial\Omega_2}, \\ -D_1(\vec{n}\cdot\nabla[C_n])|_{\partial\Omega_2} &= k_2([C_n]-[C_c])|_{\partial\Omega_2}, \\ -D_1(\vec{n}\cdot\nabla[C_n^*])|_{\partial\Omega_2} &= k_2([C_n^*]-[C_c^*])|_{\partial\Omega_2}. \end{aligned} \tag{33}$$

Nuclear C^* is deactivated according to the equations

$$\frac{\partial[C_n]}{\partial t} = D_1\Delta[C_n]+k_3[C_n^*], \quad \frac{\partial[C_n^*]}{\partial t} = D_1\Delta[C_n^*]-k_3[C_n^*]. \tag{34}$$

The steady-state spatial distribution of active C is illustrated in Figure 11, bottom. All reaction constants were arbitrarily chosen to be 1, $D_1=0.1$, $D_2=0.01$, and $\Delta x = 1/200$. The initial values were zero except for $[A_c](x,y,0) = [C_c](x,y,0) = 1$. The execution time of the simulation to run from $t=0$ until $t=20$ was 150 seconds on Mac Pro desktop computer with dual-core 2.66 GHz Intel Xeon processors.

To verify that the system is close a steady-state solution at $t=20$, we subtracted the solution of active C in the cytoplasm $[C_c^*]$ for all times from the assumed steady-state solution at the time value of $t=20$. If the system exponentially converges to the computed solution at $t=20$, we assume this time value is close to steady-state. Figure 12 shows the infinity norm of the difference between the computed solution

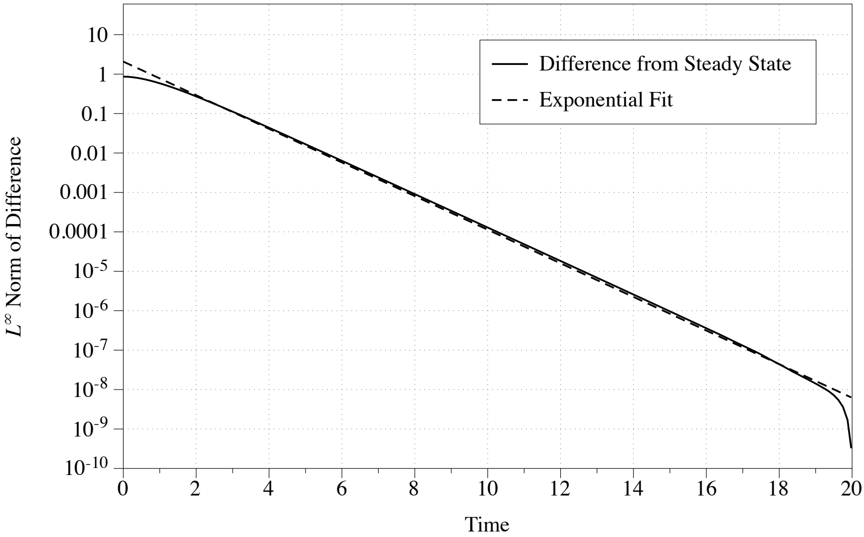
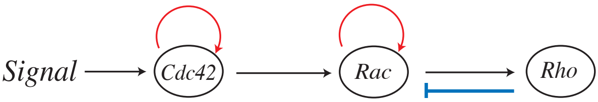


Figure 12. Solid line: norm of difference of the computed solution at the assumed steady-state value at $t = 20$ from computed solution over time. Dashed line: exponential fit.

and the solution at $t = 20$ sampled over time. Based on this data, the system is close to its steady-state solution.

The model simulation suggests a spatial activation gradient can be generated by the position of the nucleus. The inactivation of C in the nucleus leads to a higher concentration of active protein in the rear of the cell in spite of a uniform spatial signal from active A.

4.3. ***Rho family GTPase model.*** The Rho family of GTPases regulates many cellular functions, including polarization and motility. We created a model with three key members of this family, Cdc42, Rac, and Rho; the interactions, based on [3], can be schematically represented as follows:



A more complicated model involving these proteins in one dimension can be found in [6]. As in the previous example, we assume a top view of a three dimensional cell with membrane bound active forms and cytosolic inactive forms of the three proteins. The model has a total of six species. The cell boundary $\partial\Omega$ is taken from supplemental material from [19].

In our model, a uniform extracellular signal triggers the activation of Cdc42 protein on the cell edge,

$$\begin{aligned} -D\vec{n}\cdot\nabla\,[\mathrm{Cdc42}_i]\,\big|_{\partial\Omega} &= \frac{k_1 S\,[\mathrm{Cdc42}_i]}{K_{m1}+[\mathrm{Cdc42}_i]}\bigg|_{\partial\Omega}, \\ -D\vec{n}\cdot\nabla\,[\mathrm{Cdc42}_a]\,\big|_{\partial\Omega} &= -\frac{k_1 S\,[\mathrm{Cdc42}_i]}{k_{m2}+[\mathrm{Cdc42}_i]}\bigg|_{\partial\Omega}. \end{aligned} \tag{35}$$

In the cell interior, active Cdc42 is inactivated. A positive feedback loop increases the activation of Cdc42,

$$\begin{aligned} \frac{\partial\,[\mathrm{Cdc42}_i]}{\partial t} &= D\,\Delta\,[\mathrm{Cdc42}_i] + \frac{k_2\,[\mathrm{Cdc42}_a]}{K_{m3}+[\mathrm{Cdc42}_a]} - \frac{k_3\,[\mathrm{Cdc42}_a]\,[\mathrm{Cdc42}_i]}{K_{m4}+[\mathrm{Cdc42}_i]}, \\ \frac{\partial\,[\mathrm{Cdc42}_a]}{\partial t} &= D\,\Delta\,[\mathrm{Cdc42}_a] - \frac{k_2\,[\mathrm{Cdc42}_a]}{k_{m5}+[\mathrm{Cdc42}_a]} + \frac{k_3\,[\mathrm{Cdc42}_a]\,[\mathrm{Cdc42}_i]}{k_{m6}+[\mathrm{Cdc42}_i]}. \end{aligned} \tag{36}$$

Rac is activated by Cdc42, and a positive feedback loop increases the concentration of active Rac. Active Rho increases the deactivation of Rac in the cytosol,

$$\begin{aligned} \frac{\partial[\mathrm{Rac}_i]}{\partial t} &= D\Delta[\mathrm{Rac}_i] + \frac{(k_4[\mathrm{Rho}_a]+k_5)[\mathrm{Rac}_a]}{K_{m7}+[\mathrm{Rac}_a]} - \frac{(k_6[\mathrm{Cdc42}_a]+k_7[\mathrm{Rac}_a])[\mathrm{Rac}_i]}{K_{m8}+[\mathrm{Rac}_i]}, \\ \frac{\partial[\mathrm{Rac}_a]}{\partial t} &= D\Delta[\mathrm{Rac}_a] - \frac{(k_4[\mathrm{Rho}_a]+k_5)[\mathrm{Rac}_a]}{k_{m9}+[\mathrm{Rac}_a]} + \frac{(k_6[\mathrm{Cdc42}_a]+k_7[\mathrm{Rac}_a])[\mathrm{Rac}_i]}{k_{m10}+[\mathrm{Rac}_i]}. \end{aligned}$$

Rho is activated by the active form of Rac and deactivated in the interior,

$$\begin{aligned} \frac{\partial\,[\mathrm{Rho}_i]}{\partial t} &= D\,\Delta\,[\mathrm{Rho}_i] + \frac{k_8\,[\mathrm{Rho}_a]}{K_{m11}+[\mathrm{Rho}_a]} - \frac{k_9\,[\mathrm{Rac}_a]\,[\mathrm{Rho}_i]}{K_{m12}+[\mathrm{Rho}_i]}, \\ \frac{\partial\,[\mathrm{Rho}_a]}{\partial t} &= D\,\Delta\,[\mathrm{Rho}_a] - \frac{k_8\,[\mathrm{Rho}_a]}{k_{m13}+[\mathrm{Rho}_a]} + \frac{k_9\,[\mathrm{Rac}_a]\,[\mathrm{Rho}_i]}{k_{m14}+[\mathrm{Rho}_i]}. \end{aligned} \tag{37}$$

The boundary conditions for Rac and Rho species are no flux. The steady-state distribution is displayed in Figure 13. To achieve these results, a step size $\Delta x = 1/200$ and a diffusion coefficient $D = 0.1$ were used. The reaction constants for the simulation were arbitrarily chosen as follows: $S = k_3 = k_5 = k_7 = 1.0$, $k_2 = k_4 = k_8 = 3.0$, $k_1 = k_6 = k_9 = 5.0$, and all K_{mi} and k_{mi} equal to 0.2.

The initial concentration of inactive chemical species was set to one and zero for active species. The execution time was 217 seconds for 1600 time steps on a Mac Pro desktop computer with dual-core 2.66 GHz Intel Xeon processors.

In this model, a gradient is formed by protein activation on the cell edge, and propagated to the downstream signaling components Rac and Rho. Figure 13 shows that filopodia and thin protrusions have higher activation levels due the increased ratio of cell membrane to cell volume in these regions [16].

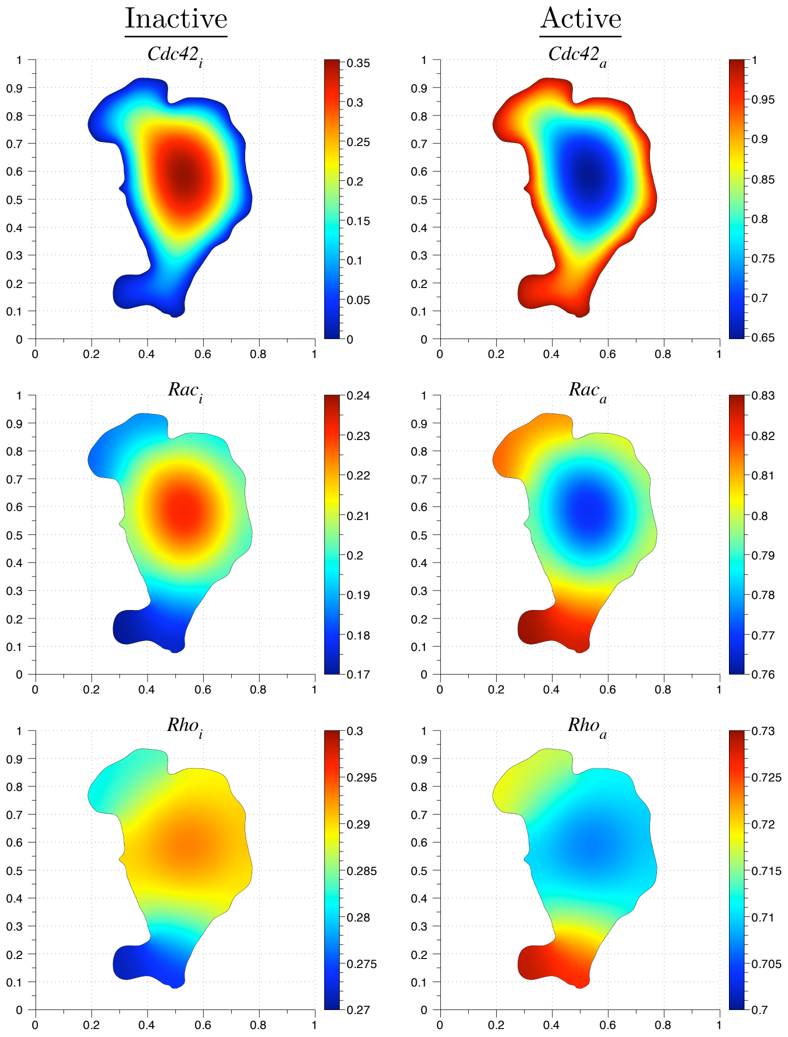


Figure 13. Rho GTPase model: steady-state distribution of protein concentration amounts in a fibroblast. The boundary was taken from a live cell image [19]. Values chosen for the constants: $S = k_3 = k_5 = k_7 = 1.0$, $k_2 = k_4 = k_8 = 3.0$, $k_1 = k_6 = k_9 = 5.0$, and all K_{mi} and k_{mi} equal to 0.2.

5. Conclusions

We have developed an accurate and efficient cut-cell method for simulating spatial models of signaling pathways in realistic cellular geometries. We demonstrated our method using models that consist of multiple species interacting in multiple compartments. The examples were chosen to illustrate the numerical methods and therefore lack many details found in real biological signaling systems. In particular, feedback and feed forward control mechanisms that regulate pathway activity were not considered in detail. Our numerical methods provide important tools for investigating such regulatory mechanisms in realistic cell geometries and, therefore, should provide important insights into the ways signaling networks process and transmit information.

Our algorithm extends previous work on embedded boundary methods [5; 9; 15; 25]. These methods have been implemented in two and three dimension for Poisson's equation, the heat equation, and hyperbolic conservation laws. Our formulation extends these methods to systems of reaction-diffusion equations with nonlinear reactions in the interior as well as nonlinear reactions affecting boundary values. The boundary conditions treated in previous work [9; 15; 25] have been homogeneous Dirichlet and Neumann, which is not sufficient for many models of signaling pathways [16]. In [15], a second-order implicit method was used to update the heat equation in time [29]. In our method, we use an implicit nonlinear solver to handle nonlinear reactions occurring in the interior. An advantage of the differential-algebraic formulation is the ability to treat the boundary conditions as algebraic constraints. This allows us to handle reactions that take place on the physical boundary of the reaction-diffusion equation.

One limitation of the finite volume discretization arises from the interpolation method to obtain the normal derivative to the surface as shown in Figure 4. Cut cells must not have a zero volume cell within two rows or columns. For biological cells with long, thin or irregularly-shaped components such as neurons, mesh adaptive refinement may be needed to resolve the cellular geometry.

The underlying Cartesian-grid based finite volume discretization allows us to use advection schemes originally developed for hyperbolic conservation laws to simulate active transport or motility. In future reports, we will show how level set methods [18; 26] can be combined with biochemical reaction networks to investigate the effect of moving boundaries on cell signaling. Future work also includes a three-dimensional implementation of our fixed boundary algorithm. A three-dimensional extension of our method could be coupled with the method for simulating diffusion on a surface presented in [24] to obtain an algorithm for simulating models that take into account processes that occur both in the cytoplasm and on the cell membrane.

6. Acknowledgments

We thank Meng Jin and Yi Wu for insightful discussions. This work was supported by NIH grants R01-GM079271 and R01-GM078994.

References

[1] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical solution of initial-value problems in differential-algebraic equations*, Classics in Applied Mathematics, no. 14, SIAM, Philadelphia, 1996. MR 96h:65083 Zbl 0844.65058

[2] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold, *Using Krylov methods in the solution of large-scale differential-algebraic systems*, SIAM J. Sci. Comput. **15** (1994), no. 6, 1467–1488. MR 95g:65092 Zbl 0812.65060

[3] K. Burridge and K. Wennerberg, *Rac and Rho take center stage*, Cell **116** (2004), 167–179.

[4] P. Colella, D. Graves, T. Ligocki, D. Trebotich, and B. V. Straalen, *Embedded boundary algorithms and software for partial differential equations*, J. Phys. Conf. Ser. **125** (2008), 012084.

[5] P. Colella, D. T. Graves, B. J. Keen, and D. Modiano, *A Cartesian grid embedded boundary method for hyperbolic conservation laws*, J. Comput. Phys. **211** (2006), no. 1, 347–366. MR 2006i:65142 Zbl 1120.65324

[6] A. T. Dawes and L. Edelstein-Keshet, *Phosphoinositides and Rho proteins spatially regulate actin polymerization to initiate and maintain directed movement in a one-dimensional model of a motile cell*, Biophys. J. **92** (2007), no. 3, 744–768.

[7] H. G. Dohlman, *G Proteins and pheromone signaling*, Annu. Rev. Physiol. **64** (2002), 129–152.

[8] N. Hao, S. Nayak, M. Behar, R. H. Shanks, M. J. Nagiec, B. Errede, J. Hasty, and T. C. Elston, *Regulation of cell signaling dynamics by the protein kinase-scaffold ste5*, Mol. Cell. **30** (2008), no. 5, 649–656.

[9] H. Johansen and P. Colella, *A Cartesian grid embedded boundary method for Poisson's equation on irregular domains*, J. Comput. Phys. **147** (1998), no. 1, 60–85. MR 99m:65231 Zbl 0923.65079

[10] B. N. Kholodenko, *Cell-signalling dynamics in time and space*, Nat. Rev. Mol. Cell. Biol. **7** (2006), no. 3, 165–176.

[11] B. N. Kholodenko, G. C. Brown, and J. B. Hoek, *Diffusion control of protein phosphorylation in signal transduction pathways*, Biochem. J. **350** (2000), no. Pt. 3, 901–907.

[12] H. Levine, D. A. Kessler, and W. J. Rappel, *Directional sensing in eukaryotic chemotaxis: A balanced inactivation model*, Proc. Natl. Acad. Sci. USA. **103** (2006), no. 26, 9761–9766.

[13] T. J. Ligocki, P. O. Schwartz, J. Percelay, and P. Colella, *Embedded boundary grid generation using the divergence theorem, implicit functions, and constructive solid geometry*, J. Phys. Conf. Ser. **125** (2008), 012080 (5pp).

[14] R. Malladi, J. A. Sethian, and B. C. Vemuri, *A fast level set based algorithm for topology-independent shape modeling*, J. Math. Imaging Vision **6** (1996), 269–289. MR 97a:68174

[15] P. McCorquodale, P. Colella, and H. Johansen, *A Cartesian grid embedded boundary method for the heat equation on irregular domains*, J. Comput. Phys. **173** (2001), no. 2, 620–635. MR 2002h:80009 Zbl 0991.65099

[16] J. Meyers, J. Craig, and D. J. Odde, *Potential for control of signaling pathways via cell size and shape*, Curr. Biol. **16** (2006), no. 17, 1685–1693.

[17] S. R. Neves, P. Tsokas, A. Sarkar, E. A. Grace, P. Rangamani, S. M. Taubenfeld, C. M. Alberini, J. C. Schaff, R. D. Slizter, I. I. Moraru, and R. Iyengar, *Cell shape and negative links in regulatory motifs together control spatial information flow in signaling networks*, Cell **133** (2008), no. 4, 666–680.

[18] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations*, J. Comput. Phys. **79** (1988), no. 1, 12–49. MR 89h:80012 Zbl 0659.65132

[19] O. Pertz, L. Hodgson, R. Klemke, and K. Hahn, *Spatiotemporal dynamics of RhoA activity in migrating cells*, Nature **440** (2006), no. 7087, 1069–1072.

[20] Y. Saad, *SPARSKIT: A basic tool-kit for sparse matrix computations* (version 2), 2005, user manual.

[21] Y. Saad and M. H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput. **7** (1986), 856–869. MR 87g:65064 Zbl 0599.65018

[22] J. C. Schaff, B. M. Slepchenko, Y. Choi, J. Wagner, D. Resasco, and L. Loew, *Analysis of nonlinear dynamics on arbitrary geometries with the Virtual Cell*, Chaos **11** (2001), no. 1, 115–131. Zbl 0992.92021

[23] I. C. Schneider, E. M. Parrish, and J. M. Haugh, *Spatial analysis of 3' phosphoinositide signaling in living fibroblasts, III: influence of cell morphology and morphological polarity*, Biophys. J. **89** (2005), no. 2, 1420–1430.

[24] P. Schwartz, D. Adalsteinsson, P. Colella, A. Arkin, and M. Onsum, *Numerical computation of diffusion on a surface*, P. Natl. Acad. Sci. USA **102** (2005), no. 32, 11151–11156.

[25] P. Schwartz, M. Barad, P. Colella, and T. Ligocki, *A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions*, J. Comput. Phys. **211** (2006), no. 2, 531–550. MR 2006e:65194 Zbl 1086.65532

[26] J. A. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, 2nd ed., Cambridge Univ. Press, Cambridge, 1999. MR 2000c:65015 Zbl 0973.76003

[27] J. R. Shewchuk, *Engineering a 2D quality mesh generator and Delaunay triangulator*, Applied computational geometry (M. C. Lin and D. Manocha, eds.), Lecture Notes in Computer Science, no. 1148, Springer, Berlin, 1996, pp. 203–222. MR 97k:68004

[28] Y. Shimada, M. P. Gulli, and M. Peter, *Nuclear sequestration of the exchange factor Cdc24 by Far1 regulates cell polarity during yeast mating*, Nat. Cell. Biol. **2** (2000), no. 2, 117–124.

[29] E. H. Twizell, A. B. Gumel, and M. A. Arigu, *Second-order, L_0-stable methods for the heat equation with time-dependent boundary conditions*, Adv. Comput. Math. **6** (1996), no. 3-4, 333–352 (1997). MR 97m:65164 Zbl 0872.65084

[30] J. Valdez-Taubas and H. R. B. Pelham, *Slow diffusion of proteins in the yeast plasma membrane allows polarity to be maintained by endocytic cycling*, Curr. Biol. **13** (2003), no. 18, 1636–1640.

[31] O. C. Zienkiewicz and R. L. Taylor, *The finite element method set*, 6th ed., Butterworth–Heinemann, 2005.

Received June 24, 2009. Revised December 4, 2009.

WANDA STRYCHALSKI: wandastr@email.unc.edu
Carolina Center for Interdisciplinary Applied Mathematics, Department of Mathematics, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, United States
http://www.unc.edu/∼wandastr

DAVID ADALSTEINSSON: david@amath.unc.edu
Carolina Center for Interdisciplinary Applied Mathematics, Department of Mathematics, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, United States
http://amath.unc.edu/David/David

TIMOTHY ELSTON: telston@med.unc.edu
Department of Pharmacology, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, United States
http://www.amath.unc.edu/Faculty/telston/

COMM. APP. MATH. AND COMP. SCI.
Vol. 5, No. 1, 2010

AN URN MODEL ASSOCIATED WITH JACOBI POLYNOMIALS

F. ALBERTO GRÜNBAUM

We consider an urn model leading to a random walk that can be solved explicitly in terms of the well-known Jacobi polynomials.

1. Urn models and orthogonal polynomials

There are two simple and classical models in statistical mechanics which have recently been associated with very important classes of orthogonal polynomials. The oldest one of these models is due to D. Bernoulli (1770) and S. Laplace (1810), while the more recent model is due to Paul and Tatiana Ehrenfest (1907) [6]. While both of these models are featured in very classical texts in probability theory, such as [7], the connection with orthogonal polynomials is of much more recent vintage. In fact, the polynomials in question due to Krawtchouk and Hahn had not been recognized as basic objects with rich properties until around 1950. For a few pertinent and useful references, see [1; 2; 4; 5; 11; 14; 16; 18].

From these comments one could get the impression that the relations between orthogonal polynomials — especially some well-known classes of them — are only a matter of historical interest. Nothing could be further from the truth: there are several areas of probability and mathematical physics where recent important progress hinges on the connections with orthogonal polynomials.

The entire area of random matrix theory starts with the work of E. Wigner and F. Dyson and reaches a new stage in the hands of M. Mehta who brought the power of orthogonal polynomials into the picture.

In the area of random growth processes, the seminal work of K. Johansson depends heavily on orthogonal polynomials, specifically Laguerre and Meixner ones; see [15].

The connection between birth-and-death processes and orthogonal polynomials has many parents, but the people that made the most of it are S. Karlin and J. McGregor [17]. We will have a chance to go back to their work in connection with our model here. The ideas of using the spectral analysis of the corresponding

MSC2000: primary 33C45, 60J10; secondary 60G99.

Keywords: random walks, urn models, Jacobi polynomials, orthogonal polynomials.

The author was supported in part by NSF Grant DMS-0603901 and AFOSR Contract FA9550-08-1-0169.

one-step transition matrix have been pushed recently in the case of quantum random walks, an area where physics, computer science and mathematics could make important contributions. See [3; 12].

The study of the so-called ASEP (asymmetric simple exclusion processes), going back to F. Spitzer [21] and very much connected with the work of K. Johansson mentioned earlier, has recently profited from connections with the Askey–Wilson polynomials. All of this has important and deep connections with combinatorics and a host of other areas of mathematics; for example, the study of nonintersecting or noncolliding random processes, which goes back to F. Dyson.

There are many interrelations among these areas. For one example: the Hahn polynomials that were mentioned in connection with the Bernoulli–Laplace model were studied by Karlin and McGregor [18] in connection with a model in genetics due to Moran. They have also been found to be useful in discussing random processes with nonintersecting paths [8].

All of these areas are places where orthogonal polynomials have been put to very good use. For a review of several of these items, see [19]. Orthogonal polynomials of several variables, as well as matrix-valued orthogonal polynomials, have recently been connected to certain random walks. For three examples, see [9; 10; 13].

2. The Jacobi polynomials

The classical Jacobi polynomials are usually considered either over the interval $[-1, 1]$ or, as we will do, over $[0, 1]$.

These polynomials are orthogonal with respect to the weight function

$$W(x) = x^{\alpha}(1-x)^{\beta}.$$

Here we assume that $\alpha, \beta > -1$; in fact it will be assumed throughout that α, β are nonnegative integers.

These polynomials are eigenfunctions of the differential operator

$$x(1-x)\frac{d^2}{dx^2} + \bigl(\alpha + 1 + x(\alpha+\beta+2)\bigr)\frac{d}{dx},$$

a fact that will not play any role in our discussion but which is crucial in most physical and geometrical applications of Jacobi polynomials. These applications cover a vast spectrum including potential theory, electromagnetism and quantum mechanics.

Neither the orthogonality, nor the fact that our polynomials are eigenfunctions of this differential operator are enough to determine them uniquely. One can multiply each polynomial by a constant and preserve these properties. We chose to normalize our polynomials by the condition

$$Q_n(1) = 1.$$

For us it will be important that these polynomials satisfy (and in fact be defined by) the three-term recursion relation

$$x\,Q_n(x) = A_n\,Q_{n+1}(x) + B_n\,Q_n(x) + C_n\,Q_{n-1}(x),$$

with $Q_0 = 1$ and $C_0 Q_{-1} = 0$.

The coefficients A_n, B_n, C_n are given by

$$A_n = \frac{(n+\beta+1)(n+\alpha+\beta+1)}{(2n+\alpha+\beta+1)(2n+\alpha+\beta+2)}, \qquad n \geq 0,$$

$$B_n = 1 + \frac{n(n+\beta)}{2n+\alpha+\beta} - \frac{(n+1)(n+\beta+1)}{2n+\alpha+\beta+2}, \quad n \geq 0,$$

$$C_n = \frac{n(n+\alpha)}{(2n+\alpha+\beta)(2n+\alpha+\beta+1)}, \qquad n \geq 1.$$

The coefficient B_n can be rewritten as

$$B_n = \frac{2n(n+\alpha+\beta+1) + (\alpha+1)\beta + \alpha(\alpha+1)}{(2n+\alpha+\beta)(2n+\alpha+\beta+2)},$$

which makes it clear that, along with the other coefficients, it is nonnegative.

Since we insist on the condition $Q_n(1) = 1$, we can see, for instance by induction and using the recursion relation, that

$$A_n + B_n + C_n = 1, \quad n \geq 1.$$

We also have

$$A_0 + B_0 = 1.$$

There are, of course, several explicit expressions for the different variants of the Jacobi polynomials, and they can be used, for instance, in computing the integrals that appear in the last section of this paper.

In terms of hypergeometric functions, the usual Jacobi polynomials are given by

$$\big((\alpha+1)_n/n!\big)\,{}_2F_1\big(-n, n+\alpha+\beta+1;\,(1-x)/2\big),$$

while our polynomials $Q_n(x)$ are obtained by multiplying these standard Jacobi polynomials by $(-1)^n n!/(\beta+1)_n$ and replacing x by $1-2x$.

The normalization chosen above is natural when one thinks of these polynomials (at least for some values of α, β) as the spherical functions for some appropriate symmetric space, and insists that these functions take the value 1 at the north pole of the corresponding sphere. The simplest of all cases is the one where $\alpha = \beta = 0$; one gets the Legendre polynomials and the usual two-dimensional sphere sitting in $\mathbb{R}^3$ (see [23]).

The fact that the coefficients are nonnegative and add up to one cries out for a probabilistic interpretation of these quantities. This is the purpose of this paper. We have not seen in the literature concrete models of random walks where the Jacobi polynomials play this role. The urn model we give is admittedly a bit contrived, but it is quite concrete. Hopefully it will motivate other people to find a more natural and simpler model that goes along with this recursion relation.

3. The model

Here we consider a discrete-time random walk on the nonnegative integers whose one-step transition probability matrix coincides with the one that gives the three-term recursion relation satisfied by the Jacobi polynomials.

At times $t = 0, 1, 2, \dots$, an urn contains n blue balls and this determines the state of our random walk on $\mathbb{Z} \geq 0$.

The urn sits in a "bath" consisting of an infinite number of red balls. The transition mechanism is made up of a few steps which are described now, leaving some of the details for later.

In the first step a certain number of red balls from the surrounding bath are mixed with the n blue balls in the urn.

In the second step a ball is selected (with uniform distribution) from among the balls in the urn. This "chosen ball" can be blue or red. In either case an experiment is performed in a parallel world, using an appropriate "auxiliary urn", to determine if this chosen ball will retain its color or have it changed (from red to blue or vice versa).

Once this is settled, and the possible change of color has taken place, the main urn contains the initial n balls plus a certain number of balls taken form the bath in the first step, and we are ready for the third and last step. This final step consists of having all red balls in the main urn removed and dropped back into the bath.

The state of the system at time $t+1$ is given by the number of blue balls in the urn after these three steps are completed. Clearly, the new state can take any of the values $n-1$, n, $n+1$.

A more detailed description of the three steps above is given in the next section.

4. The details of the model

If at time t the urn contains n blue balls, with $n = 0, 1, 2, \dots$, we pick

$$n + \alpha + \beta + 1$$

red balls from the bath to get a total of $2n+\alpha+\beta+1$ balls in the urn at the end of step one.

We now perform step 2: this gives us a blue ball with probability

$$\frac{n}{2n+\alpha+\beta+1}$$

and a red ball with probability

$$\frac{n+\alpha+\beta+1}{2n+\alpha+\beta+1}.$$

If the chosen ball is blue, then we throw α blue balls and $n+\beta$ red balls into an "auxiliary urn" with n blue balls, mix all these balls, and pick one with uniform distribution. We imagine the auxiliary urn surrounded by a bath of an infinite number of blue and red balls which are used to augment the n blue balls in this auxiliary urn.

The probability of getting a blue ball in the auxiliary urn is

$$\frac{n+\alpha}{2n+\alpha+\beta},$$

and if this is the outcome, the chosen ball in the main urn has its color changed from blue to red. If we get a red ball in this auxiliary urn, then the chosen ball retains its blue color.

On the other hand, if in step 2 we had chosen a red ball, then we throw $\alpha+1$ blue balls and $n+\beta+1$ red balls into a different auxiliary urn with n blue balls. This auxiliary urn is also surrounded by a bath of an infinite number of blue and red balls.

These balls are mixed and one is chosen with the uniform distribution. The probability that this ball in the auxiliary urn is red is given by

$$\frac{n+\beta+1}{2n+\alpha+\beta+2},$$

and if this is the case, the chosen ball in the main urn has its color changed from red to blue. Otherwise the chosen ball retains its red color.

Notice that the chosen ball in the main urn has a change of color only when we get a match of colors for the balls drawn in the main and an auxiliary urn: blue followed by blue or red followed by red.

In either case once the possible change of color of the chosen ball in the main urn has been decided upon, we remove all the red balls from the main urn.

We see that the state of the system goes from n to $n-1$ when the chosen ball is blue and then its color gets changed into red. This event has probability

$$\frac{n}{2n+\alpha+\beta+1} \times \frac{n+\alpha}{2n+\alpha+\beta}.$$

Observe that this coincides with the value of C_n in the recursion relation satisfied by our version of the Jacobi polynomials.

The state increases from n to $n+1$ if the chosen ball is red and its color gets changed into blue. This event has probability

$$\frac{n+\alpha+\beta+1}{2n+\alpha+\beta+1} \times \frac{n+\beta+1}{2n+\alpha+\beta+2}.$$

This coincides with the values of A_n given earlier.

As we noticed earlier, when the chosen ball is blue and the ball in the corresponding auxiliary urn is red then the chosen ball retains its color. Likewise if the chosen ball is red and the ball in the corresponding auxiliary urn is blue then the chosen one retains its color. In either case, the total number of blue balls in the main urn remains unchanged and the state goes from n to n.

Recall the basic property of the coefficients A_n, B_n, C_n, namely

$$A_n + B_n + C_n = 1.$$

This shows that the probability of going from state n to state n is given by B_n.

In summary, we have built a random walk whose one-step transition probability is the one given by the three-term relation satisfied by our version of the Jacobi polynomials.

5. Birth-and-death processes and orthogonal polynomials

A Markov chain with state space given by the nonnegative integers and a tridiagonal one-step transition probability matrix $\mathbb{P}$ is called a birth-and-death process. Our model given above clearly fits in this framework.

One of the most important connections between orthogonal polynomials and birth-and-death processes, such as the one considered here, is given by the Karlin–McGregor formula [17].

If the polynomials satisfy

$$\pi_j \int_0^1 Q_i(x) Q_j(x) W(x) dx = \delta_{ij},$$

one gets the following representation formula for the entries of the powers of the one-step transition probability matrix

$$(\mathbb{P}^n)_{ij} = \pi_j \int_0^1 x^n Q_i(x) Q_j(x) W(x) dx.$$

This compact expression gives the solution to the dynamics of our random walk and allows for the study of many of its properties.

In the case of our version of the Jacobi polynomials, the squares of the norms of the polynomials Q_i are given by

$$\frac{\Gamma(i+1)\Gamma(i+\alpha+1)\Gamma(\beta+1)^2}{\Gamma(i+\beta+1)\Gamma(i+\alpha+\beta+1)(2i+\alpha+\beta+1)}.$$

In our case, when α, β are assumed to be nonnegative integers, this expression can of course be written without any reference to the Gamma function.

We recall how one can compute in the case of our transition matrix P its invariant (stationary) distribution, that is, the (unique up to scalars) row vector

$$\boldsymbol{\pi} = (\pi_0, \pi_1, \pi_2, \dots)$$

such that

$$\boldsymbol{\pi} P = \boldsymbol{\pi}.$$

It is a simple matter of using the recursion relation for the polynomials Q_i to show that the components π_i are given, up to a common multiplicative constant, by the inverses of the integrals

$$\int_0^1 Q_i^2(x) W(x) dx$$

mentioned above. This justifies the notation π_i for these two apparently unrelated quantities, and in our case furnishes an explicit expression for an invariant distribution.

We close this paper with a note of historical interest. One of the referees suggested that I contact Dick Askey, who reportedly had pointed out that the Legendre polynomials had surfaced for the first time in connection with a problem in probability theory.

Askey recalls that Arthur Erdélyi told him once that this occurred in a work by J. L. Lagrange. Indeed in [20] , Lagrange considers such a problem. In the process of solving it he needs to find the power series expansion of the expression

$$\frac{1}{\sqrt{1-2az+(a^2-4b^2)z^2}}$$

in powers of z.

The three-term recursion for the coefficients in this expansion is explicitly written down and considered well-known by Lagrange. The account of Lagrange's work given in the very complete and scholarly book [22] has a derivation of this recursion. The case $a^2 - 4b^2 = 1$ gives the Legendre polynomials in the variable a. The work of Lagrange took place in the period 1770–1773, and predates the work of Legendre and Laplace. I am thankful to the anonymous referee and to Askey for pointing me in the correct direction.

References

[1] G. E. Andrews, R. Askey, and R. Roy, *Special functions*, Encyclopedia of Mathematics and its Applications, no. 71, Cambridge University Press, 1999. MR 2000g:33001 Zbl 0920.33001

[2] R. Askey, *Evaluation of Sylvester type determinants using orthogonal polynomials*, Advances in analysis (H. G. W. Begehr, R. P. Gilbert, M. E. Muldoon, and M. W. Wong, eds.), World Sci. Publ., Hackensack, NJ, 2005, pp. 1–16. MR 2006i:15013 Zbl 1090.15007

[3] M. J. Cantero, F. A. Grünbaum, L. Moral, and L. Velazquez, *Matrix valued Szegő polynomials and quantum random walks*, Comm. Pure Appl. Math. (2009). arXiv 0901.2244

[4] T. S. Chihara, *An introduction to orthogonal polynomials*, Mathematics and its Applications, no. 13, Gordon and Breach, New York, 1978. MR 58 #1979 Zbl 0389.33008

[5] P. Diaconis and M. Shahshahani, *Time to reach stationarity in the Bernoulli–Laplace diffusion model*, SIAM J. Math. Anal. **18** (1987), no. 1, 208–218. MR 88e:60014 Zbl 0617.60009

[6] P. Ehrenfest and T. Eherenfest, *über zwei bekannte Einwände gegen das Boltzmannsche H-Theorem*, Phys. Z. **8** (1907), 311–314. JFM 38.0931.01

[7] W. Feller, *An introduction to probability theory and its applications*, 3rd ed., vol. I, Wiley, New York, 1968. MR 37 #3604 Zbl 0155.23101

[8] V. E. Gorin, *Nonintersecting paths and the Hahn orthogonal polynomial ensemble*, Funktsional. Anal. i Prilozhen. **42** (2008), no. 3, 23–44, 96. MR 2010a:60027 Zbl 05519876

[9] F. A. Grünbaum, *The Karlin–McGregor formula for a variant of a discrete version of walsh's spider*, J. Phys. **45**, art. 454010. Zbl 05638242

[10] F. A. Grünbaum, *The Rahman polynomials are bispectral*, Symmetry, Integrability Geom. Methods Appl. **3** (2007), Paper 065, 11. MR 2009m:33019 Zbl 05241578

[11] ———, *Random walks and orthogonal polynomials: some challenges*, Probability, geometry and integrable systems (M. Pinsky and B. Birnir, eds.), Math. Sci. Res. Inst. Publ., no. 55, Cambridge Univ. Press, 2008, pp. 241–260. MR 2009i:33011

[12] M. Hamada, N. Konno, and W. Mlotkowski, *Orthogonal polynomials induced by discrete-time quantum walks in one dimension*, preprint, 2009. arXiv 0903.4047

[13] M. R. Hoare and M. Rahman, *A probabilistic origin for a new class of bivariate polynomials*, Symmetry Integrability Geom. Methods Appl. **4** (2008), Paper 089, 18 pp. MR 2010a:33028

[14] M. E. H. Ismail, D. R. Masson, J. Letessier, and G. Valent, *Birth and death processes and orthogonal polynomials*, Orthogonal polynomials (P. Nevai, ed.), NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., no. 294, Kluwer, Dordrecht, 1990, pp. 229–255. MR 93c:42021 Zbl 0704.60084

[15] K. Johansson, *Shape fluctuations and random matrices*, Comm. Math. Phys. **209** (2000), no. 2, 437–476. MR 2001h:60177 Zbl 0969.15008

[16] M. Kac, *Random walk and the theory of Brownian motion*, Amer. Math. Monthly **54** (1947), 369–391. MR 9,46c Zbl 0031.22604

[17] S. Karlin and J. McGregor, *Random walks*, Illinois J. Math. **3** (1959), 66–81. MR 20 #7352 Zbl 0104.11804

[18] S. Karlin and J. L. McGregor, *The Hahn polynomials, formulas and an application*, Scripta Math. **26** (1961), 33–46. MR 25 #2249 Zbl 0104.29103

[19] W. König, *Orthogonal polynomial ensembles in probability theory*, Probab. Surv. **2** (2005), 385–447. MR 2007e:60007

[20] J. L. Lagrange, *Mémoire sur l'utilité de la méthode de prendre le milieu entre les résultats de plusieurs observations*, Miscellanea Taurinensia **5**, Reprinted in his Œuvres, volume 2, pages 173–185, Min. de l'Instr. Pub. Paris.

[21] F. Spitzer, *Interaction of Markov processes*, Advances in Math. **5** (1970), 246–290 (1970). MR 42 #3856 Zbl 0312.60060

[22] I. Todhunter, *A history of the mathematical theory of probability*, MacMillan, 1865.

[23] N. J. Vilenkin and A. U. Klimyk, *Representation of Lie groups and special functions, Vol 3: Classical and quantum groups and special functions*, Mathematics and its Applications (Soviet Series), no. 75, Kluwer, Dordrecht, 1992. MR 96f:33043 Zbl 0778.22001

Received August 26, 2009. Revised December 13, 2009.

F. ALBERTO GRÜNBAUM: grunbaum@math.berkeley.edu
Department of Mathematics, University of California, Berkeley, CA 94720, United States

COMM. APP. MATH. AND COMP. SCI.
Vol. 5, No. 1, 2010

ENSEMBLE SAMPLERS WITH AFFINE INVARIANCE

JONATHAN GOODMAN AND JONATHAN WEARE

We propose a family of Markov chain Monte Carlo methods whose performance is unaffected by affine tranformations of space. These algorithms are easy to construct and require little or no additional computational overhead. They should be particularly useful for sampling badly scaled distributions. Computational tests show that the affine invariant methods can be significantly faster than standard MCMC methods on highly skewed distributions.

1. Introduction

Markov chain Monte Carlo (MCMC) sampling methods typically have parameters that need to be adjusted for a specific problem of interest [9; 10; 1]. For example, a trial step-size that works well for a probability density $\pi(x)$, with $x \in \mathbb{R}^n$, may work poorly for the scaled density

$$\pi_\lambda(x) \;=\; \lambda^{-n}\,\pi\,(\lambda x)\,, \tag{1}$$

if λ is very large or very small. Christen [2] has recently suggested a simple method whose performance sampling the density π_λ is independent of the value of λ. Inspired by this idea we suggest a family of many particle (ensemble) MCMC samplers with the more general *affine invariance* property. Affine invariance implies that the performance of our method is independent of the aspect ratio in highly anisotropic distributions such as the one depicted in Figure 1.

An affine transformation is an invertible mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$ of the form $y = Ax + b$. If X has probability density $\pi(x)$, then $Y = AX + b$ has density

$$\pi_{A,b}(y) \;=\; \pi_{A,b}(Ax+b) \;\propto\; \pi(x). \tag{2}$$

Consider, for example, the skewed probability density on $\mathbb{R}^2$ pictured in Figure 1:

$$\pi(x) \;\propto\; \exp\Bigl(\frac{-(x_1-x_2)^2}{2\epsilon} \;-\; \frac{(x_1+x_2)^2}{2}\Bigr). \tag{3}$$

MSC2000: 65C05.

Keywords: Markov chain Monte Carlo, affine invariance, ensemble samplers.

Both authors were supported in part by the Applied Mathematical Sciences Program of the U.S. Department of Energy under contract DEFG0200ER25053.

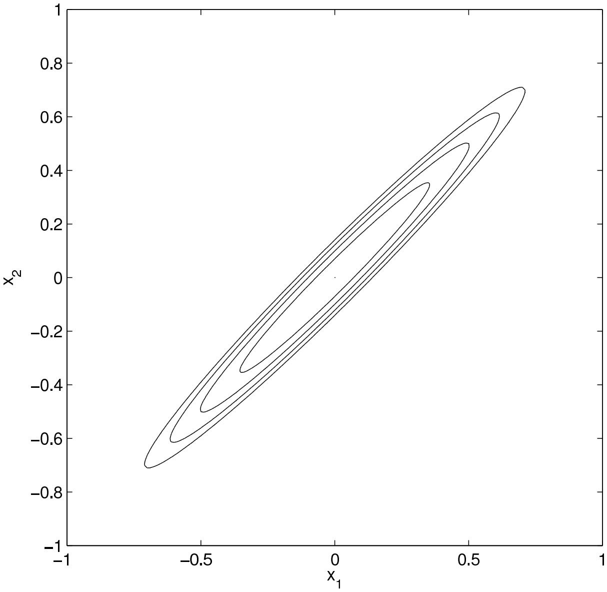


Figure 1. Contours of the Gaussian density defined in expression (3).

Single-variable MCMC strategies such as Metropolis or heat bath (Gibbs sampler) [13; 10] would be forced to make perturbations of order $\sqrt{\epsilon}$ and would have slow equilibration. A better MCMC sampler would use perturbations of order $\sqrt{\epsilon}$ in the $(1, -1)$ direction and perturbations of order one in the $(1, 1)$ direction.

On the other hand, the affine transformation

$$y_1 = \frac{x_1 - x_2}{\sqrt{\epsilon}}, \qquad y_2 = x_1 + x_2,$$

turns the challenging sampling problem (3) into the easier problem

$$\pi_A(y) \propto e^{-(y_1^2 + y_2^2)/2}. \tag{4}$$

Sampling the well scaled transformed density (4) does not require detailed customization. An affine invariant sampler views the two densities as equally difficult. In particular, the performance of an affine invariant scheme on the skewed density (3) is independent of ϵ. More generally, if an affine invariant sampler is applied to a nondegenerate multivariate normal $\pi(x) \propto e^{-x^t Hx/2}$, the performance is independent of H.

We consider general MCMC samplers of the form $X(t+1) = R(X(t), \xi(t), \pi)$, where $X(t)$ is the sample after t iterations, $\xi(t)$ is a sequence of iid (independent identically distributed) random variables[1], and π is a probability density. General

[1] The probability space for ξ is not important. A Monte Carlo code would typically take $\xi(t)$ to be an infinite sequence of independent uniform [0, 1] random variables.

purpose samplers such as Gibbs samplers have this form. We call such an MCMC algorithm *affine invariant* if, for any affine transformation $Ax + b$,

$$R(Ax + b, \xi(t), \pi_{A,b}) = AR(x(t), \xi(t), \pi) + b,$$

for every x and almost all $\xi(t)$.

Less formally, suppose we make two Monte Carlo runs using the same random number generator and seed so that the $\xi(t)$ will be identical for both runs. Suppose one of the runs uses probability density π and starting point $X(0)$. Suppose the other uses $\pi_{A,b}$ and initial point $Y(0) = AX(0) + b$. If the algorithm is affine invariant, the sequences will satisfy $Y(t) = AX(t) + b$. We are not aware of a practical sampler that has this affine invariance property for any general class of densities.

In this paper we propose a family of affine invariant *ensemble* samplers. An ensemble, $\vec{X}$, consists of L *walkers*[2] $X_k \in \mathbb{R}^n$. Since each walker is in $\mathbb{R}^n$, we may think of the ensemble $\vec{X} = (X_1, \dots, X_L)$ as being in $\mathbb{R}^{nL}$. The target probability density for the ensemble is the one in which the walkers are independent and drawn from π, that is,

$$\Pi(\vec{x}) = \Pi(x_1, \dots, x_L) = \pi(x_1)\,\pi(x_2)\cdots\pi(x_L). \tag{5}$$

An ensemble MCMC algorithm is a Markov chain on the state space of ensembles. Starting with $\vec{X}(1)$, it produces a sequence $\vec{X}(t)$. The ensemble Markov chain can preserve the product density (5) without the individual walker sequences $X_k(t)$ (as functions of t) being independent, or even being Markov. This is because the distribution of $X_k(t + 1)$ can depend on $X_j(t)$ for $j \neq k$.

We apply an affine transformation to an ensemble by applying it separately to each walker:

$$\vec{X} = (X_1, \dots, X_L) \xrightarrow{A,b} (AX_1 + b, \dots, AX_L + b) = (Y_1, \dots, Y_L) = \vec{Y}. \tag{6}$$

Suppose that $\vec{X}(1) \xrightarrow{A,b} \vec{Y}(1)$ and that $\vec{Y}(t)$ is the sequence produced using $\pi_{A,b}$ in place of π in (5) and the same initial random number generator seed. The ensemble MCMC method is affine invariant if $\vec{X}(t) \xrightarrow{A,b} \vec{Y}(t)$. We will describe the details of the algorithms in Section 2.

Our ensemble methods are motivated in part by the Nelder–Mead [11] simplex algorithm for solving deterministic optimization problems. Many in the optimization community attribute its robust convergence to the fact that it is affine invariant. Applying the Nelder–Mead algorithm to the ill conditioned optimization problem for

[2] Here x_k is walker k in an ensemble of L walkers. This is inconsistent with (3) and (4), where x_1 was the first component of $x \in \mathbb{R}^2$.

the function (3) in Figure 1 is exactly equivalent to applying it to the easier problem of optimizing the well scaled function (4). This is not the case for noninvariant methods such as gradient descent [6].

The Nelder–Mead simplex optimization scheme evolves many copies of the system toward a local minimum (in our terminology: many walkers in an ensemble). A new position for any one copy is suggested by an affine invariant transformation which is constructed using the current positions of the other copies of the system. Similarly, our Monte Carlo method moves one walker using a proposal generated with the help of other walkers in the ensemble. The details of the construction of our ensemble MCMC schemes are given in the next section.

An additional illustration of the power of affine invariance was pointed out to us by our colleague Jeff Cheeger. Suppose we wish to sample X uniformly in a convex body, K (a bounded convex set with nonempty interior). A theorem of Fritz John [8] states that there is a number r depending only on the dimension such that for any convex body K there is an affine transformation $Ax+B$ that makes $\widetilde{K} = AK + b$ well conditioned in the sense that $B_1 \subseteq \widetilde{K}$ and $\widetilde{K} \subseteq B_r$, where B_r is the ball of radius r centered at the origin. An affine invariant sampling method should, therefore, be uniformly effective over all the convex bodies of a given dimension regardless of their shape.

After a discussion of the integrated autocorrelation time as a means of comparing our ensemble methods with single-particle methods in Section 3 we present the results of several numerical tests in Section 4. The first of our test distributions is a difficult two-dimensional problem that illustrates the advantages and disadvantages of our scheme. In the second example we use our schemes to sample from a 101-dimensional approximation to the invariant measure of stochastic partial differential equation. In both cases the affine invariant methods significantly outperform the single site Metropolis scheme. Finally, in Section 5 we give a very brief discussion of the method used to compute the integrated autocorrelation times of the algorithms.

2. Construction

As mentioned in the introduction, our ensemble Markov chain is evolved by moving one walker at time. We consider one step of the ensemble Markov chain $\vec{X}(t) \to \vec{X}(t+1)$ to consist of one cycle through all L walkers in the ensemble. This is expressed in pseudo-code as

```
for $k = 1, \ldots, L$
 {
   update:   $X_k(t) \to X_k(t+1)$
 }
```

Each walker X_k is updated using the current positions of all of the other walkers in the ensemble. The other walkers (besides X_k) form the *complementary ensemble*

$$\vec{X}_{[k]}(t) = \{X_1(t+1), \ldots, X_{k-1}(t+1), X_{k+1}(t), \ldots, X_L(t)\}\,.$$

Let $\mu(d\tilde{x}_k, x_k \mid \vec{x}_{[k]})$ be the transition kernel for moving walker X_k. The notation means that for each $x_k \in \mathbb{R}^n$ and $\vec{x}_{[k]} \in \mathbb{R}^{(L-1)n}$, the measure $\mu(\cdot, x_k \mid \vec{x}_{[k]})$ is the probability measure for $X_k(t+1)$, if $X_k(t) = x_k$ and $\vec{X}_{[k]}(t) = \vec{x}_{[k]}$.

Our single walker moves are based on *partial resampling* [13; 10]. This states that the transformation $\vec{X}(t) \to \vec{X}(t+1)$ preserves the joint distribution Π if the single walker moves $X_k(t) \to X_k(t+1)$ preserve the conditional distribution of x_k given $X_{[k]}$. For our Π (which makes walkers independent), this is the same as saying that $\mu(\cdot, \cdot \mid \vec{x}_{[k]})$ preserves π for all $\vec{x}_{[k]}$, or (somewhat informally)

$$\pi(\tilde{x}_k)\, d\tilde{x}_k \;=\; \int_{\mathbb{R}^n} \mu(d\tilde{x}_k, x_k \mid \vec{x}_{[k]})\pi(x_k)\, dx_k.$$

As usual, this condition is achieved using detailed balance. We use a trial distribution to propose a new value of X_k and then accept or reject this move using the appropriate Metropolis Hastings rule [13; 10]. Our motivation is that the distribution of the walkers in the complementary ensemble carries useful information about the density π. This gives an automatic way to adapt the trial move to the target density. Christen [2] uses an ensemble of 2 walkers to generate scale invariant trial moves using the relative positions of the walkers.

The simplest (and best on the Rosenbrock test problem in Section 4) move of this kind that we have found is the *stretch move*. In a stretch move, we move walker X_k using one complementary walker $X_j \in \vec{X}_{[k]}(t)$ (that is, $j \neq k$). The move consists of a proposal of the form (see Figure 2)

$$X_k(t) \to Y = X_j \,+\, Z(X_k(t) - X_j). \tag{7}$$

The stretch move defined in expression (7) is similar to what is referred to as the "walk move" in [2] though the stretch move is affine invariant while the walk move of [2] is not. As pointed out in [2], if the density g of the scaling variable Z satisfies the symmetry condition

$$g\left(\frac{1}{z}\right) = z\, g(z), \tag{8}$$

then the move (7) is symmetric in the sense that (in the usual informal way Metropolis is discussed)

$$\Pr\left(X_k(t) \to Y\right) = \Pr\left(Y \to X_k(t)\right).$$

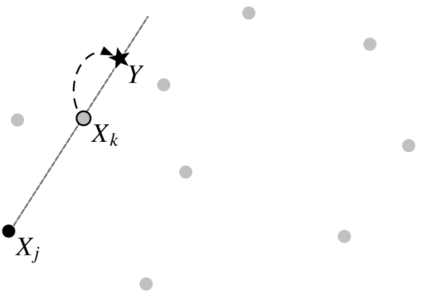


Figure 2. A stretch move. The light dots represent the walkers not participating in this move. The proposal is generated by stretching along the straight line connecting X_j to X_k.

The particular distribution we use is the one suggested in [2]:

$$g(z) \;\propto\; \begin{cases} \dfrac{1}{\sqrt{z}} & \text{if } z \in \left[\dfrac{1}{a}, a\right], \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

where the parameter $a > 1$ can be adjusted to improve performance.

To find the appropriate acceptance probability for this move we again appeal to partial resampling. Notice that the proposal value Y lies on the ray

$$\left\{ y \in \mathbb{R}^n : \; y - X_j = \lambda\, (X_k(t) - X_j),\;\; \lambda > 0 \right\}.$$

The conditional density of π along this ray is proportional to

$$\| y - X_j \|^{n-1}\, \pi(y).$$

Since the proposal in (7) is symmetric, partial resampling then implies that if we accept the move $X_k(t+1) = Y$ with probability

$$\min\left\{ 1,\; \frac{\|Y - X_j\|^{n-1}\, \pi(Y)}{\|X_k(t) - X_j\|^{n-1}\, \pi(X_k(t))} \right\} = \min\left\{ 1,\; Z^{n-1}\, \frac{\pi(Y)}{\pi(X_k(t))} \right\},$$

and set $X_k(t+1) = X_k(t)$ otherwise, the resulting Markov chain satisfies detailed balance.

The stretch move, and the walk and replacement moves below, define irreducible Markov chains on the space of *general* ensembles. An ensemble is general if there is no lower-dimensional hyperplane (dim $< n$) that contains all the walkers in the ensemble. The space of general ensembles is $\mathcal{G} \subset \mathbb{R}^{nL}$. For $L \geq n+1$, a condition we always assume, almost every ensemble (with respect to Π) is general. Therefore, sampling Π restricted to $\mathcal{G}$ is (almost) the same as sampling Π on all of $\mathbb{R}^{nL}$. It is clear that if $\vec{X}(1) \in \mathcal{G}$, then almost surely $\vec{X}(t) \in \mathcal{G}$ for $t = 2, 3, \ldots$. We assume that $\vec{X}(1)$ is general. It is clear that any general ensemble can be transformed to any other general ensemble by a finite sequence of stretch moves.

The operation $\vec{X}(t) \to \vec{X}(t+1)$ using one stretch move per walker is given by

```
for $k = 1, \dots, L$
 {
   choose $X_j \in \vec{X}_{[k]}(t)$ at random
   generate $Y = X_j + Z(X_k(t) - X_j)$, all $Z$ choices independent
   accept, set $X_k(t+1) = Y$, with probability (7)
   otherwise reject, set $X_k(t+1) = X_k(t)$
 }
```

We offer two alternative affine invariant methods. The first, which we call the *walk* move, is illustrated in Figure 3. A walk move begins by choosing a subset S of the walkers in $\vec{X}_{[k]}(t)$. It is necessary that $|S| \geq 2$, and that the choice of S is independent of $X_k(t)$. The walk move offers a proposal $X_k \to X_k + W$, where W is normal with mean zero and the same covariance as the walkers $X_j \in S$.

More formally, let

$$\pi_S(x) = (1/\,|S|) \sum_{X_j \in S} \delta(x - X_j)$$

be the empirical distribution of the walkers in S. Given S, the mean of a random variable $X \sim \pi_S$ is

$$\overline{X}_S = \frac{1}{|S|} \sum_{X_j \in S} X_j .$$

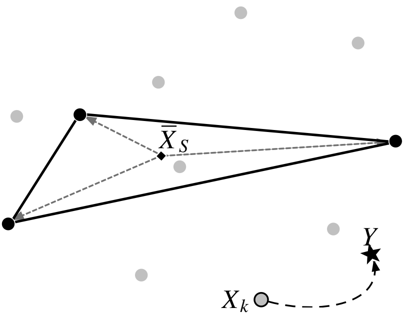


Figure 3. A walk move. The dots represent the ensemble of particles. The dark ones represent the walkers in $\vec{X}_S$. The diamond inside the triangle represents the sample mean $\overline{X}_S$. The proposed perturbation has covariance equal to the sample covariance of the three dark dots. The perturbation is generated by summing random multiples of the arrows from $\overline{X}_S$ to the vertices of the triangle.

The covariance is

$$C_S = \frac{1}{|S|} \sum_{X_j \in S} (X_j - \overline{X}_S)(X_j - \overline{X}_S)^t. \tag{10}$$

It is easy to check that if the Z_j are univariate standard normals then, conditioned on S,

$$W = \sum_{X_j \in S} Z_j (X_j - \overline{X}_S) \tag{11}$$

is normal with mean zero and covariance (10). The proposed trial move is

$$X_k(t) \to X_k(t) + W.$$

The random variable (11) is symmetric in that

$$\Pr(X \to X + W = Y) = \Pr(Y \to Y - W = X).$$

Therefore, we insure detailed balance by accepting the move $X_k(t) \to X_k(t) + W$ with the Metropolis acceptance probability

$$\min\left\{ 1,\ \frac{\pi\ ((X_k(t) + W)}{\pi\ ((X_k(t))} \right\}.$$

The walk move ensemble Monte Carlo method just described clearly is affine invariant in the sense discussed above. In the invariant density $\Pi(\vec{x})$ given by (5), the covariance matrix for W satisfies (an easy check)

$$\operatorname{cov}[W] \propto \operatorname{cov}_\pi[X].$$

The constant of proportionality depends on σ^2 and $|S|$. If π is highly skewed in the fashion of Figure 1, then the distribution of the proposed moves will have the same skewness.

Finally, we propose a variant of the walk move called the *replacement move*. Suppose $\pi_S(x \mid S)$ is an estimate of $\pi(x)$ using the subensemble $S \subset X_{[k]}(t)$. A replacement move seeks to replace $X_k(t)$ with an independent sample from $\pi_S(x \mid S)$. The probability of an $x \to y$ proposal is $\pi(x)\pi_S(y \mid S)$, and the probability of a $y \to x$ proposal is $\pi(y)\pi_S(x \mid S)$. It is crucial here, as always, that S is the same in both expressions. If $P_{x\to y}$ is the probability of accepting an $x \to y$ proposal, detailed balance is the formula

$$\pi(x)\pi_S(y \mid S) P_{x\to y} = \pi(y)\pi_S(x \mid S) P_{y\to x}.$$

The usual reasoning suggests that we accept an $x \to y$ proposal with probability

$$P_{x\to y} = \min\left\{ 1,\ \frac{\pi(y)}{\pi_S(y \mid S)} \cdot \frac{\pi_S(x \mid S)}{\pi(x)} \right\}. \tag{12}$$

In the case of a Gaussian π, one can easily modify the proposal used in the walk move to define a density $\pi_S(x \mid S)$ that is an accurate approximation to π if L and $|S|$ are large. This is harder if π is not Gaussian. We have not done computational tests of this method yet.

3. Evaluating ensemble sampling methods

We need criteria that will allow us to compare the ensemble methods above to standard *single-particle* methods. Most Monte Carlo is done for the purpose of estimating the expected value of something:

$$A = E_\pi \left[f(X) \right] = \int_{\mathbb{R}^n} f(x)\pi(x)\, dx, \tag{13}$$

where π is the target density and f is some function of interest.[3] Suppose $X(t)$, for $t = 1, 2, \dots, T_s$, are the successive states of a single-particle MCMC sampler for π. The standard single-particle MCMC estimator for A is

$$\widehat{A}_s = \frac{1}{T_s} \sum_{t=1}^{T_s} f(X(t)). \tag{14}$$

An ensemble method generates a random path of the ensemble Markov chain $\vec{X}(t) = (X_1(t), \dots, X_L(t))$ with invariant distribution Π given by (5). Let T_e be the length of the ensemble chain. The natural ensemble estimator for A is

$$\widehat{A}_e = \frac{1}{T_e} \sum_{t=1}^{T_e} \left(\frac{1}{L} \sum_{k=1}^{L} f(X_k(t)) \right). \tag{15}$$

When $T_s = LT_e$, the two methods do about the same amount of work, depending on the complexity of the individual samplers.

For practical Monte Carlo, the accuracy of an estimator is given by the asymptotic behavior of its variance in the limit of long chains [13; 10]. For large T_s we have

$$\operatorname{var}\left[\widehat{A}_s\right] \approx \frac{\operatorname{var}_\pi \left[f(X) \right]}{T_s / \tau_s}, \tag{16}$$

where τ_s is the *integrated autocorrelation time* given by

$$\tau_s = \sum_{t=-\infty}^{\infty} \frac{C_s(t)}{C_s(0)}, \tag{17}$$

[3] Kalos and Whitlock [9] make a persuasive case for making this the definition: *Monte Carlo* means using random numbers to estimate some number that itself is not random. Generating random samples for their own sakes is *simulation*.

and the lag t *autocovariance* function is

$$C_s(t) = \lim_{t' \to \infty} \mathrm{cov}\big[f(X(t'+t)) \, , \, f(X(t')) \big]. \tag{18}$$

We estimate τ_s from the time series $f(X(t))$ using a shareware procedure called `Acor` that uses a variant (described below) of the self consistent window method of [7].

Define the ensemble average as $F(\vec{x}) = \frac{1}{L} \sum_{k=1}^{L} f(x_k)$. Then (15) is

$$\widehat{A}_e = \frac{1}{T_e} \sum_{t=1}^{T_e} F(\vec{X}(t)).$$

The analogous definitions of the autocovariance and integrated autocorrelation time for the ensemble MCMC method are

$$\tau_e = \sum_{t=-\infty}^{\infty} \frac{C_e(t)}{C_e(0)},$$

with

$$C_e(t) = \lim_{t' \to \infty} \mathrm{cov}\big[F(\vec{X}(t'+t)) \, , \, F(\vec{X}(t')) \big].$$

Given the obvious relation (Π in (5) makes the walkers in the ensemble independent)

$$\mathrm{var}_\Pi\,[F(\vec{X})] = \frac{1}{L} \mathrm{var}_\pi\,[f(X)],$$

the ensemble analogue of (16) is

$$\mathrm{var}[\widehat{A}_e] \approx \frac{\mathrm{var}_\pi\,[f(X)]}{LT_e/\tau_e} \,.$$

The conclusion of this discussion is that, in our view, a sensible way to compare single-particle and ensemble Monte Carlo is to compare τ_s to τ_e. This compares the variance of two estimators that use a similar amount of work. Comparing variances is preferred to other possibilities such as comparing the mixing times of the two chains [4] for two reasons. First, the autocorrelation time may be estimated directly from Monte Carlo data. It seems to be a serious challenge to measure other mixing rates from Monte Carlo data (see, however, [5] for estimating the spectral gap). Second, the autocorrelation time, not the mixing rate, determines the large time error of the Monte Carlo estimator. Practical Monte Carlo calculations that are not in this large time regime have no accuracy.

Of course, we could take as our ensemble method one in which each $X_k(t)$ is an independent copy of a single Markov chain sampling π. The reader can easily convince herself or himself that in this case $\tau_e = \tau_s$ exactly. Thus such an ensemble method with $T_e = LT_s$ would have exactly the same large time variance as the single-particle method. Furthermore with $T_e = LT_s$ the two chains would require exactly the same computation effort to generate. The two methods would therefore be indistinguishable in the long time limit.

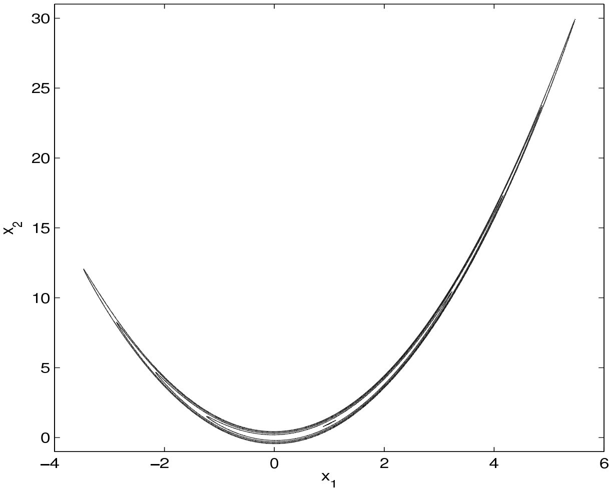
4. Computational tests

In this section we present and discuss the results of computational experiments to determine the effectiveness of our ensemble methods relative to a standard single-particle Markov chain Monte Carlo method. The MCMC method that we choose for comparison is the single site Metropolis scheme in which one cycles through the coordinates of $X(t)$ perturbing a single coordinate at a time and accepting or rejecting that perturbation with the appropriate Metropolis acceptance probability before moving on to the next coordinate. For the perturbations in the Metropolis scheme we choose Gaussian random variables. All user defined parameters are chosen (by trial and error) to optimize performance (in terms of the integrated autocorrelation times). In all cases this results in an acceptance rate close to 30%. For the purpose of discussion, we first present results from tests on a difficult two-dimensional example. The second example is a 101-dimensional, badly scaled distribution that highlights the advantages of our scheme.

4.1. ***The Rosenbrock density.*** In this subsection we present numerical tests on the Rosenbrock density, which is given by[4]

$$\pi(x_1, x_2) \propto \exp\Big(-\frac{100(x_2 - {x_1}^2)^2 + (1 - x_1)^2}{20}\Big). \tag{19}$$

Here are some contours of the Rosenbrock density:



[4] To avoid confusion with earlier notation, in the rest of this section (x_1, x_2) represents an arbitrary point in $\mathbb{R}^2$.

ensemble	$f(x_1,x_2)=x_1$				$f(x_1,x_2)=x_2$			
method↓ size →	1	10	100	∞	1	10	100	∞
Metropolis	163	–	–	–	322	–	–	–
stretch moves	–	19.4	8.06	8.71	–	67.0	18.4	23.5
walk moves, $\lvert S\rvert=3$	–	46.4	19.8	18.6	–	68.0	44.2	47.1

Table 1. Autocorrelation times (multiplied by 10^{-3}) with the functionals $f(x_1,x_2)=x_1$ and $f(x_1,x_2)=x_2$ for single-particle isotropic Metropolis and the chains generated by the two ensemble methods. The ensemble methods with ensemble size $L=\infty$ generate complementary walkers by exact sampling of the Rosenbrock density. The per-step cost of the methods are roughly equivalent on this problem.

Though only two-dimensional, this is a difficult density to sample efficiently as it exhibits the scaling and degeneracy issues that we have discussed throughout the paper. Further the Rosenbrock density has the feature that there is not a single affine transformation that can remove these problems. Thus in some sense this density is designed to cause difficulties for our affine invariant estimators. Of course its degeneracy will cause problems for the single-particle estimator and we will see that the affine invariant schemes are still superior.

Table 1 presents results for the functionals $f(x_1,x_2)=x_1$ and $f(x_1,x_2)=x_2$. The values given should be multiplied by 1000 because we subsampled every Markov chain by 1000. In both cases, the best ensemble sampler has an autocorrelation time about ten times smaller than that of isotropic Metropolis. The walk move method with $|S|=3$ has autocorrelation times a little more than twice as long as the stretch move method. All estimates come from runs of length $T_s=10^{11}$ and $T_e=T_s/L$. In all cases we estimate the autocorrelation time using the `Acor` procedure.

To simulate the effect of $L=\infty$ (infinite ensemble size), we generate the complementary X_j used to move X_k by independent sampling of the Rosenbrock density (19). For a single step, this is exactly the same as the finite L ensemble method. The difference comes in possible correlations between steps. With finite L, it is possible that at time $t=1$ we take $j=4$ for $k=5$ (that is, use $X_4(1)$ to help move $X_5(1)$), and then use $j=4$ for $k=5$ again at the next time $t=2$. Presumably, possibilities like this become unimportant as $L\to\infty$. We sample the Rosenbrock density using the fact that the marginal of X is Gaussian, and the conditional density of Y given X also is Gaussian.

Finally, we offer a tentative explanation of the fact that stretch moves are better than walk moves for the Rosenbrock function. The walk step, W, is chosen using three points as in Figure 3; see (11). If the three points are close to X_k, the covariance

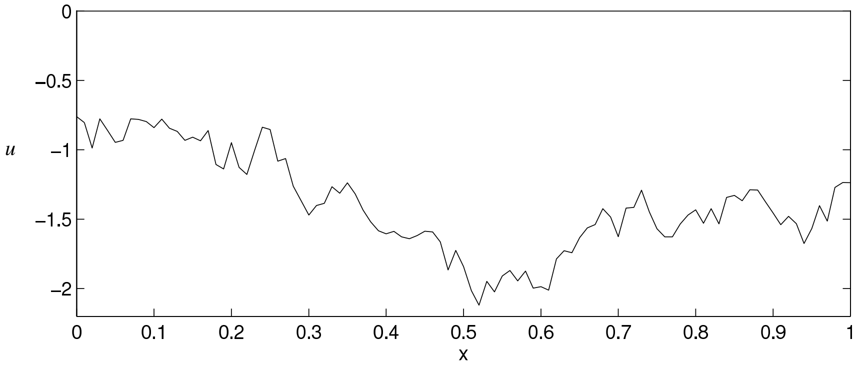


Figure 4. Sample path generated according to π in (22).

of W will be skewed in the same direction of the probability density near X_k. If one or more of the X_m are far from X_k, the simplex formed by the X_m will have the wrong shape. In contrast, the stretch move only requires that we choose one point X_j in the same region as X_k. This suggests that it might be desirable to use proposals which depend on clusters of near by particles. We have been unable to find such a method that is at the same time reasonably quick and has the Markov property, and is even approximately affine invariant. The replacement move may have promise in this regard.

4.2. ***The invariant measure of an SPDE.*** In our second example we attempt to generate samples of the infinite-dimensional measure on continuous functions of [0, 1] defined formally by

$$\exp\Bigl(-\int_0^1 \tfrac{1}{2}u_x(x)^2 + V(u(x))\,dx\Bigr), \tag{20}$$

where V represents the double well potential

$$V(u) = (1-u^2)^2.$$

This measure is the invariant distribution of the stochastic Allen–Cahn equation

$$u_t = u_{xx} - V'(u) + \sqrt{2}\,\eta, \tag{21}$$

with free boundary condition at $x = 0$ and $x = 1$ [3; 12]. In these equations η is a space time white noise. Samples of this measure tend to resemble rough horizontal lines found either near 1 or near -1 (see Figure 4).

In order to sample from this distribution (or approximately sample from it) one must first discretize the integral in (20). The finite-dimensional distribution can

method	time
Metropolis	80
stretch moves	5.2
walk moves, $\lvert S\rvert = 3$	1.4

Table 2. Autocorrelation times (multiplied by 10^{-3} with f given in (23) for single-particle Metropolis and the chains generated by the two ensemble methods. The ensemble size is 102. Note that in terms of CPU time in our implementation, the Metropolis scheme is about five times more costly per step than the other two methods. We have not adjusted these autocorrelation times to incorporate the extra computational requirements of the Metropolis scheme.

then be sampled by Markov chain Monte Carlo. We use the discretization

$$\pi(u(0), u(h), u(2h)\ldots, u(1)) = \\ \exp\biggl(-\sum_{i=0}^{N-1}\frac{1}{2h}(u((i+1)h)-u(ih))^2+\frac{h}{2}\bigl(V(u((i+1)h)+u(ih))\bigr)\biggr), \tag{22}$$

where N is a large integer and $h = 1/N$. This distribution can be seen to converge to (20) in an appropriate sense as $N \to \infty$. In our experiments we choose $N = 100$. Note that the first term in (22) strongly couples neighboring values of u in the discretization while the entire path roughly samples from the double well represented by the second term in (22).

For this problem we compare the auto correlation time for the function

$$f(u(0), u(h), \ldots, u(1)) = \sum_{i=0}^{N-1}\frac{h}{2}\bigl(u((i+1)h)+u(ih)\bigr), \tag{23}$$

which is the trapezoidal rule approximation of the integral $\int_0^1 u(x)\,dx$. As before we use $\lvert S\rvert = 3$ for the walk step and $T_e = T_s/L$ where $T_s = 10^{11}$ and $L = 102$. As with most MCMC schemes that employ global moves (moves of many or all components at a time), we expect the performance to decrease somewhat as one considers larger and larger problems. However, as the integrated auto correlation times reported in Table 2 indicate, the walk move outperforms single site Metropolis by more than a factor of 50 on this relatively high-dimensional problem. Note that in terms of CPU time in our implementation, the Metropolis scheme is about 5 times more costly per step than the other two methods tested. We have not adjusted the autocorrelation times in Table 2 to incorporate the extra computational requirements of the Metropolis scheme.

5. Software

Most of the software used here is available on the web (for example, `Acor`). We have taken care to supply documentation and test programs, and to create easy general user interfaces. The user needs only to supply procedures in `C` or `C++` that evaluate $\pi(x)$ and $f(x)$, and one that supplies the starting ensemble $\vec{X}(1)$. We appreciate feedback on user experiences.

The `Acor` program for estimating τ uses a self consistent window strategy related to that of [7] to estimate (18) and (17). Suppose the problem is to estimate the autocorrelation time for a time series, $f^{(0)}(t)$, and to get an error bar for its mean, $\overline{f}$. The old self consistent window estimate of τ (see (17) and [13]) is

$$\widehat{\tau}^{(0)} = \min\left\{ s \;\middle|\; 1 + 2 \sum_{1 \le t \le Ms} \frac{\widehat{C}^{(0)}(t)}{\widehat{C}^{(0)}(0)} = s \right\}, \tag{24}$$

where $\widehat{C}(t)$ is the estimated autocovariance function

$$\widehat{C}^{(0)}(t) = \frac{1}{T-t} \sum_{t'=1}^{T-t} (f^{(0)}(t') - \overline{f})(f^{(0)}(t+t') - \overline{f}). \tag{25}$$

The window size is taken to be $M = 10$ in computations reported here. An efficient implementation would use an FFT to compute the estimated autocovariance function. The overall running time would be $O(T \ln(T))$.

The new `Acor` program uses a trick that avoids the FFT and has an $O(T)$ running time. It computes the quantities $\widehat{C}^{(0)}(t)$ for $t = 0, \ldots, R$. We used $R = 10$ in the computations presented here. If (24) indicates that $M\widehat{\tau} > R$, we restart after a pairwise reduction

$$f^{(k+1)}(t) = \tfrac{1}{2}\left(f^{(k)}(2t) \;+\; f^{(k)}(2t+1)\right).$$

The new time series is half as long as the old one and its autocorrelation time is shorter. Repeating the above steps (25) and (24) successively for $k = 1, 2, \ldots$ gives an overall $O(T)$ work bound. Of course, the (sample) mean of the time series $f^{(k)}(t)$ is the same $\overline{f}$ for each k. So the error bar is the same too. Eventually we should come to a k where (24) is satisfied for $s \le R$. If not, the procedure reports failure. The most likely cause is that the original time series is too short relative to its autocorrelation time.

6. Conclusions

We have presented a family of many particle ensemble Markov chain Monte Carlo schemes with an affine invariance property. Such samplers are uniformly effective on problems that can be rescaled by affine transformations to be well conditioned. All Gaussian distributions and convex bodies have this property. Numerical tests indicate that even on much more general distributions our methods can offer significant performance improvements over standard single-particle methods. The computational cost of our methods over standard single-particle schemes is negligible.

Acknowledgments

Our work grew out of discussions of a talk by Colin Fox at the 2009 SIAM meeting on Computational Science. We are grateful to our colleagues Jeff Cheeger and Esteban Tabak for useful comments and suggestions.

References

[1] C. Andrieu and J. Thoms, *A tutorial on adaptive MCMC*, Stat. Comput. **18** (2008), 343–373.

[2] J. Christen, *A general purpose scale-independent MCMC algorithm*, technical report I-07-16, CIMAT, Guanajuato, 2007.

[3] G. Da Prato and J. Zabczyk, *Ergodicity for infinite-dimensional systems*, London Math. Soc. Lecture Note Series, no. 229, Cambridge University Press, Cambridge, 1996. MR 97k:60165 Zbl 0849.60052

[4] P. Diaconis and L. Saloff-Coste, *What do we know about the Metropolis algorithm?*, J. Comput. System Sci. **57** (1998), no. 1, 20–36. MR 2000b:68094 Zbl 0920.68054

[5] K. Gade, *Numerical estimation of the second largest eigenvalue of a reversible markov transition matrix*, Ph.D. thesis, Courant Institute, New York University, 2008.

[6] P. E. Gill, W. Murray, and M. H. Wright, *Practical optimization*, Academic Press, London, 1981. MR 83d:65195 Zbl 0503.90062

[7] J. Goodman and A. Sokal, *Multigrid Monte Carlo method: conceptual foundations*, Phys. Rev. D **40** (1989), no. 6, 2035–2071.

[8] F. John, *Extremum problems with inequalities as subsidiary conditions*, Studies and essays presented to R. Courant on his 60th birthday (K. O. Friedrichs, O. E. Neugebauer, and J. J. Stoker, eds.), Interscience, New York, 1948, pp. 187–204. MR 10,719b Zbl 0034.10503

[9] M. H. Kalos and P. A. Whitlock, *Monte Carlo methods*, 2nd ed., Wiley, 2008. MR 2503174 Zbl 1170.65302

[10] J. S. Liu, *Monte Carlo strategies in scientific computing*, Springer Series in Stat., no. 16, Springer, New York, 2001. MR 2002i:65006 Zbl 0991.65001

[11] J. A. Nelder and R. Mead, *A simplex method for function minimization*, Comput. J. **7** (1965), 308–313. Zbl 0229.65053

[12] M. G. Reznikoff and E. Vanden-Eijnden, *Invariant measures of stochastic partial differential equations and conditioned diffusions*, C. R. Math. Acad. Sci. Paris **340** (2005), no. 4, 305–308. MR 2121896 Zbl 1063.60092

[13] A. Sokal, *Monte Carlo methods in statistical mechanics: foundations and new algorithms*, Functional integration, Plenum, NY, 1997, pp. 131–192. Zbl 0890.65006

Received November 6, 2009.

JONATHAN GOODMAN: goodman@cims.nyu.edu
Courant Institute, New York University, 251 Mercer St., New York, NY 10012, United States

JONATHAN WEARE: weare@cims.nyu.edu
Courant Institute, New York University, 251 Mercer St., New York, NY 10012, United States

COMM. APP. MATH. AND COMP. SCI.
Vol. 5, No. 1, 2010

A SECOND-ORDER ACCURATE METHOD FOR SOLVING THE SIGNED DISTANCE FUNCTION EQUATION

PETER SCHWARTZ AND PHILLIP COLELLA

We present a numerical method for computing the signed distance to a piecewise-smooth surface defined as the zero set of a function. It is based on a marching method by Kim (2001) and a hybrid discretization of first- and second-order discretizations of the signed distance function equation. If the solution is smooth at a point and at all of the points in the domain of dependence of that point, the solution is second-order accurate; otherwise, the method is first-order accurate, and computes the correct entropy solution in the presence of kinks in the initial surface.

1. Introduction

Let Γ be a continuous, piecewise smooth $(\mathbf{D}-1)$-dimensional manifold in $\mathbb{R}^{\mathbf{D}}$ defined implicitly as the zero set of a function, that is, there is a continuous piecewise smooth ϕ defined on some ϵ-neighborhood of Γ such that

$$\Gamma = \{\boldsymbol{x} : \phi(\boldsymbol{x}) = 0\}. \tag{1}$$

We also assume that $\nabla\phi$ is bounded and piecewise smooth on Γ, and that there is a constant $c > 0$ such that $|\nabla\phi(\boldsymbol{x}_0)| \geq c$ at all points, $\boldsymbol{x}_0 \in \Gamma$ where $\nabla\phi$ is defined. At such points, $\hat{\boldsymbol{n}}$, the unit normal to Γ, is given by

$$\hat{\boldsymbol{n}} = \frac{\nabla\phi}{|\nabla\phi|}.$$

Given such a surface Γ, we can define the signed distance function ψ

$$\psi(\boldsymbol{x}) = s \min_{\boldsymbol{x}' \in \Gamma} |\boldsymbol{x} - \boldsymbol{x}'| = \mathrm{sdist}(\boldsymbol{x}, \Gamma), \tag{2}$$

where s is defined to be the positive on one side of Γ and negative on the other. If $\boldsymbol{x}_0 \in \Gamma$, is a point at which the minimum in the right-hand side of (2) is achieved, and Γ is smooth at that point, then, $s = \mathrm{sign}((\boldsymbol{x} - \boldsymbol{x}_0) \cdot \nabla\phi(\boldsymbol{x}_0))$. If Γ is not smooth at that point, then s is the single value taken on by $\mathrm{sign}((\boldsymbol{x} - \boldsymbol{x}') \cdot \nabla\phi(\boldsymbol{x}'))$ at all

MSC2000: primary 65-02; secondary 76-02.
Keywords: eikonal, narrow band, Hamilton–Jacobi, signed distance function.

points sufficiently close to $\boldsymbol{x}_0$ such that $\nabla\phi(\boldsymbol{x}')$ is defined. In any case, $s = \pm 1$ on $\mathbb{R}^{\mathbf{D}} - \Gamma$ and changes only at Γ.

If $\psi(\boldsymbol{x})$ is smooth, then ψ satisfies the signed distance function equation.

$$|\nabla\psi(\boldsymbol{x})| = 1. \tag{3}$$

In that case, solutions to the signed distance function equation satisfy the characteristic equations

$$\begin{aligned} \frac{d\boldsymbol{x}}{d\sigma} &= \boldsymbol{w}, \quad && \boldsymbol{x}(0) = \boldsymbol{x}_0, \\ \frac{d\boldsymbol{w}}{d\sigma} &= 0, \quad && \boldsymbol{w}(0) = (\nabla\psi)(\boldsymbol{x}_0), \\ \frac{d\psi}{d\sigma} &= 1, \quad && \psi(0) = \psi(\boldsymbol{x}(0)), \end{aligned}$$

where σ denotes arc length. These equations can be solved analytically to obtain

$$\boldsymbol{x}(\sigma) = \boldsymbol{x}(0) + \sigma(\nabla\psi)(\boldsymbol{x}(0)), \quad \boldsymbol{w}(\sigma) = (\nabla\psi)(\boldsymbol{x}(0)), \quad \psi(\sigma) = \psi(\boldsymbol{x}(0)) + \sigma, \tag{4}$$

that is, the curves are straight lines in $(\boldsymbol{x}, \psi)$ space, while $\boldsymbol{w} = \nabla\psi$ is constant along each trajectory.

The characteristic form of the equations suggest that signed-distance functions can be constructed incrementally. Given that ψ is known on $\Omega_r = \{\boldsymbol{x} : |\psi(\boldsymbol{x})| \leq r\}$, then one can extend ψ to $\Omega_{r+\delta}$ using (4). It is easy to show that this reasoning extends to nonsmooth signed distance functions, that is, ones defined by (2). Fast marching methods [13; 7] are numerical methods for computing the signed distance function based on this observation. Fast marching methods have two components:

(1) A discretization of the signed distance function equation that permits the calculation of the signed distance at a given grid point by using a stencil of nearby values that have already been computed.

(2) A marching algorithm, which is a method for determining the order in which grid values are to be computed.

For example, the method in [7; 13] uses a first-order accurate discretization of the signed distance function equation, and a marching algorithm based on computing, at each step, the value of ψ that has the minimum magnitude among all of the uncomputed values adjacent to valid values.

A number of problems in numerical simulation related to implicit function representation of surfaces require the computation of the signed distance from a given surface; see [14; 2]. The motivating application for this paper is the use of narrow-band level-set methods for representing the propagation of fronts in large-scale fluid dynamics simulations combined with second-order accurate volume-of-fluid methods [4] for discretizing the PDE on either side of the front. This imposes two requirements that have not been simultaneously met by previous methods. The first is the use of a marching method that is a good match for adaptive and parallel implementation based on patch-based domain decomposition. We impose this requirement for compatibility with the software frameworks typically used for high-performance implementations of block-structured adaptive grid methods. In such an approach, the construction of a solution is based on steps that update independently the points on a collection of rectangles whose disjoint union covers the domain, interleaved with steps that communicate ghost cell data. The marching method in [7; 13; 6] does not fit into this category: it is specified as a serial algorithm, in that the values on a grid are computed one at a time, with the next value/location determined by the previously computed values. Not only is this a poor match for the block-structured software frameworks, but it also imposes a serial bottleneck in a parallel computation. The second requirement is that we obtain a solution that is second-order accurate at all points whose domain of dependence includes no singularities, since the volume-of-fluid discretizations requires that level of accuracy [12; 5; 11]. In all cases, the solution should converge to a signed-distance function, even in regions whose domain of dependence include discontinuities in the derivatives. While second-order accurate algorithms have been proposed [14; 3; 9; 10; 15], not a great deal of attention has been paid to distinguishing between converging and diverging characteristics for an initial surface that contains kinks in the context of second-order accurate methods.

In the present work, we present a method that meets our requirements. We use a variation on the global marching method in [8]. Given the values at grid points in Ω_r, we compute simultaneously and independently all of the grid values in $\Omega_{r+\delta}$, where δ is comparable to the mesh spacing. Since the method computes the solution at a large number of points independently as local functions of the previously computed values in Ω_r, the method maps naturally onto a block-structured domain-decomposition implementation. Second, our discretization of the signed distance function equation is analogous to the construction of the fluxes for a second-order Godunov method for a scalar conservation law. It is a hybridization of a high-order and low-order method, where the choice of hybridization coefficient is based on a local curvature calculation. The high-order method is a straightforward difference approximation to the characteristic form of the equations (4). The low-order method is similar to the method in [7; 13] but uses a least-squares approach for computing $\nabla\psi$ based on different approximations depending on whether the characteristics are locally converging or diverging. The choice of δ is based on a condition analogous to a Courant–Friedrichs–Lewy (CFL) condition under which all the points in the high-order stencil should be available for computing the value of ψ at a grid point. The use of a least-squares algorithm for approximating the gradient in the low-order method involving all of the valid nearest neighbors maximizes the likelihood that there will be sufficient valid points for computing the low-order estimate for ψ when it is needed.

The resulting method is second-order accurate in regions where the solution is smooth, and characteristics trace back to portions of the original surface Γ that are smooth. If there are kinks in the original surface or that form away from the original surface due to convergence of characteristics, the method is first-order accurate in the range of influence of the kinks. The method appears to provide solutions that satisfy the entropy condition, correctly distinguishing between the two directions of propagation from kinks in the original surface. The solution on the side of the surface corresponding to converging characteristics propagates as a kink, while the solution on the side corresponding to diverging characteristics takes the form of a centered expansion fan.

2. Kim's global marching method

We discretize the problem to a grid consisting of equally spaced points in $\mathbb{Z}^{\mathbf{D}}$. We denote the grid-spacing by h. Given

$$\phi_{\boldsymbol{i}} = \phi(\boldsymbol{i}h),$$

where $\boldsymbol{i} \in \mathbb{Z}^{\mathbf{D}}$ and $\boldsymbol{i}h$ in a $\epsilon-$neighborhood of Γ, we wish to compute

$$\psi_{\boldsymbol{i}} \approx \psi(\boldsymbol{i}h)\ ,\ |\psi_{\boldsymbol{i}}| \leq R. \tag{5}$$

Our marching algorithm for computing such solutions is given in the box on the next page.

Here the function $\mathscr{E}(\psi, \Omega^{\text{valid}}, \boldsymbol{i})$ computes a value for ψ at $\boldsymbol{i}$ using only the set of values $\{\psi_{\boldsymbol{i}}\}$ that have been computed on Ω^{valid}. $\mathscr{E}$ can be undefined, for example, if there are insufficient points in a neighborhood of $\boldsymbol{i}$ to perform the computation. The quantity σ is a CFL number for the marching method, and depends on the details of $\mathscr{E}$. In determining which points over which to iterate in the **for** loop, we have assumed that $\sigma < 1$. The computation in the **for** loop can be performed in parallel using a domain-decomposition strategy over the points adjacent to the valid region denoted by

$$\bigcup_{\boldsymbol{s}:|s_d|\leq 1} (\Omega^{\text{valid}} + \boldsymbol{s}) - \Omega^{\text{valid}}.$$

In principle, the method described here could iterate an arbitrarily large number of times before updating r. For the discretization method described in the next section,

```
$\Omega^{\text{new}} = \varnothing$
$r = \epsilon + \sigma h$
while $r \le R$ do
    for $\boldsymbol{i} \in \bigcup_{\boldsymbol{s}:|s_d|\le 1} (\Omega^{\text{valid}} + \boldsymbol{s}) - \Omega^{\text{valid}}$ do
        if $\mathscr{E}(\psi, \boldsymbol{v}, \Omega^{\text{valid}}, \boldsymbol{i})$ is defined then
            $(\tilde{\psi}_{\boldsymbol{i}}, \tilde{\boldsymbol{v}}_{\boldsymbol{i}}) = \mathscr{E}(\psi, \boldsymbol{v}, \Omega^{\text{valid}}, \boldsymbol{i})$
            if $|\tilde{\psi}_{\boldsymbol{i}}| \le \texttt{r}$ then
                $\Omega^{\text{new}}$ += $\{\boldsymbol{i}\}$
                numUpdate += 1
            end if
        end if
    end for
    $\Omega^{\text{valid}}$ += $\Omega^{\text{new}}$
    $\psi = \tilde{\psi}$, $\boldsymbol{v} = \tilde{\boldsymbol{v}}$ on $\Omega^{\text{new}}$
    $\Omega^{\text{new}} = \varnothing$
    if numUpdate $= 0$ then
        $r$ += $\sigma h$
    end if
    numUpdate = 0
end while
```

The global marching method. In each iteration of the **while** loop, we compute the solution to on points adjacent to $\Omega_{r-\sigma h} \subseteq \Omega^{\text{valid}} \subseteq \Omega_r$ independently of the other values being computed in that iteration. After there are no longer any points to compute, we increment $r \to r + \sigma h$.

we have observed that numUpdate $= 0$ on the third iteration, so we could replace the **while** loop by one performing a fixed number of iterations before updating r.

3. Discretizing the signed distance function equation

In this section, we define the discretization of the signed distance function equation used to define $\mathscr{E}$. It is computed as a linear combination of a low-order (first-order) method and a high-order (second-order) method, with the hybridization coefficient depending on the local curvature. This approach is analogous to that taken in constructing fluxes for hyperbolic conservation laws. The low-order method is based on a least-squares discretization of the gradient that distinguishes between locally converging and diverging characteristics. The signed distance function (3) is used to determine the free parameter in the gradient corresponding to the unknown value of Ψ. This step is similar to that used in [13] and [7]. The high-order method is based on solving the characteristic form of the equations.

3.1. ***Least-squares discretization.*** Given a collection of points $\boldsymbol{p} \in \boldsymbol{P} \subset \mathbb{Z}^{\mathbf{D}}$, we have the following relationship between the values of the distance function, ψ, and the gradient:

$$\frac{1}{h}\big(\psi(\boldsymbol{i}h) - \psi((\boldsymbol{i}+\boldsymbol{p})h)\big) = -\boldsymbol{p}\cdot\nabla\psi + O(h) \text{ for } \boldsymbol{p} \in \boldsymbol{P}. \tag{6}$$

If $\boldsymbol{P}$ has $\mathbf{D}$ linearly independent elements, then we can use (6) as the starting point for deriving a first-order accurate method for computing solutions to (3). Given

$$\psi_{\boldsymbol{i}+\boldsymbol{p}} \approx \psi((\boldsymbol{i}+\boldsymbol{p})h) \text{ for } \boldsymbol{p} \in \boldsymbol{P}, \tag{7}$$

we define $\tilde{\psi} \approx \psi(\boldsymbol{i}h)$, $\boldsymbol{v} \approx (\nabla\psi)(\boldsymbol{i}h)$ as satisfying a least-squares solution to the coupled equations:

$$A\boldsymbol{v} = -\frac{1}{h}(\tilde{\psi}\Upsilon - \Psi), \tag{8}$$

where the unknown $\tilde{\psi}$ is viewed as a free parameter, to be determined later, and

$$\Psi = (\psi_{\boldsymbol{i}+\boldsymbol{p}_1}, \psi_{\boldsymbol{i}+\boldsymbol{p}_2}, \dots \psi_{\boldsymbol{i}+\boldsymbol{p}_r})^T, \tag{9}$$

$$A = (\boldsymbol{p}_1, \boldsymbol{p}_2, \dots \boldsymbol{p}_r)^T, \tag{10}$$

$$\Upsilon = (1, 1, \dots 1)^T. \tag{11}$$

Since A is of rank $\mathbf{D}$, the least-squares solution to (8) is given by

$$\boldsymbol{v} = -(A^TA)^{-1}A^T\frac{1}{h}(\tilde{\psi}\Upsilon - \Psi) = \frac{\tilde{\psi}-\bar{\psi}}{\ell}\hat{\boldsymbol{n}} - (\boldsymbol{\omega}_2 - (\boldsymbol{\omega}_2\cdot\hat{\boldsymbol{n}})\hat{\boldsymbol{n}}), \tag{12}$$

where

$$\begin{gathered}\boldsymbol{\omega}_1 = -\frac{1}{h}(A^TA)^{-1}A^T\Upsilon, \qquad \boldsymbol{\omega}_2 = -\frac{1}{h}(A^TA)^{-1}A^T\Psi, \\ \ell = \frac{1}{\|\boldsymbol{\omega}_1\|}, \quad \hat{\boldsymbol{n}} = \boldsymbol{\omega}_1\ell, \quad \bar{\psi} = (\boldsymbol{\omega}_2\cdot\boldsymbol{\omega}_1)\ell^2 = \psi(\boldsymbol{i}h - \ell\hat{\boldsymbol{n}}) + O(h^2).\end{gathered} \tag{13}$$

We assume here that $\boldsymbol{\omega}_1$ is not the zero vector. If $\boldsymbol{\omega}_1 = \boldsymbol{\omega}_1(\boldsymbol{P}) = \mathbf{0}$, then the least-squares problem does not produce a value for $\tilde{\psi}$, although the expression (12) for the gradient is still well-defined.

Following [7; 13], the condition $\|\boldsymbol{v}\|^2 = 1$ leads to a quadratic equation for $\tilde{\psi}$:

$$(\tilde{\psi} - \bar{\psi})^2 + \ell^2\|(\boldsymbol{\omega}_2 - (\boldsymbol{\omega}_2\cdot\hat{\boldsymbol{n}})\hat{\boldsymbol{n}})\|^2 = \ell^2. \tag{14}$$

If (14) has two real roots, we choose the root for which $|\tilde{\psi}| > |\bar{\psi}|$. If (14) has no real roots, we set $\tilde{\psi} = \bar{\psi}$.

We denote by $\mathscr{L}(\psi, \boldsymbol{i}, h, \boldsymbol{P})$ the value of $\tilde{\psi}$ obtained from the least-squares algorithm above. We can then define

$$\mathscr{E}^L(\psi, \Omega^{\text{valid}}, \boldsymbol{i}, h) = (\psi^L, \boldsymbol{v}^L), \tag{15}$$

$$\psi^L = \begin{cases} s_{\boldsymbol{i}} \min_{\boldsymbol{B}} |\mathscr{L}(\psi, \boldsymbol{i}, h, \boldsymbol{B})| & \text{if } \kappa_{\boldsymbol{i}} < 0 \ \text{ or } \ \boldsymbol{\omega}_1 = \boldsymbol{0}, \\ \mathscr{L}(\psi, \boldsymbol{i}, h, \boldsymbol{P}) & \text{if } \kappa_{\boldsymbol{i}} \geq 0 \text{ and } \boldsymbol{\omega}_1 \neq \boldsymbol{0}, \end{cases} \tag{16}$$

$$\boldsymbol{P} = \boldsymbol{U} \cap (\Omega^{\text{valid}} - \boldsymbol{i}), \qquad (A, \Psi, \boldsymbol{\omega}_1) = (A(\boldsymbol{P}), \Psi(\boldsymbol{P}), \boldsymbol{\omega}_1(\boldsymbol{P})), \tag{17}$$

$$\boldsymbol{v}^L = -(AA^T)^{-1} A^T \frac{1}{h} (\psi^L \Upsilon - \Psi). \tag{18}$$

In (17) we have introduced $\boldsymbol{U} = \{\boldsymbol{u} : |\boldsymbol{u}_d - \boldsymbol{i}_d| \leq 1\}$. The minimum in (16) is over the collection of all sets $\boldsymbol{B}$ of pairs of adjacent points (if $\mathbf{D} = 2$) or 2×2 blocks of points (if $\mathbf{D} = 3$) contained in $\boldsymbol{U}$ such that $\boldsymbol{B} + \boldsymbol{i} \subset \Omega^{\text{valid}}$. The quantity κ is a local estimate of the curvature:

$$\kappa_{\boldsymbol{i}} = \min_{\boldsymbol{t}} s_{\boldsymbol{i}} (\Delta^h \psi)_{\boldsymbol{i}+\boldsymbol{t}}, \tag{19}$$

where Δ^h is the $2\mathbf{D} + 1$-point centered-difference discretization of Laplacian, and the minimum is taken over all points $\boldsymbol{t} \in [-2 \ldots 2]^{\mathbf{D}}$ such that the stencil for Δ^h evaluated at $\boldsymbol{i} + \boldsymbol{t}$ is contained in Ω^{valid}. The minimum assumption for $\mathscr{E}^L$ to be defined is that at least one of the $\boldsymbol{B}$ in (16) is defined, and at least one of the $\Delta^h \psi$ in (19) is defined. Otherwise, $\mathscr{E}^L$ is undefined.

We use the two different least-squares algorithm depending on the sign of the curvature in order to obtain the correct distance function in the neighborhood of a kink. If the curvature is negative, the characteristics are converging, and the distance function is the minimum over as many candidates as possible based on using the least-squares algorithm on $2^{\mathbf{D}-1}$ points, analogous to choosing the minimum over multiple distinct characteristics that might be reaching the same point. If the curvature is positive, the characteristics are diverging, and the use of the single stencil involving all of the valid points in $\boldsymbol{U} + \boldsymbol{i}$ leads to interpolated intermediate values for $\overline{\psi}$ and $\hat{\boldsymbol{n}}$, analogous to sampling inside a centered rarefaction fan in computing a flux for Godunov's method at a sonic point.

3.2. ***A second-order accurate method.*** We define a function that computes a second order approximation to the distance function and the gradient of the distance function. In the following, let $\pi = \psi, \boldsymbol{v}$ denote the field that we wish to compute at $\boldsymbol{i} \notin \Omega^{\text{valid}}$, assuming that π is known on Ω^{valid}. We also assume that we know $\hat{\boldsymbol{v}} \approx (\nabla \psi)(\boldsymbol{i}h)$. The calculation of

$$\bar{\pi} = Q(\pi, \boldsymbol{i}, \hat{\boldsymbol{v}}, h) \approx \pi(\boldsymbol{i}h) \tag{20}$$

is given as follows.

1. Compute $\bar{\boldsymbol{x}}$, the first point along the ray $\{\boldsymbol{i}h - s_{\boldsymbol{i}}\hat{\boldsymbol{v}}\delta : \delta > 0\}$ that intersects a coordinate plane of gridpoints:
$$\bar{\boldsymbol{x}} = \boldsymbol{i}h - s_{\boldsymbol{i}}h\frac{\hat{\boldsymbol{v}}}{\hat{v}_{\text{max}}}, \tag{21}$$
where $\hat{v}_{\text{max}}$ is the component of $\hat{\boldsymbol{v}}$ whose magnitude is largest, with d_{max} the corresponding coordinate direction.

2. Compute a quadratic interpolant in the coordinate plane containing $\bar{\boldsymbol{x}}$:
$$\boldsymbol{j} = \big\lfloor \bar{\boldsymbol{x}}/h - \tfrac{1}{2}(\boldsymbol{u} - \boldsymbol{e}^{d_{\text{max}}})\big\rfloor, \qquad \bar{\boldsymbol{y}} = \bar{\boldsymbol{x}} - \boldsymbol{j}h, \tag{22}$$
$$\bar{\pi} = \pi_{\boldsymbol{j}} + \sum_{d \neq d_{\text{max}}} \left(\frac{\partial \pi}{\partial x_d}\bar{y}_d + \frac{1}{2}\frac{\partial^2 \pi}{\partial x_d^2}\bar{y}_d^2\right) + \frac{\partial^2 \pi}{\partial x_{d_1}\partial x_{d_2}}\bar{y}_{d_1}\bar{y}_{d_2}, \tag{23}$$
where all of the derivatives are evaluated at $\boldsymbol{j}h$. The last term in (23) is defined only for $\mathbf{D} = 3$ and $d_1 \neq d_2, d_1, d_2 \neq d_{\text{max}}$. We denote by $\boldsymbol{e}^d$ the unit vector in the dth coordinate direction, and $\boldsymbol{u} = (1 \ldots 1)$, both elements of $\mathbb{Z}^{\mathbf{D}}$.

The derivatives appearing in the sum in (23) are computed using second-order accurate centered differences at $\boldsymbol{j}h$, assuming $\boldsymbol{j}, \boldsymbol{j} \pm \boldsymbol{e}^d \in \Omega^{\text{valid}}$. The mixed derivative is approximated by the average of centered differences:
$$\frac{\partial^2 \pi}{\partial x_{d_1}\partial x_{d_2}} \approx \frac{1}{N}\sum (D^2_{d_1,d_2}\pi)_{\boldsymbol{j}+\boldsymbol{s}/2} \tag{24}$$
where
$$(D^2_{d_1,d_2}\pi)_{\boldsymbol{k}+\boldsymbol{e}^{d_1}/2+\boldsymbol{e}^{d_1}/2} = \frac{1}{h^2}(\pi_{\boldsymbol{k}} + \pi_{\boldsymbol{k}+\boldsymbol{e}^{d_1}+\boldsymbol{e}^{d_1}} - \pi_{\boldsymbol{k}+\boldsymbol{e}^{d_1}} - \pi_{\boldsymbol{k}+\boldsymbol{e}^{d_2}}) \tag{25}$$
is defined if $\boldsymbol{k},\ \boldsymbol{k} + \boldsymbol{e}^{d_1},\ \boldsymbol{k} + \boldsymbol{e}^{d_2},\ \boldsymbol{k} + \boldsymbol{e}^{d_1} + \boldsymbol{e}^{d_2}$ are all in Ω^{valid}. The sum in (24) is taken over all $\boldsymbol{s}$ of the form $\alpha_1\boldsymbol{e}^{d_1} + \alpha_2\boldsymbol{e}^{d_2}$, $\alpha_1 = \pm 1, \alpha_2 = \pm 1$ for which $(D^2_{d_1,d_2})$ is defined, and N is the number of terms in the sum.

Given the function Q defined above, we can define a second-order accurate discretization of the characteristic form of the equations (4) at $\boldsymbol{i}h$. We iterate twice to obtain a sufficiently accurate computation of $\boldsymbol{v}$, computing $\hat{\boldsymbol{v}}$ at the point $\boldsymbol{i}$ using the least-squares algorithm defined in the previous section, and then
$$\hat{\boldsymbol{v}} := Q(\boldsymbol{v}, \Omega^{\text{valid}}, \boldsymbol{i}, \hat{\boldsymbol{v}}, h),\ \boldsymbol{v}^H = Q(\boldsymbol{v}, \Omega^{\text{valid}}, \boldsymbol{i}, \hat{\boldsymbol{v}}, h). \tag{26}$$
We then use $\boldsymbol{v}^H$ to compute ψ^H:
$$\psi^H = Q(\psi, \Omega^{\text{valid}}, \boldsymbol{i}, \hat{\boldsymbol{v}}^H, h) + s_{\boldsymbol{i}}h\left|\frac{\boldsymbol{v}^H}{v^H_{\text{max}}}\right|. \tag{27}$$
We denote by $\mathscr{E}^H$ the resulting second-order accurate method for computing $\psi, \boldsymbol{v}$:
$$\mathscr{E}^H(\phi, \boldsymbol{v}, \Omega^{\text{valid}}, \boldsymbol{i}) \equiv (\psi^H, \boldsymbol{v}^H). \tag{28}$$

If the low-order method is defined, and the points required for the various evaluations of Q are defined, then (28) is defined. Otherwise, it is undefined.

3.3. ***Hybridization.*** We hybridize the low- and high-order methods based on the magnitude of the curvature. Assuming both $\mathscr{E}^L$ and $\mathscr{E}^H$ are defined, we compute

$$(\psi^L, \boldsymbol{v}^L) = \mathscr{E}^L(\psi, \boldsymbol{v}, \Omega^{\text{valid}}, \boldsymbol{i}), \tag{29}$$

$$(\psi^H, \boldsymbol{v}^H) = \mathscr{E}^H(\psi, \boldsymbol{v}, \Omega^{\text{valid}}, \boldsymbol{i}), \tag{30}$$

$$\mathscr{E}(\psi, \boldsymbol{v}, \Omega^{\text{valid}}, \boldsymbol{i}) = \big((1-\eta_{\boldsymbol{i}})\psi^H + \eta_{\boldsymbol{i}}\psi^L, (1-\eta_{\boldsymbol{i}}^2)\boldsymbol{v}^H + \eta_{\boldsymbol{i}}^2\boldsymbol{v}^L\big), \tag{31}$$

where the hybridization parameter η is given by

$$\eta_{\boldsymbol{i}} = \begin{cases} 1 & \text{if } h|\Delta^h\psi|_{\max} > C, \\ h/C|\Delta^h\psi|_{\max} & \text{otherwise,} \end{cases} \tag{32}$$

$$|\Delta^h\psi|_{\max} = \max_{\boldsymbol{t}} |(\Delta^h\psi)_{\boldsymbol{i}+\boldsymbol{t}}|, \tag{33}$$

where the range over which the max is taken is the same as in (19). If the high-order value $\mathscr{E}^H(\psi, \boldsymbol{v}, \Omega^{\text{valid}}, \boldsymbol{i})$ is not defined, but the low-order value is, then

$$\mathscr{E}(\psi, \boldsymbol{v}, \Omega^{\text{valid}}, \boldsymbol{i}) = \mathscr{E}^L(\psi, \boldsymbol{v}, \Omega^{\text{valid}}, \boldsymbol{i}). \tag{34}$$

If the low order value is not defined, then $\mathscr{E}(\psi, \boldsymbol{v}, \Omega^{\text{valid}}, \boldsymbol{i})$ is not defined. The constant C is an empirically determined parameter, independent of h. In our numerical experiments, $C = 1$.

If $\sigma < 1/\sqrt{5}$, and we replace the values of ψ, $\boldsymbol{v}$ on grid points in Ω_r with those of a smooth distance function ψ^e, it is possible to show that, for sufficiently small h, both $\mathscr{E}^H$ and $\mathscr{E}^L$ are defined for all grid points in $\Omega_{r+\sigma h}$ and that

$$\psi_{\boldsymbol{i}}^H = \psi^e(\boldsymbol{i}h) + O(h^3), \boldsymbol{v}^H = \nabla\psi^e(\boldsymbol{i}h) + O(h^3), \tag{35}$$

$$\psi_{\boldsymbol{i}}^L = \psi^e(\boldsymbol{i}h) + O(h^2), \boldsymbol{v}^L = \nabla\psi^e(\boldsymbol{i}h) + O(h), \tag{36}$$

from which it follows that

$$\mathscr{E}(\psi, \boldsymbol{v}, \Omega^{\text{valid}}, \boldsymbol{i}) = (\psi^e(\boldsymbol{i}h), \nabla\psi^e(\boldsymbol{i}h)) + O(h^3). \tag{37}$$

Thus we expect that the global error in our solution will be $O(h^2)$. This also explains why we use η^2, rather than η, to hybridize the gradient calculation. Otherwise, we would introduce an $O(h^2)$ contribution to the error in the gradient at every step, leading to a first-order accurate method for the gradient, and hence for ψ. In the neighborhood of kinks in the level sets of ψ, the value of the curvature is $O(h^{-1})$, and we will use the low-order method, leading to a first-order accurate method in the range of influence of the kinks.

3.4. ***Initialization.*** We now describe the method we use to provide test problems with an initial, narrow band three or four cells wide. We are given an initial representation of the surface by a discretized implicit function, from which we construct the distance function and the gradient of the distance function an $O(h)$ distance. If the surface is smooth, then our initialization procedure is an $O(h^2)$ approximation to the distance function. If the characteristics cross near the surface or the surface is not smooth, then the initialization reduces to a first-order accurate method within the range of influence of the kink.

We require some more notation. Denote by $(G^0\phi)$ the centered difference approximation to the gradient of ϕ. Given a grid location $\boldsymbol{i}$, let

$$d = \frac{\phi_{\boldsymbol{i}}}{\|(G^0\phi)_{\boldsymbol{i}}\|}.$$

Let $\mathcal{P} \subset \Omega$ denote $\boldsymbol{i}$ and its neighbors. Let

$$m_{\boldsymbol{i}} = \min_{\boldsymbol{p}\in\mathcal{P}} \|(G^0\phi)_{\boldsymbol{p}}\|, \quad M_{\boldsymbol{i}} = \max_{\boldsymbol{p}\in\mathcal{P}} \|(G^0\phi)_{\boldsymbol{p}}\|. \tag{38}$$

We choose a nondimensional parameter, ϵ, independent of h and attempt to detect a discontinuity in the gradient by checking whether M exceeds m by an amount greater than ϵ. If so, we make a robust but lower order estimate of the gradient:

$$\text{if } 1 - \frac{m_{\boldsymbol{i}}}{M_{\boldsymbol{i}}} \geq \epsilon, \text{ then } \boldsymbol{v} = (G^0\phi)_{\boldsymbol{p}} : \|(G^0\phi)_{\boldsymbol{p}}\| = M_{\boldsymbol{i}}. \tag{39}$$

In our numerical experiments, $\epsilon = 1/(2\sqrt{2})$. Alternatively, if

$$1 - \frac{m_{\boldsymbol{i}}}{M_{\boldsymbol{i}}} < \epsilon, \tag{40}$$

then we define a point,

$$\boldsymbol{x}_0 = \boldsymbol{i}h - d\frac{(G^0\phi)_{\boldsymbol{i}}}{\|(G^0\phi)_{\boldsymbol{i}}\|}. \tag{41}$$

At $\boldsymbol{x}_0$ we biquadratically interpolate an estimate of the gradient $\boldsymbol{v}$. Finally, we use root-finding in the direction $\boldsymbol{v}$ to make an estimate of the distance.

4. Numerical results

For our fast marching problems, we always compute the max norm of the error. Where useful, we also compute the L_1 and the L_2- norm of the solution error.

For a discrete variable, ζ, the max norm is given by

$$\|\zeta\|_\infty = \max_{i} |\zeta_i|. \tag{42}$$

L^1 norm	rate	L^2 norm	rate	L^∞ norm	rate
4.4566e-02		1.0240e-02		2.1393e-02	
1.0592e-02	2.07	2.4083e-03	2.08	5.9743e-03	1.84

Table 1. Solution error for 2**D** curve in polar coordinates: $h = \frac{1}{100}$ and $\frac{1}{200}$.

The L_p-norm is given by

$$\|\zeta\|_p = \left(\sum_i |(\zeta_i)^p h^{\mathbf{D}}\right)^{1/p}. \tag{43}$$

For all of the test problems that follow we have used a marching parameter of $\sigma = 1/(2\sqrt{5})$.

Our first test problem uses the implicit function $r = 2\cos 4\theta + 7$. The domain has a lower left corner with coordinates $(-10, -10, -10)$ and an upper right corner with coordinates $(10, 10, 10)$. The initial bandwidth is approximately six grid cells wide at all resolutions. The final bandwidth is approximately 1.2. Calculations were performed on grids with $h = \frac{1}{100}$, $\frac{1}{200}$, and $\frac{1}{400}$. Richardson error extrapolation was used to calculate the results presented in Table 1. The solution is shown in Figure 1 and the error is shown in Figure 2.

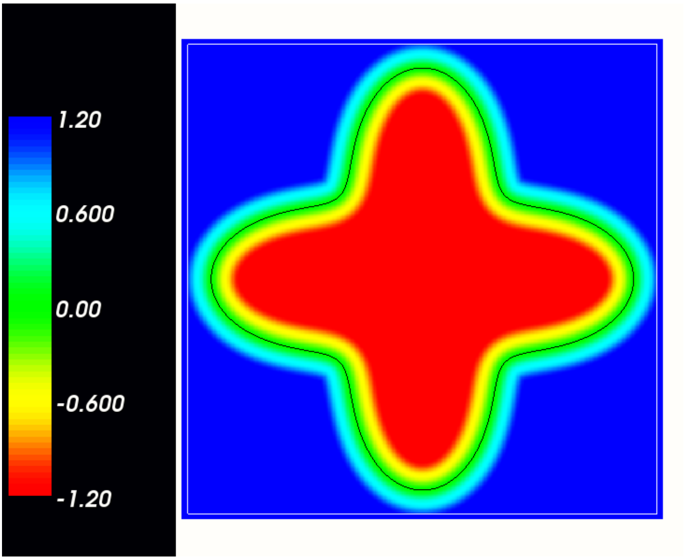


Figure 1. Curve in polar coordinates.

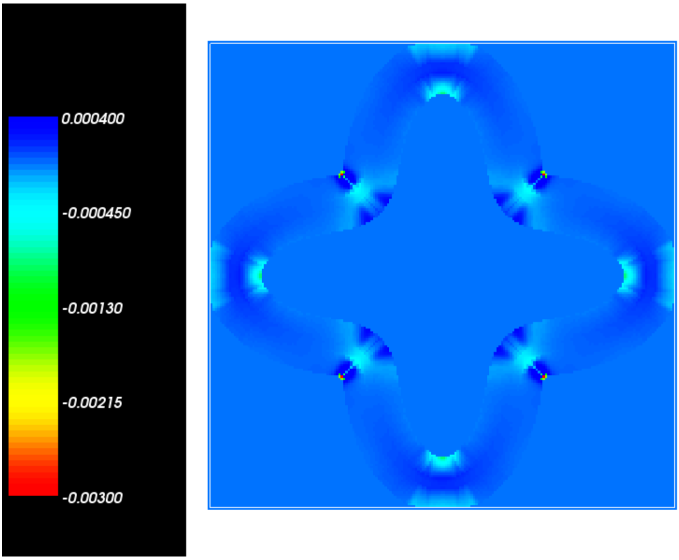


Figure 2. Error for a curve given in polar coordinates.

Our second test problem has as its zero-level set a surface of revolution. The domain has a lower left corner with coordinates $(-10, -10, -10)$ and an upper right corner with coordinates $(10, 10, 10)$. The surface is centered at $(0, 0, 0)$ and obtained by rotating the function $r = 2\cos 2\theta + 7$ around the y-axis. The initial bandwidth is approximately six grid cells wide at all resolutions. The final bandwidth is 1.5. Calculations were performed on grids with $h = \frac{1}{100}$, $\frac{1}{200}$, and $\frac{1}{400}$. Richardson error estimation was used to calculate the results presented in Table 2. Slices of the error are presented in Figure 3.

Our next example uses as an implicit function whose zero set is the surface of a cube. In this case, to test the robustness of the algorithm we initialized the annular region to the wrong weak solution of the signed distance function equation. In particular, where the characteristics diverge we do not round the corners in the initial narrow band. Nonetheless our algorithm extends this initial data to a distance function.

In this example, the initial band has a diameter of about four grid cells at the coarse resolution. The final bandwidth is about two and one half times the diameter

L^1 norm	rate	L^2 norm	rate	L^∞ norm	rate
7.0725 e-01		1.5893 e-02		7.2842e-04	
1.2275 e-01	2.52	3.0446 e-03	2.38	1.813 e-04	2.00

Table 2. Solution error for surface of revolution: $h = \frac{1}{100}$ and $\frac{1}{200}$.

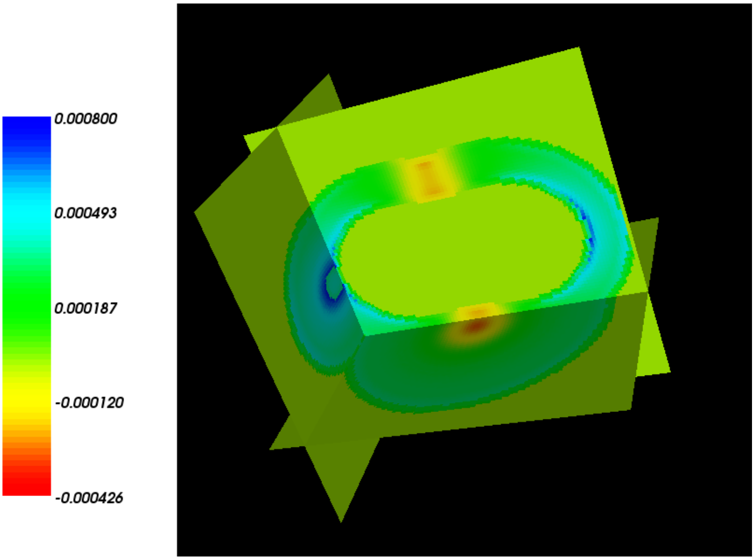


Figure 3. Slices of the error for a surface of revolution.

of the initial band. Since the only error occurs in places where the gradient is discontinuous, we present the max norm of the error in Table 3.

Our final example uses an implicit function generated by taking the union of parallelepipeds. The zero-set is in the shape of a cube whose corners are removed. Two-dimensional slices are in the shape of a cross. This example tests cases where characteristics meet at a corner as well cases where the characteristics diverge at a corner.

In this problem the domain has a lower left corner with coordinates (0, 0) and an upper right corner with coordinates (1, 1). The initial band is approximately six cells in diameter at all resolutions. The final bandwidth is 0.15. Since the errors only occur in places where the gradient is discontinuous, we present the max norm of the error in Table 4. The error is in Figure 4. Three isosurfaces, including the zero level set, are presented in Figures 5–7.

L^∞ norm	rate
0.00493	
0.00260	0.92
0.0013	1.0

Table 3. Solution error for distance to a cube: $h = \frac{1}{50}$, $\frac{1}{100}$, and $\frac{1}{200}$.

L^∞ norm	rate
.00120	
.000580	1.05

Table 4. Solution error for distance to a union of parallelepipeds: $h = \frac{1}{50}, \frac{1}{100}$, and $\frac{1}{200}$.

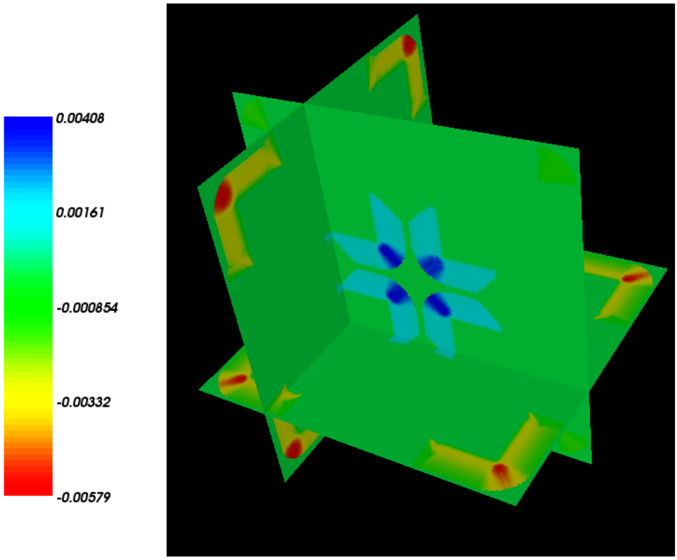


Figure 4. Slices of the error for a union of parallelepipeds.

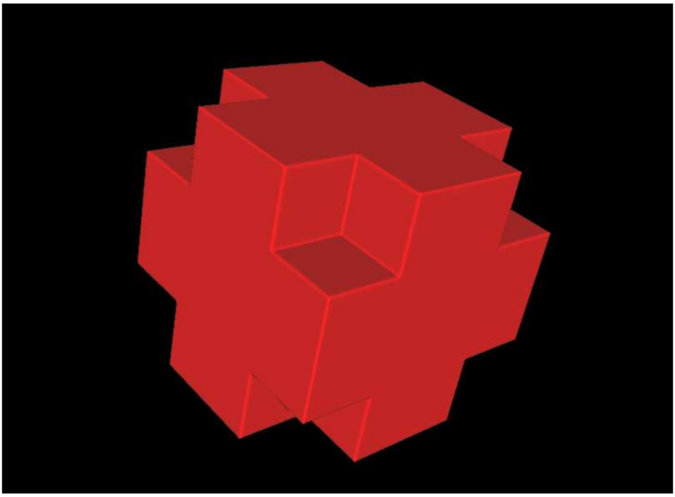


Figure 5. The zero isosurface of the union of parallelepipeds.

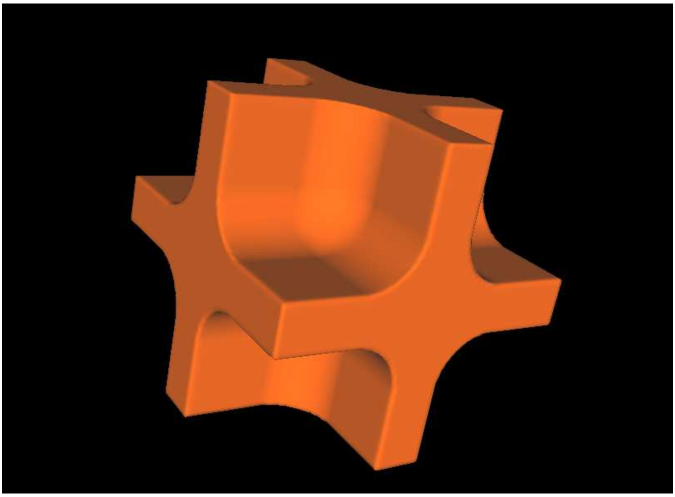


Figure 6. An interior isosurface (at a distance $= -0.12$ from the zero set) of the union of parallelepipeds.

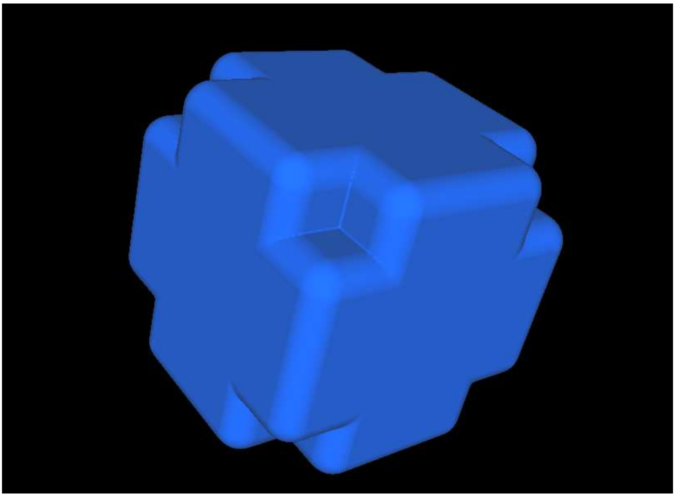


Figure 7. An exterior isosurface (at a distance $= 0.12$ from the zero set) of the union of parallelepipeds.

5. Conclusion

We have described a numerical method for solving the signed distance function equation that is second-order accurate at points whose domain of dependence includes no singularities, which is useful for second-order accurate volume-of-fluid discretizations. A salient feature of our algorithm is the hybridization of a high-order and low-order method, where the choice of hybridization coefficient is based on a local curvature calculation. The resulting calculation appears to provide solutions that satisfy the entropy condition, correctly distinguishing between the two directions of propagation from kinks in the original surface. In addition, we use a marching method that is a good match for adaptive and parallel implementation based on patch-based domain decomposition, which is the software framework typically used for high-performance implementations of block-structured adaptive grid methods.

Our future work will focus on tracking moving fronts in hyperbolic problems. In these problems, the motion of the interface naturally decomposes into advection by a vector velocity combined with motion of the interface normal to itself at a known scalar speed. The importance of the signed distance function equation may be observed in the special case where the vector velocity is zero and the scalar speed is spatially constant. In this context, a method of solving the Hamilton–Jacobi equation reduces to a method for computing the signed distance function, up to a relabeling of contours, which leads to the conclusion that numerical methods for Hamilton–Jacobi can be no more accurate than the associated solution to the signed distance function equation. Considering the general front-tracking problem, one may begin by extending velocities and scalar speeds in the normal direction off the interface by solving the transport equation, as was done in [1]. Established algorithms for advection may be employed for the velocity component of the motion, while an algorithm for solving the signed distance function equation may be employed for motion given by scalar speeds.

References

[1] D. Adalsteinsson and J. A. Sethian, *The fast construction of extension velocities in level set methods*, J. Comput. Phys. **148** (1999), no. 1, 2–22. MR 99j:65189 Zbl 0919.65074

[2] D. Adalsteinsson and J. A. Sethian, *A fast level set method for propagating interfaces*, J. Comput. Phys. **118** (1995), no. 2, 269–277. MR 96a:65154 Zbl 0823.65137

[3] D. L. Chopp, *Some improvements of the fast marching method*, SIAM J. Sci. Comput. **23** (2001), no. 1, 230–244. MR 2002h:65012 Zbl 0991.65105

[4] P. Colella, *Volume-of-fluid methods for partial differential equations*, Proceedings of an International Conference on Godunov methods: Theory and Applications (E. F. Toro, ed.), Kluwer, New York, 2001, pp. 161–177. MR 1963590 Zbl 0989.65118

[5] P. Colella, D. T. Graves, B. J. Keen, and D. Modiano, *A Cartesian grid embedded boundary method for hyperbolic conservation laws*, J. Comput. Phys. **211** (2006), no. 1, 347–366. MR 2006i:65142 Zbl 1120.65324

[6] R. Dial, *Algorithm 360: Shortest path forest with topological ordering*, CACM **12** (1969), 632–633.

[7] J. Helmsen, E. G. Puckett, P. Colella, and M. Dorr, *Two new methods for simulating photolithography development*, Optical Microlithography IX (G. E. Fuller, ed.), SPIE Conference Proceedings, no. 2726, 1996, pp. 503–555.

[8] S. Kim, *An $O(N)$ level set method for eikonal equations*, SIAM J. Sci. Comput. **22** (2001), no. 6, 2178–2193. MR 2002h:65126 Zbl 0994.76080

[9] C. Min and F. Gibou, *A second order accurate level set method on non-graded adaptive Cartesian grids*, J. Comput. Phys. **225** (2007), no. 1, 300–321. MR 2008g:65128 Zbl 1122.65077

[10] G. Russo and P. Smereka, *A remark on computing distance functions*, J. Comput. Phys. **163** (2000), no. 1, 51–67. MR 2001d:65139 Zbl 0964.65089

[11] P. Schwartz, D. Adalsteinsson, P. Colella, A. Arkin, , and M. Onsum, *Numerical computation of diffusion on a surface*, Proc. Natl. Aca. Sci. **102** (2006), 11151–56.

[12] P. Schwartz, M. Barad, P. Colella, and T. Ligocki, *A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions*, J. Comput. Phys. **211** (2006), no. 2, 531–550. MR 2006e:65194 Zbl 1086.65532

[13] J. A. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proc. Nat. Acad. Sci. USA **93** (1996), no. 4, 1591–1595. MR 97c:65171 Zbl 0852.65055

[14] ———, *Fast marching methods*, SIAM Rev. **41** (1999), no. 2, 199–235. MR 2000m:65125 Zbl 0926.65106

[15] M. Sussman, P. Smereka, and S. Osher, *A level set approach for computing solutions to incompressible two-phase flow*, J. Comput. Phys. **114** (1994), 146–159. Zbl 0808.76077

Received February 5, 2008. Revised July 9, 2009.

PETER SCHWARTZ: poschwartz@lbl.gov
Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 50A-1148MS 50A-1148, Berkeley CA 94720, United States
http://seesar.lbl.gov/ANAG/staff/schwartz/index.html

PHILLIP COLELLA: PColella@lbl.gov
*Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory,
1 Cyclotron Road MS 50A-1148, Berkeley CA 94720, United States*

COMM. APP. MATH. AND COMP. SCI.
Vol. 5, No. 1, 2010

ON THE SECOND-ORDER ACCURACY OF VOLUME-OF-FLUID INTERFACE RECONSTRUCTION ALGORITHMS: CONVERGENCE IN THE MAX NORM

ELBRIDGE GERRY PUCKETT

Given a two times differentiable curve in the plane, I prove that — using only the volume fractions associated with the curve — one can construct a piecewise linear approximation that is second-order in the max norm. I derive two parameters that depend only on the grid size and the curvature of the curve, respectively. When the maximum curvature in the 3 by 3 block of cells centered on a cell through which the curve passes is less than the first parameter, the approximation in that cell will be second-order. Conversely, if the grid size in this block is greater than the second parameter, the approximation in the center cell can be less than second-order. Thus, this parameter provides an a priori test for when the interface is *under-resolved*, so that when the interface reconstruction method is coupled to an adaptive mesh refinement algorithm, this parameter may be used to determine when to *locally* increase the resolution of the grid.

1. Introduction

In this article I study the *interface reconstruction problem* for a volume-of-fluid method in two space dimensions. Let $\Omega \in R^2$ denote a simply connected domain and let $\mathbf{z}(s) = (x(s), y(s))$, where s is arc length, denote a curve in Ω. The *interface reconstruction problem* is to compute an approximation $\tilde{\mathbf{z}}(s)$ to $\mathbf{z}(s)$ in Ω using only the volume fractions due to $\mathbf{z}$ on the grid. I define volume fractions and discuss this problem in more detail in Section 1.1 below.

Let L be a characteristic length of the problem. Cover Ω with a grid consisting of square cells each of side $\Delta x \leq L$ and let

$$h = \frac{\Delta x}{L} \tag{1}$$

MSC2000: 76-04, 65M06, 65M12, 76M20, 76M25.

Keywords: volume-of-fluid, piecewise linear interface reconstruction, fronts, front reconstruction, two-phase flow, multiphase systems, underresolved computations, adaptive mesh refinement, computational fluid dynamics, LVIRA, ELVIRA.

Sponsored by the U.S. Department of Energy (Mathematical, Information, and Computing Sciences Division, contracts DE-FC02-01ER25473 and DE-FG02-03ER25579).

be a *dimensionless* parameter that represents the size of a grid cell as a nondimensional quantity. Note that h is bounded above by 1. This ensures that second-order accurate methods, which have $O(h^2)$ error, will be more accurate than first-order accurate methods, which have $O(h)$ error. For the remainder of this article it will be understood that quantities such as the arc length s and the radius of curvature R are also nondimensional quantities obtained by division by L as in (1) and that the curvature κ has been nondimensionalized by dividing by $1/L$.

In this article I prove that a piecewise linear volume-of-fluid interface reconstruction method will be a second-order accurate approximation to the exact interface $\mathbf{z}(s) = (x(s), y(s))$ *in the max norm* provided the following four conditions hold:

I. The interface $\mathbf{z}$ is two times continuously differentiable: $\mathbf{z}(s) \in C^2\,(\Omega)$.

II. The maximum value

$$\kappa_{\max} = \max_s |\kappa(s)| \tag{2}$$

of the curvature $\kappa(s)$ of $\mathbf{z}(s)$ satisfies[1]

$$\kappa_{\max} \le C_\kappa \equiv \min\left\{C_h\, h^{-1},\, \left(\sqrt{h}\right)^{-1}\right\}, \tag{3}$$

where C_h is a constant that is independent of h and is defined by

$$C_h \equiv \frac{\sqrt{2}-1}{4\sqrt{3}}. \tag{4}$$

III. In each cell C_{ij} that contains a portion of the interface, the slope m_{ij} of the piecewise linear approximation

$$\tilde{g}_{ij}(x) = m_{ij}\, x \,+\, b_{ij} \tag{5}$$

to the interface in that cell is given by

$$m_{ij} = \frac{S_{i+\alpha} - S_{i+\beta}}{\alpha - \beta} \qquad \text{for } \alpha, \beta = 1, 0, -1 \text{ with } \alpha \ne \beta, \tag{6}$$

where $S_{i+\alpha}$ and $S_{i+\beta}$ denote two distinct *column sums* of volume fractions from the 3×3 block of cells B_{ij} surrounding the cell C_{ij}.[2] The column sums S_{i-1}, S_i, and S_{i+1} are defined and described in more detail in Section 1.3.

IV. The column sums $S_{i+\alpha}$ and $S_{i+\beta}$ in (6) are sufficiently accurate that the slope m_{ij} defined in (6) is a first-order accurate approximation to $g'(x_c)$, where x_c is the center of the bottom edge of the cell C_{ij}.

[1] It is only necessary that the maximum curvature of the interface satisfy this condition in a neighborhood of the cell C_{ij} in which one wishes to reconstruct the interface. For example, in the 3×3 block of cells B_{ij} centered on C_{ij}.

[2] I will usually omit the subscript i, j when writing the piecewise linear approximation $\tilde{g}$ defined in (5) and simply write $\tilde{g}(x)$ instead of $\tilde{g}_{ij}(x)$. Similarly, when no confusion is likely to arise, I will drop the subscript i, j from the slope m and the y-intercept b and simply write $\tilde{g}(x) = mx + b$.

Section 3 is devoted to proving that if condition (3) above is satisfied, one can always find an orientation of the 3×3 block of cells (say, after rotating by a multiple of 90 degrees) so that there are two column sums $S_{i+\alpha}$ and $S_{i+\beta}$, *both in the same orientation of the* 3×3 *block*, satisfying the condition in item IV above. Note that here I do not provide an algorithm for determining which orientation of the 3×3 block of cells is the correct one to use or, given a correct orientation, how to find the two column sums to use in (6). What I do prove is that if the interface satisfies Equation (3), then one can find an orientation of the 3×3 block of cells that has two distinct column sums $S_{i+\alpha}$ and $S_{i+\beta}$ such that the slope m_{ij} obtained in (6) is a first-order accurate approximation to $g'(x_c)$ and hence, $\tilde{g}$ is a second-order accurate approximation to g in the max norm as illustrated in Figure 1.[3]

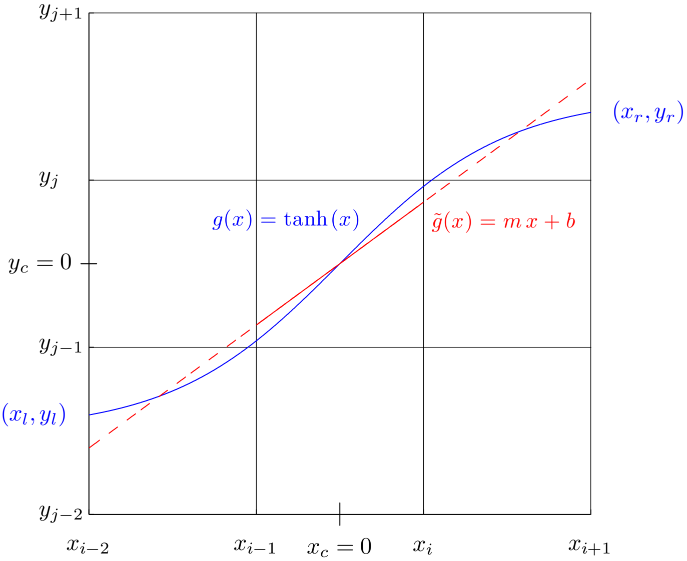


Figure 1. In this example the interface is $g(x) = \tanh x$. All three column sums are exact (in the sense of Section 1.3), but for the inverse function $x = g^{-1}(y)$ only the center column sum is exact. Also plotted is the linear approximation $\tilde{g}(x) = mx + b$ in the center cell produced by the volume-of-fluid interface reconstruction algorithm when the slope m is chosen as half the difference between the first and third column sums. The main result of this paper is that $|g(x) - \tilde{g}(x)| \leq Ch^2$ for all $x \in [x_{i-1}, x_i]$ provided that the slope m is defined in the manner described in Section 1.3.

[3] In this particular example all three of the column sums S_{i-1}, S_i and S_{i+1} are exact. Consequently, Theorem 23 in Section 4 implies any two of them can be used in (6) and that the resulting slope $m = \tilde{g}'(x_c)$ is a first-order accurate approximation to $g'(x_c)$, regardless of whether one chooses the slope to be $m = (S_i - S_{i-1})$, $m = (S_{i+1} - S_{i-1})/2$, or $m = (S_{i+1} - S_i)$.

A variety of algorithms have been proposed for determining the correct column sums to use to determine the approximate slope via Equation (6). I refer the interested reader to [6; 7; 11; 14; 22; 23; 25; 37] for further information.

Finally, I would like to emphasize that the criteria in (3) provides an a priori test to determine when a given computation of the interface is *well-resolved*; namely, the computation is well-resolved whenever

$$h \leq H_{\max} = \min\{C_h\,(\kappa_{\max})^{-1}\,,\,(\kappa_{\max})^{-2}\}. \tag{7}$$

This will enable researchers who employ block structured adaptive mesh refinement to model the motion of an interface [30; 31; 33; 34] to compute an approximation to the curvature of the interface in each cell and then check to see if the conditions in (7) are satisfied in order to determine if the computation is under-resolved in that cell. Cells in which $h > H_{\max}$ are then tagged for refinement. In this regard I note that Sussman and Ohta [32] have developed second- and fourth-order accurate volume-of-fluid algorithms for computing the curvature from the volume fraction information.

1.1. ***A detailed statement of the problem.*** Suppose that I am given a simply connected computational domain $\Omega \in R^2$ that is divided into two distinct regions Ω_d and Ω_l so that $\Omega = \Omega_d \cup \Omega_l$. I will refer to Ω_d as the "dark" fluid[4] and to Ω_l as the "light" fluid. Let $\mathbf{z}(s) = (x(s), y(s))$, where s is arc length, denote the *interface* between these two fluids. Cover Ω with a uniform square grid of cells, each with side h, and let Λ_{ij} denote the fraction of dark fluid in the (i, j)-th cell. Each number Λ_{ij} satisfies $0 \leq \Lambda_{ij} \leq 1$ and is called the *volume fraction* (of dark fluid) in the (i, j)-th cell.[5] Note that

$$0 < \Lambda_{ij} < 1 \tag{8}$$

if and only if a portion of the interface $\mathbf{z}(s)$ lies in the (i, j)-th cell and that $\Lambda_{ij} = 1$ (resp. $\Lambda_{ij} = 0$) if the i, j cell only contains dark (resp. light) fluid.

In this paper I consider the following problem. Given only the collection of volume fractions Λ_{ij} in the grid covering Ω I wish to *reconstruct* $\mathbf{z}(s)$; that is, to find a piecewise linear approximation $\tilde{\mathbf{z}}$ to $\mathbf{z}$. Furthermore, the approximate interface $\tilde{\mathbf{z}}$ must have the property that the volume fractions $\tilde{\Lambda}_{ij}$ due to $\tilde{\mathbf{z}}$ are identical to the

[4] Although these algorithms have historically been known as "volume-of-fluid" methods, they are frequently used to model the interface between any two materials, including gases, liquids, solids and any combination thereof [8; 16; 17; 18]. However, when analyzing the method, the convention is to refer to the two materials as fluids.

[5] Even though in two dimensions Λ_{ij} is technically an area fraction, the convention is to refer to it as a *volume* fraction.

original volume fractions Λ_{ij}; that is,

$$\tilde{\Lambda}_{ij} = \Lambda_{ij} \quad \text{for all cells } C_{ij}. \tag{9}$$

An algorithm for finding such an approximation is known as a *volume-of-fluid interface reconstruction method*. The property that $\tilde{\Lambda}_{ij} = \Lambda_{ij}$ is the principal feature that distinguishes volume-of-fluid interface reconstruction methods from other interface reconstruction methods. It ensures that the computational value of the total volume of each fluid is exact. In other words, all volume-of-fluid interface reconstruction methods are conservative in that they conserve the volume of each material in the computation. When the underlying numerical method is a conservative finite difference method this can be essential since, for example, in order to obtain the correct shock speed it is necessary for all of the conserved quantities to be conserved by the underlying numerical method; for example, see [5; 17; 18; 26]. More generally, a necessary condition for the numerical method to converge to the correct weak solution of the underlying partial differential equation (PDE) is that all of the quantities that are conserved in the PDE must be conserved by the numerical method [15].

Volume-of-fluid methods have been used by researchers to track material interfaces since at least the early 1970s (see [20; 21], for example), and a variety of such algorithms have been developed for modeling everything from flame propagation [3] to curvature and solidification [4]. In particular, the problem of developing high-order accurate volume-of-fluid methods for modeling the curvature and surface tension of an interface has received much attention [1; 2; 4; 10; 13; 24]. Volume-of-fluid methods were among the first interface tracking algorithms to be implemented in codes originally developed at the U.S. National Laboratories and subsequently released to the general public which are capable of tracking fluid interfaces in a variety of complex fluid flow problems [9; 12; 19; 35; 36]).

In this paper I do not consider the related problem of approximating the movement of the interface in time, for which one would use a *volume-of-fluid advection algorithm*. See [23; 27; 28] for a detailed description and analysis of several such algorithms. In the present paper I only consider the accuracy that one can obtain when using a volume-of-fluid interface reconstruction algorithm to approximate a given *stationary* interface $\mathbf{z}(s)$.

1.2. ***Basic assumptions and definitions.*** Unless explicitly stated otherwise, I will always assume that the *exact* interface $\mathbf{z}(s) = (x(s), y(s))$ is twice continuously differentiable: $\mathbf{z} \in C^2(\Omega)$. In particular, the derivatives $\dot{x}(s)$, $\dot{y}(s)$, $\ddot{x}(s)$ and $\ddot{y}(s)$ exist and are continuous. I also assume that the curvature $\kappa(s)$ of the interface $\mathbf{z}(s)$ is bounded in Ω, so that there always exists a constant $\kappa_{\max}$ independent of s such that (2) holds.

By the *center cell* C_{ij} I mean the square with side h that contains a portion of the interface $\mathbf{z}(s) = (x(s), y(s))$ for s in some interval, say $s \in (s_l, s_r)$. In what follows I will consider the 3×3 block of square cells B_{ij} — each with side h, surrounding the center cell as shown, for example, in Figure 1. Unless I note otherwise, I will denote the coordinates of the vertical edges of the cells in the 3×3 block B_{ij} centered on the cell C_{ij} by x_{i-2}, x_{i-1}, x_i and x_{i+1} and the horizontal edges of the cells in B_{ij} by y_{j-2}, y_{j-1}, y_j, y_{j+1} as shown, for example, in Figure 1. It will always be the case that

$$x_{i+1} - x_i = h, \qquad x_i - x_{i-1} = h,$$
$$y_{j+1} - y_j = h, \qquad y_j - y_{j-1} = h,$$

and so on, where h is the (nondimensional) grid size.

1.3. ***The column sums.*** The volume fraction Λ_{ij} in the (i, j)-th cell C_{ij} is a nondimensional way of storing the volume of dark fluid in that cell. Consider the column consisting of C_{ij} and the cells immediately above and below C_{ij}. The *column sum*

$$S_i \equiv \sum_{j'=j-1}^{j+1} \Lambda_{ij'}$$

is a nondimensional way of storing the total volume of dark fluid in those three cells. In order to approximate the portion of the interface $g(x)$ lying in the (i, j)-th cell C_{ij}, I will use the three column sums in the 3×3 block of cells B_{ij} that have C_{ij} in its center to compute the slope m of the piecewise linear approximation $\tilde{g}(x)$ to $g(x)$ (for example, see Figure 1). I use S_{i-1} to denote the column sum to the left of S_i and S_{i+1} to denote the column sum to the right of S_i, so that

$$S_{i-1} \equiv \sum_{j'=j-1}^{j+1} \Lambda_{i-1,j'}, \quad S_{i+1} \equiv \sum_{j'=j-1}^{j+1} \Lambda_{i+1,j'}. \tag{10}$$

Now consider an arbitrary column consisting of three cells with left edge $x = x_i$ and right edge $x = x_{i+1}$. Furthermore, assume that the interface can be written as a function $y = g(x)$ on the interval $[x_i, x_{i+1}]$. Assume also that the interface enters the column through its left edge and exits the column through its right edge and does not cross the top or bottom edges of the column, as is the case with all three columns in the example shown in Figure 1. Then the total volume of dark fluid that occupies the three cells in this particular column and lies below the interface $g(x)$ is equal to the integral of g over the interval $[x_i, x_{i+1}]$. This leads to the following relationship between the column sum and the *normalized*[6] volume of dark fluid in

[6] The normalized volume is the nondimensional quantity obtained by dividing the integral of $g(x)$ over the interval $[x_i, x_{i+1}]$ by h^2.

the column:

$$S_i \equiv \sum_{j'=j-1}^{j+1} \Lambda_{ij'} = \frac{1}{h^2} \int_{x_i}^{x_{i+1}} \big(g(x) - y_{j-2}\, h \big)\, dx. \tag{11}$$

I will use the phrase *the i-th column sum S_i is exact* whenever (11) holds, and I will refer to integrals such as the one on the right in (11) as *the normalized integral of g* in that column.

Given the 3×3 block of cells surrounding a cell C_{ij} that contains a portion $y = g(x)$ of the interface, most of the important results in this paper are based on how well the column sums S_{i-1}, S_i and S_{i+1} approximate the normalized integral of g in that particular column. This is because the slope m_{ij} of the piecewise linear approximation to g in C_{ij} will be the divided difference of two of these column sums; that is, m_{ij} is chosen to be one of the three quantities

$$m^l_{ij} = S_i - S_{i-1}, \quad m^c_{ij} = \tfrac{1}{2}(S_{i+1} - S_{i-1}), \quad m^r_{ij} = S_{i+1} - S_i. \tag{12}$$

In particular, if two of the column sums $S_{i+\alpha}$ and $S_{i+\beta}$ where $\alpha, \beta = 1, 0, -1$ and $\alpha \neq \beta$ are exact, then the slope

$$m_{ij} = \frac{\big(S_{i+\alpha} - S_{i+\beta}\big)}{(\alpha - \beta)} \tag{13}$$

will produce a piecewise linear approximation $\tilde{g}(x)$ to the portion of the interface $g(x)$ in C_{ij} that is second-order accurate in the max norm as shown, for example, in Figure 1.

In order to see why this will be the most accurate choice for the approximate slope m_{ij}, consider the case when the block B_{ij} has two exact column sums as shown in Figure 2. In this example the interface is a line $g(x) = mx + b$. In this particular orientation of the 3×3 block of cells g has two exact column sums; namely, the sums in the first and second columns. It is easy to check that

$$m = \frac{1}{h^2} \int_{x_{i-1}}^{x_i} (g(x) - y_{j-2} h)\, dx \\ - \frac{1}{h^2} \int_{x_{i-2}}^{x_{i-1}} (g(x) - y_{j-2} h)\, dx = (S_i - S_{i-1}) = m^l_{ij},$$

where S_i denotes the column sum associated with the interval $[x_{i-1}, x_i]$ and S_{i-1} denotes the column sum associated with the interval $[x_{i-2}, x_{i-1}]$.

In this example, the divided difference m^l_{ij} of the column sums S_{i-1} and S_i is exactly equal to the slope m of the exact interface. It is *always* the case that when the exact interface is a line one can find an orientation of the 3×3 block of cells such that at least one of the divided differences of the column sums in (12) is exact.

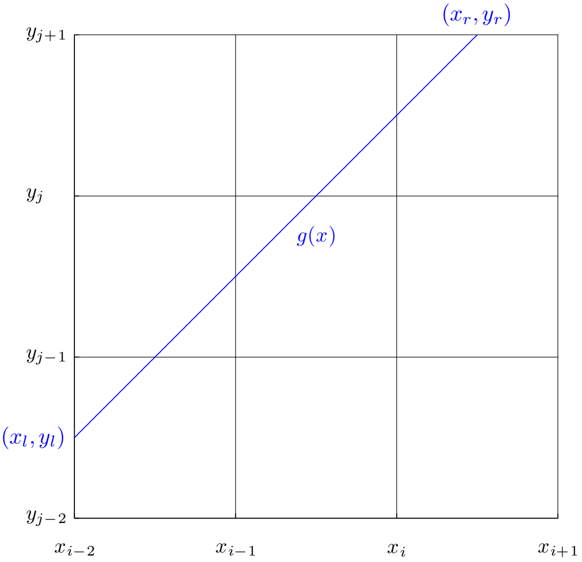


Figure 2. Here the interface is a line, $g(x) = mx + b$, having two exact column sums (those in the first and second columns). The slope m^l_{ij} from (12) is then exactly equal to the slope m of the interface: $m^l_{ij} = m$. Whenever the exact interface is a line, one can find an orientation of the 3×3 block of cells such that at least one of the divided differences of the column sums in (12) is exact.

For example, in the case shown in Figure 2 one could rotate the 3×3 block of cells 90 degrees clockwise and in this orientation the correct slope to use when forming the piecewise linear approximation $\tilde{g}(x) = m_{ij} + b_{ij}$ would be $m_{ij} = m^r_{ij}$, which again would be exactly equal to the slope m of the exact interface.

However, as I will show in Section 3, there are some instances in which the interface satisfies (3) but the center column sum S_i is not exact. Much of the work in Section 3 is devoted to showing that when the interface satisfies (3), the center column sum S_i are exact to $O(h)$:

$$\frac{1}{h^2}\int_{x_i}^{x_{i+1}} (g(x) - y_{j-2}h)\,dx - S_i = Ch,$$

where $C > 0$ is a constant that is independent of h. Then, in Section 4, I prove that this is sufficient to still obtain second-order accuracy in the max norm.

I am now ready to finish the description of the volume-of-fluid interface reconstruction algorithms that I study in this article. Given an arbitrary interface $\mathbf{z}$ in the domain Ω, I choose an orientation of the 3×3 block of cells such that at least two of the column sums are sufficiently accurate that one of the divided differences in

(12) satisfies

$$|m_{ij} - g'(x_c)| \leq Ch, \tag{14}$$

where C is a constant that is independent of h. In this article I prove that, provided the condition in (3) is satisfied, it is possible to find such an orientation.

1.4. ***A brief overview of the structure of this article.*** In the next section I begin by proving several lemmas that lead to Theorem 6, which states that if

$$h \ \leq \ C_h \, (\kappa_{\max})^{-1} \tag{15}$$

where C_h is defined in (4), then the interface can be written as a function of one of the coordinate variables in terms of the other on an interval $[a, b]$ with $|b-a\,| \geq 4\,h$. This ensures that, given a cell C_{ij} that contains a portion of the interface, I can always find a 3 × 3 block of cells centered on the cell C_{ij} in which I can write the interface as a function of one of the variables in terms of the other; for example, $y = g(x)$. To achieve this, it may be necessary to rotate the 3 × 3 block of cells centered on C_{ij} by 90, 180, or 270 degrees and/or reflect the coordinates about one of the coordinate axes: $x \to -x$ or $y \to -y$. No other coordinate transformations besides one of these three rotations and a possible reversal of one or both of the variables $x \to -x$ and/or $y \to -y$ are required in order for the algorithms studied in this article to converge to the exact interface as $h \to 0$. Furthermore, these coordinate transformations are only used to determine a first-order accurate approximation to the slope of the tangent to the interface $\mathbf{z}$ in the current cell of interest, or equivalently, a first-order accurate approximation m to $g'(x_c)$ in the center cell, as shown, for example, in Figure 1. The grid covering the domain Ω always remains the same.

In particular, if one is using the interface reconstruction algorithm as part of a numerical method to solve a more complex problem than the one posed here (for example, the movement of a fluid interface where the fluid flow is a solution of the Euler or Navier–Stokes equations), it is not necessary to perform these coordinate transformations on the underlying numerical fluid flow solver. Therefore, unless noted otherwise, in what follows I will always write $y = g(x)$ and denote the coordinates of the edges of the cells in the 3 × 3 block by $x = x_{i-2}, x_{i-1}, x_i, x_{i+1}$ and $y = y_{j-2}, y_{j-1}, y_j, y_{j+1}$, it being implicitly understood that a transformation of the coordinate system as described above may have been performed in order for this representation of the interface to be valid, and that I may have interchanged the names of the variables x and y in order to write the interface as $y = g(x)$.

In Section 2 I will also prove that in the (possibly transformed) coordinates the function $y = g(x)$ that represents the interface satisfies

$$|g'(x)| \leq \sqrt{3}, \quad \max_x |g''(x)| \leq 8\kappa_{\max}. \tag{16}$$

These inequalities are a part of Theorem 6. I use these bounds to prove several of the results in Sections 3 and 4.

In Section 3 I prove that if h satisfies

$$h \le \max\{C_h(\kappa_{\max})^{-1}, (\kappa_{\max})^{-2}\},$$

then, using one of the transformations described above, I can find a coordinate frame in which there are at least two columns with column sums $S_{i+\alpha}$ and $S_{i+\beta}$ in the 3×3 block of cells B_{ij} centered on the cell C_{ij} which contains the portion of the interface of interest, such that their divided difference,

$$m_{ij} = \frac{(S_{i+\alpha} - S_{i+\beta})}{(\alpha - \beta)} \quad \text{for } \alpha, \beta = -1, 0, 1 \text{ and } \alpha \ne \beta,$$

satisfies (14).

In Section 4 I use this result to prove Theorem 24, which is the main result of this paper. Namely that $\tilde{g}(x)$ is a second-order accurate approximation to $g(x)$ in I_i in the max norm:

$$|\, g(x) - \tilde{g}_{ij}(x) \,| \le \left(\tfrac{50}{3}\kappa_{\max} + C_S\right)h^2 \quad \text{for all } x \in I_i = [x_{i-1}, x_i].$$

Here C_S is a constant that is independent of h and the approximate interface $\tilde{g}_{ij}(x)$ is being constructed in the center cell $C_{ij} = [x_{i-1}, x_i] \times [y_{j-1}, y_j]$ of the 3×3 block of cells B_{ij} that contains the portion of the interface that is of interest, as shown, for example, in Figure 1. A corollary of this result is that when the size of the computational grid h is too large

$$h \ge H_{\max}, \tag{17}$$

where $H_{\max}$ is defined in (7), then the convergence rate *may* be less than second-order. Thus, (17) may be used as a criterion for predicting when the computation of the interface may be under-resolved.

2. The first constraint on the grid size *h*

The principle purpose of this section is to show that for a given interface $\mathbf{z}(s)$ with a maximum curvature $\kappa_{\max}$ there exists a value of the grid size $h = h_{\max}$ such that the interface can be written as a function of one of the coordinate variables in terms of the other in any given 3×3 block of cells B_{ij} of side $h \le h_{\max}$ centered on a cell C_{ij} that contains a portion of the interface. The main result in this section is Theorem 6, in which I derive the constraint

$$h \le h_{\max} \equiv C_h(\kappa_{\max})^{-1}, \tag{18}$$

where C_h is the constant defined in (4). I also prove that in the same 3×3 block of cells B_{ij} centered on the cell C_{ij} the bounds in (16) hold.

The constraint in (18) is not sufficient to guarantee that the volume-of-fluid interface reconstruction algorithm will be second-order accurate in the limit as $h \to 0$. In Section 3 below, I will show that this requires a more stringent constraint on h, namely

$$h \leq (\kappa_{\max})^{-2}.$$

Suppose that I am interested in a neighborhood of the point $\mathbf{z}(s_0) = (x(s_0), y(s_0)) \equiv (x_0, y_0)$ on the interface[7] and at this point I have

$$\dot{x}^2(s_0) \geq \tfrac{1}{2}. \tag{19}$$

I will now show that in some neighborhood of the point (x_0, y_0) I can represent the interface $(x(s), y(s))$ as the single valued function $y(s) = g(x(s))$. Then, in Lemma 4 I will answer the question: *Over how large an interval* $[x_l, x_r]$ *where* $x_l < x_0 = x(s_0) < x_r$ *is this representation of the interface valid*? I will now proceed to address this question.

Let $s_l < s_r$[8] chosen such that s_l is the largest number less than s_0 and s_r is the smallest number greater than s_0 such that

$$\dot{x}^2(s) \geq \tfrac{1}{4} \quad \text{for all } s \in [s_l, s_r]. \tag{20}$$

Given that at the point $\mathbf{z}(s_0)$ the inequality in (19) holds there are two possibilities for the point $\mathbf{z}(s_l)$ (resp. $\mathbf{z}(s_r)$).

(1) At the point $\mathbf{z}(s_l)$ (resp. $\mathbf{z}(s_r)$) I have

$$\dot{x}^2(s_l) = \tfrac{1}{4} \quad \bigl(\text{resp. } \dot{x}^2(s_r) = \tfrac{1}{4}\bigr). \tag{21}$$

In this case I can estimate the size of the interval $[x_l, x_0]$ (resp. $[x_0, x_r]$) over which I can represent the interface as a function of one of the coordinate variables in terms of the other, say $y = g(x)$, and bound the first and second derivatives of this function. All of these estimates will be in terms of one quantity; namely, $\kappa_{\max}$, the maximum curvature of the interface.

(2) For all $s < s_0$ (resp. $s > s_0$) I have

$$\dot{x}^2(s) > \tfrac{1}{4},$$

and at some point $\mathbf{z}(s_l)$ (resp. $\mathbf{z}(s_r)$) the interface $\mathbf{z}(s)$ intersects the boundary of the computational domain Ω. In this case the bound in (2) holds from the point x_0 up to the point x_l (resp. x_r) on the boundary. In this case, I can

[7] In this section, and this section only, x_0 and y_0 denote a point on the interface $\mathbf{z}(s_0) = (x(s_0), y(s_0)) \equiv (x_0, y_0)$ rather than the location of one of the grid lines in the 3×3 block of cells.

[8] Without loss of generality I can assume that $x(s)$ increases with increasing s, since otherwise the change of variables $s \to -s$ is also a parametrization of the interface by arc length for which $x(s)$ increases with increasing s.

express the interface as a function such as $y = g(x)$ from $x_0 \in [-h/2, h/2]$ all the way to the boundary on the left (resp. right); that is, in the interval $[x_l, x_0]$ (resp. in the interval $[x_0, x_r]$).

Note that since I have assumed that the domain Ω is bounded and that either the interface enters and exits the domain across the boundary or it is a closed curve in Ω, these are the only two possibilities. For if the interface is a closed curve, such as a circle, it must be the case that eventually $\dot{x}(s) \to 0$.

In either case, there is an interval $[x_l, x_r]$ upon which I can express the interface as a function $y = g(x)$ and upon which all of the bounds that I prove below will hold. The only difference between cases (1) and (2) above is that in case (2) one or both of the points x_l and x_r lie on the boundary of the domain.

Since, for the purposes of the proving the lemmas and theorems below, I do not know a priori the distance from x_0 to the boundary, for the remainder of this section I will assume that case (1) above holds and proceed to estimate the size of the intervals $[x_l, x_0]$ and $[x_0, x_r]$ in terms of the bound $\kappa_{\max}$ on the curvature of the interface. This will allow me to explicitly estimate the size of the interval $[x_l, x_r]$ containing the point of interest $(x_0, y_0) \equiv (x(s_0), y(s_0))$ over which I can express the interface as a function $y = g(x)$ and prove explicit bounds on the first and second derivatives of g.

Remark 1. If the inequality in (19) fails to hold at the point $\mathbf{z}(s_0)$ at which I wish to reconstruct the interface, then $\dot{y}^2(s_0) \geq \frac{1}{2}$ instead, since s is arc length and hence $\dot{x}^2(s) + \dot{y}^2(s) = 1$. In this case I instead choose y to be the independent variable and the same analysis will produce the same estimates throughout. Therefore, in all of what follows x will denote the independent variable, it being understood that in some cases y is the correct variable to choose.

Remark 2. The choice of the constant $\frac{1}{2}$ in (19) and the constant $\frac{1}{4}$ in (21) is arbitrary. One could have chosen instead any two constants C_1 and C_2 that satisfy $C_1 > C_2 > 0$ in the proof of Lemma 3. The lemma will continue to hold, but the values of the constants C_h and $h_{\max}$ in Theorem 6 below will change. In other words, all of our results will remain true, albeit with different constants.

I begin by finding a bounds on the second derivatives $\ddot{x}(s)$ and $\ddot{y}(s)$ of the functions $x(s)$ and $y(s)$ in terms of the global bound $\kappa_{\max}$ on the curvature of the interface. I will use these bounds to estimate the size of the intervals $[x_l, x_0]$ and $[x_0, x_r]$ in terms of the intervals $[s_l, s_0]$ and $[s_0, s_r]$, respectively, in the two subsequent lemmas.

Lemma 3 (A bound on $\ddot{x}(s)$ and $\ddot{y}(s)$). *Suppose that I am given a point* $\mathbf{z}(s_0) = (x(s_0), y(s_0))$ *on the interface at which the inequality*

$$\dot{y}^2(s) \leq \tfrac{1}{2} \leq \dot{x}^2(s) \tag{22}$$

holds. Let $s_l < s_0$ be the largest number less than s_0 and $s_r > s_0$ be the smallest number greater than s_0 such that

$$\tfrac{1}{4} \le \dot{x}^2(s) \quad \bigl(\textit{and hence } \dot{y}^2(s) \le \tfrac{3}{4}\bigr) \quad \textit{for all } s \in [s_l, s_r]\,. \tag{23}$$

Then

$$|\ddot{x}(s)| \le \tfrac{\sqrt{3}}{2}\kappa_{\max} \quad \textit{for all } s \in [s_l, s_r]\,. \tag{24}$$

Similarly, if the roles of $\dot{x}(s)$ and $\dot{y}(s)$ are reversed in the inequalities in Equations (22) and (23) above, then I have

$$|\ddot{y}(s)| \le \tfrac{\sqrt{3}}{2}\kappa_{\max} \quad \textit{for all } s \in [s_l, s_r]\,. \tag{25}$$

Proof. To begin, recall that since the parameter s is arc length,

$$\dot{x}^2(s) + \dot{y}^2(s) = 1 \tag{26}$$

holds for all s, and hence the curvature $\kappa(s)$ can be written as

$$\kappa(s) = \dot{x}(s)\ddot{y}(s) - \dot{y}(s)\ddot{x}(s) \tag{27}$$

(see [29, page 555]). Differentiating (26) with respect to s I find that

$$\dot{x}(s)\ddot{x}(s) = -\dot{y}(s)\ddot{y}(s), \tag{28}$$

or equivalently

$$-\dot{x}^2(s)\ddot{x}(s) = \dot{y}(s)\dot{x}(s)\ddot{y}(s). \tag{29}$$

Multiplying (27) by $\dot{y}(s)$ I have

$$\dot{y}(s)\kappa(s) = \dot{y}(s)\dot{x}(s)\ddot{y}(s) - \dot{y}^2(s)\ddot{x}(s), \tag{30}$$

and thus, using (29) in (30), I obtain

$$\dot{y}(s)\kappa(s) = -\ddot{x}(s)(\dot{x}^2(s) + \dot{y}^2(s)) = -\ddot{x}(s). \tag{31}$$

Combining (31) and (23) I obtain the following bound on $\ddot{x}(s)$ in terms of the bound $\kappa_{\max}$ on the curvature $\kappa(s)$,

$$|\ddot{x}(s)| = |\dot{y}(s)\kappa(s)| \le |\dot{y}(s)|\kappa_{\max} \le \tfrac{\sqrt{3}}{2}\kappa_{\max}.$$

One can use an identical argument to prove the bound on $\ddot{y}(s)$ in (25). □

In the next lemma I explicitly demonstrate how the size of the intervals $[x_l, x_0]$ and $[x_0, x_r]$ depend on the size of the intervals $[s_l, s_0]$ and $[s_0, s_r]$ respectively. In the lemma after that I provide an explicit relationship between the size of the intervals $[s_l, s_0]$ and $[s_0, s_r]$ the bound $\kappa_{\max}$ in (2) on the curvature of the interface.

Lemma 4. *Let* $\mathbf{z}(s_0) = (x(s_0), y(s_0))$ *be a point on the interface at which the inequality*

$$\dot{y}^2(s_0) \le \tfrac{1}{2} \le \dot{x}^2(s_0)$$

holds, and let $s_l < s_0$ *be the greatest number less than* s_0 *and* $s_r > s_0$ *the smallest number greater than* s_0 *such that*

$$\tfrac{1}{4} \le \dot{x}^2(s) \quad \textit{for all } s \in [s_l, s_r]. \tag{32}$$

Then, letting $x_l \equiv x(s_l)$, $x_0 \equiv x(s_0)$, *and* $x_r \equiv x(s_r)$, *the following inequalities hold*:

$$\tfrac{1}{2}|s_0 - s_l| \le |x_0 - x_l| \le |s_0 - s_l|, \quad \tfrac{1}{2}|s_r - s_0| \le |x_r - x_0| \le |s_r - s_0|. \tag{33}$$

Proof. I prove that the inequalities involving s_l are true. The proof of the other pair of inequalities is identical. By the mean-value theorem I have

$$x_0 - x_l = \dot{x}(\tilde{s})(s_0 - s_l) \quad \text{for some} \quad \tilde{s} \in (s_0, s_l). \tag{34}$$

Since both (26) and (32) hold I have $\frac{1}{4} \le \dot{x}^2(s) \le 1$ for all $s \in [s_l, s_r]$, and hence

$$\tfrac{1}{2} \le |\dot{x}(s)| \le 1 \quad \text{for all } s \in [s_l, s_r]. \tag{35}$$

Combining (34) and (35) I obtain

$$\tfrac{1}{2}|s_0 - s_l| \le |x_0 - x_l| \le |s_0 - s_l|,$$

as claimed. □

But how large are the intervals $[s_l, s_0]$ and $[s_0, s_r]$ in terms of the physical coordinates x and y? The following lemma addresses this question.

Lemma 5. *Let* $\mathbf{z}(s_0) = (x(s_0), y(s_0))$ *be a point on the interface at which the inequality*

$$\dot{y}^2(s_0) \le \tfrac{1}{2} \le \dot{x}^2(s_0) \tag{36}$$

holds. If $s_l < s_0$ *is the greatest number less than* s_0 *and* $s_r > s_0$ *is the smallest number greater than* s_0 *such that*

$$\dot{x}^2(s_l) = \tfrac{1}{4} = \dot{x}^2(s_r), \tag{37}$$

then the distances $|s_r - s_0|$ *and* $|s_0 - s_l|$ *satisfy*

$$|s_0 - s_l| \ge \frac{\sqrt{2}-1}{\sqrt{3}}\,(\kappa_{\max})^{-1}, \quad |s_r - s_0| \ge \frac{\sqrt{2}-1}{\sqrt{3}}\,(\kappa_{\max})^{-1}. \tag{38}$$

Proof. I will prove the first inequality; the proof of the second is identical. Let $\dot{x}_l = \dot{x}(s_l)$ and $\dot{x}_0 = \dot{x}(s_0)$. By the mean-value theorem I have $\dot{x}_0 - \dot{x}_l = \ddot{x}(\tilde{s})(s_0 - s_l)$ for some $\tilde{s} \in (s_0, s_r)$, and hence

$$|\dot{x}_0 - \dot{x}_l| = |\ddot{x}(\tilde{s})||s_0 - s_l| \le \tfrac{\sqrt{3}}{2}|s_0 - s_l|\kappa_{\max}, \tag{39}$$

where the inequality in (39) follows from (24). Thus

$$|s_0 - s_l| \geq \tfrac{2}{\sqrt{3}} |\dot{x}_0 - \dot{x}_l| (\kappa_{\max})^{-1}. \tag{40}$$

Now from (36) and (37), I have $|\dot{x}_l| = \frac{1}{2}$ and $|\dot{x}_0| \geq \frac{1}{\sqrt{2}}$, and hence

$$|\dot{x}_0 - \dot{x}_l| \geq \frac{\sqrt{2}-1}{2}. \tag{41}$$

Combining (40) and (41) I obtain, as needed,

$$|s_0 - s_l| \geq \frac{2}{\sqrt{3}} |\dot{x}_0 - \dot{x}_l| (\kappa_{\max})^{-1} \geq \frac{\sqrt{2}-1}{\sqrt{3}} (\kappa_{\max})^{-1}, \qquad \square$$

I am now prepared to explicitly demonstrate the relationship between the maximum allowable cell size $h_{\max}$ and the bound on the curvature $\kappa_{\max}$ such that for all $h \leq h_{\max}$ the inequality in (20) holds for all x in the interval $[x_0 - 2h, x_0 + 2h]$, and hence the interface can be represented as a single-valued function $y = g(x)$ in the 3×3 block of cells B_{ij} of side h surrounding the cell C_{ij} containing the point (x_0, y_0) on the interface.

Theorem 6. *Suppose that I wish to reconstruct the interface in a neighborhood of the point* $\mathbf{z}(s_0) = (x(s_0), y(s_0))$ *and that at this point*

$$\dot{y}^2(s_0) \leq \tfrac{1}{2} \leq \dot{x}^2(s_0). \tag{42}$$

Let $s_l < s_0$ *be the greatest number less than* s_0 *and* $s_r > s_0$ *be the smallest number greater than* s_0 *such that*

$$\tfrac{1}{4} \leq \dot{x}^2(s) \quad \textit{for all } s \in [s_l, s_r]. \tag{43}$$

Let $x_0 = x(s_0)$ *and let*

$$h_{\max} = C_h (\kappa_{\max})^{-1}, \tag{44}$$

where

$$C_h \equiv \frac{\sqrt{2}-1}{4\sqrt{3}} \tag{45}$$

is the constant defined in (4)*. Then the interface can be represented as a single-valued function* $y = g(x)$ *on the interval* $[x_0 - 2\,h_{\max}, x_0 + 2\,h_{\max}]$*. Furthermore,*

$$\max_{x \in [a,b]} |g'(x)| \leq \sqrt{3} \tag{46}$$

and

$$\max_{x \in [a,b]} |g''(x)| \leq 8\kappa_{\max} \tag{47}$$

where $a = x_0 - 2\,h_{\max}$ *and* $b = x_0 + 2\,h_{\max}$.

Remark 7. As a consequence of this theorem, if the point $\mathbf{z}_0 = \mathbf{z}(s_0)$ lies in some cell C_{ij} of side $h \le h_{\max}$, then the interface can be represented as a single-valued function $y = g(x)$ throughout the 3×3 block B_{ij} of square cells of side h surrounding C_{ij} and the bounds in (46) and (47) hold throughout B_{ij}.

Remark 8. It is apparent that interchanging the roles of $x(s)$ and $y(s)$ in Lemmas 3–5 and Theorem 6 above will show that the interface can be represented as a single-valued function $x = G(y)$ throughout the 3×3 block B_{ij} of square cells of side h surrounding C_{ij} and the bounds in (46) and (47) hold throughout the B_{ij} with x replaced by y and g replaced by G.

Proof. Let $x_l = x(s_l)$ and $x_r = x(s_r)$. Since, by the implicit function theorem, the interface can be represented as a single-valued function $y = g(x)$ on any interval over which $\dot{x}^2(s) \ge \frac{1}{4} \ne 0$, it follows immediately from the assumption in (43) that the interface $\mathbf{z}(s) = (x(s), y(s))$ can be written as $(x(s), g(x(s))$ for all $s \in [s_l, s_r]$; or, equivalently, as $(x, g(x))$ for all $x \in [x_l, x_r]$.

Now I need to prove that $[x_0 - 2h_{\max}, x_0 + 2h_{\max}] \subseteq [x_l, x_r]$, or equivalently, that

$$x_l \le x_0 - 2h_{\max} \tag{48}$$

and

$$x_r \ge x_0 + 2h_{\max}. \tag{49}$$

To see that (48) holds note that (33) and (38) imply

$$|x_0 - x_l| \ge \tfrac{1}{2}|s_0 - s_l| \ge \frac{\sqrt{2}-1}{\sqrt{3}}(\kappa_{\max})^{-1} = \frac{\sqrt{2}-1}{2\sqrt{3}}(\kappa_{\max})^{-1} = 2h_{\max}.$$

Since $x_0 - x_l > 0$, Equation (48) follows immediately. The proof of (49) is nearly identical.

To see that (46) holds for $x \in [x_l, x_r]$ note that from (43) I have

$$\frac{1}{\dot{x}^2(s)} \le 4 \quad \text{for all } s \in [s_l, s_r]. \tag{50}$$

Furthermore, since s is arc length, I know that $\dot{x}^2(s) + \dot{y}^2(s) = 1$ for all s, and hence (43) also implies that

$$\dot{y}^2(s) \le \tfrac{3}{4} \quad \text{for all } s \in [s_l, s_r]. \tag{51}$$

Combining (50) and (51) yields

$$\left|g'(x(s))\right|^2 = \left|\frac{\dot{y}^2(s)}{\dot{x}^2(s)}\right| \le 3, \tag{52}$$

from which the expression in (46) follows immediately.

To see that (47) holds on the interval $[x_0 - 2h_{\max}, x_0 + 2h_{\max}]$, write the curvature of the interface $\kappa(x)$ in terms of the first and second derivatives of g [29, page 555]:

$$\kappa(x) = \frac{g''(x)}{\left(1 + g'(x)^2\right)^{3/2}}. \tag{53}$$

The inequality in (47) follows immediately from the fact that (52) holds on $x \in [x_0 - 2\,h_{\max},\; x_0 + 2\,h_{\max}]$. □

3. The accuracy of the column sums in a 3 x 3 block of cells

Notation. In this section I will often denote the edges of the 3×3 block of cells by x_0, x_1, x_2, x_3 and y_0, y_1, y_2, y_3 as shown, for example, in Figure 3, rather than x_{i-2}, x_{i-1}, x_i, x_{i+1} and y_{j-2}, y_{j-1}, y_j, y_{j+1}.

It is important to note that there is no bound of the form (3) that will ensure that the interface will always have at least two exact column sums in any of the

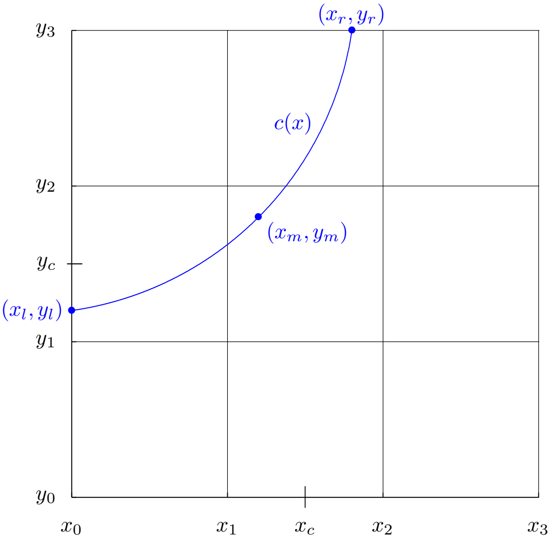


Figure 3. An example of a circular interface $c(x)$ that satisfies (3), but for which the center column sum is not exact in any of the four standard orientations of the grid. Hence, any approximation m to the slope $c'(x_c)$ of the form (13) will perforce have a nonexact column sum S_i. Theorem 15 shows that the error between the sum S_i and the normalized integral of c over the second column is $O(h)$ (that is, (3) implies that (54) holds). Theorem 23 shows that this suffices to prove $|m - c'(x_c)| = O(h)$. Finally, Theorem 24 shows that this yields an approximate interface $\tilde{g}(x)$ which is a second-order accurate approximation of $c(x)$ in the max norm.

four standard orientations of the grid. The argument is as follows. Consider the curve shown in Figure 3, where I have chosen h so that $(\sqrt{h})^{-1} \le C_h h^{-1}$. Let $0 < \epsilon < h$ be a small parameter. I can always find a circle $c(x)$[9] that passes through the three noncollinear points $(x_l, y_l) = (x_0, y_1 + \epsilon)$, $(x_m, y_m) = (x_1 + \epsilon, y_2 - \epsilon)$ and $(x_r, y_r) = (x_2 - \epsilon, y_3)$ as shown in the figure. As $\epsilon \to 0$ the arc of the circle passing through (x_l, y_l), (x_m, y_m) and (x_r, y_r) tends to the chord connecting (x_l, y_l) and (x_r, y_r), which, since the curvature of the chord is 0, implies that the radius R of the circle tends to ∞. Therefore, for some $\epsilon > 0$, the radius will satisfy $R \ge \sqrt{h}$, or equivalently, $\kappa_{\max} = R^{-1} \le (\sqrt{h})^{-1}$, and hence the circle satisfies (3). However, since by construction $y_1 < y_l$ and $x_r < x_2$, the center column sum will not be exact in any of the four standard orientations of the block B_{ij}. Consequently, if one wishes to construct an approximation to $c(x)$ based solely on the volume fraction information contained in the 3×3 block B_{ij} centered on the cell C_{ij} containing the point (x_m, y_m), the best result that one can hope for is that the center column sum S_i is exact to $O(h)$.

Much of the work in this section is devoted to showing that when cases such as the one shown in Figure 3 occur, the error between the column sum S_i and the normalized integral of the interface in that column is $O(h)$:

$$\left| S_i - \frac{1}{h^2} \int_{I_i} \big(g(x) - y_{j-2} h \big)\, dx \right| \le Ch, \tag{54}$$

where the constant $C > 0$ is independent of h. In Section 4 I prove that this is sufficient to ensure that the approximations

$$m^l_{ij} = (S_i - S_{i-1}), \quad m^r_{ij} = (S_{i+1} - S_i)$$

to $g'(x_c)$ are still first-order accurate, provided that the column sum S_{i-1} (resp. S_{i+1}) is exact. This fact is essential to the proof of Theorem 24, which is the main result of this paper; namely, that the volume-of-fluid approximation $\tilde{g}(x)$ to the interface $g(x)$ is second-order accurate in the max norm.

In this regard, I introduce the following terminology.

Definition 9. Let $C > 0$ be a constant that is independent of h and let S_i denote the column that is made up of the three cells that are centered on the cell $C_{ij} = [x_{i-1}, x_i] \times [y_{j-1}, y_j]$ in which the interface will be reconstructed. Then I will say that *the i-th column sum S_i is exact to $O(h)$* if and only if (54) holds.

The main result in this section is Theorem 10; that a *well-resolved* interface has two column sums that are exact to $O(h)$. In other words, given a function g that satisfies (3), one will always be able to find two columns whose divided difference

[9] When the exact interface is a circle, I will usually denote it by $c(x)$, as I have done in Figure 3. Otherwise, I always denote the exact interface by $g(x)$.

as defined in (12) will yield a first-order accurate approximation m to $g'(x_c)$ where $x_c = (x_1+x_2)/2$. This — together with the fact that I know the exact volume of fluid in the center cell — will allow me to construct a piecewise linear approximation $\tilde{g}(x)$ to the interface in that cell which is second-order accurate in the max norm.

I have chosen to present the results in the remainder of this section (and only in this section) in "top down" form. In other words, I state the main result first and prove it, in part, using the results of lemmas and theorems that I state and prove later in the section. I have chosen to structure the paper in this manner because I believe that this makes it much easier for the reader to follow the motivation for the various minor results that I need in order to prove the main results of the section.

3.1. ***Assumptions concerning the interface function g.*** In what follows, when I speak about the interface entering and exiting the 3×3 block of cells B_{ij}, I am only concerned with the *last* time that it enters B_{ij} before entering the center cell C_{ij} of the block B_{ij} and the *first* time that it exits B_{ij} after having exited the center column S_i of B_{ij}. As will be apparent from the material below, the condition in (3) prevents a C^2 function of x from entering B_{ij} through one of its edges, passing through the center cell C_{ij}, exiting B_{ij} and then turning around and reentering B_{ij} as shown, for example, in Figure 4. The critical assumptions are that the interface must be a C^2 function of x in some domain

$$D = [x_{i-2}, x_{i+1}] \times [y_b, y_t] \subset \Omega$$

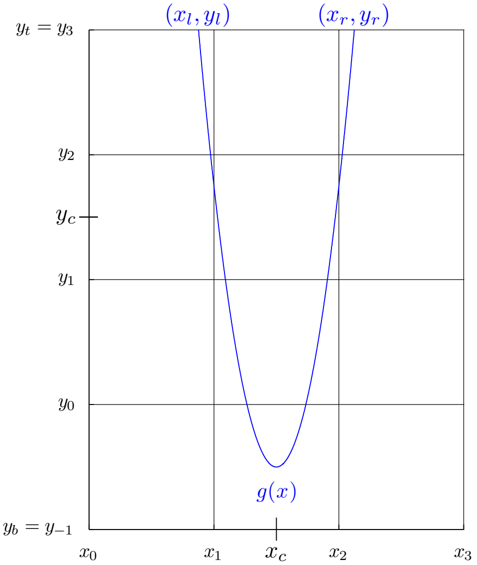


Figure 4. Here $h = 1$ and the interface is the parabola $g(x) = a(x-x_c)^2 - h/2$ with $a = 9$. The maximum curvature $\kappa_{\max} = 18$ exceeds $(\sqrt{h})^{-1} = 1$, so g does not satisfy (3). The interface enters the 3×3 block of cells B_{ij} through the top edge of the first column, passes through the center cell C_{ij}, exits B_{ij} through the bottom edge of the center column (that is, the line $y = y_0$), and then passes through B_{ij} again; the second path being symmetric to the first. In general, as $h \to 0$ the constraint $\kappa_{\max} \le (\sqrt{h})^{-1}$ on the curvature ensures that the interface does not have "hairpin" turns on the scale of the 3×3 block of cells B_{ij}. A finer grid (that is, a smaller h) is required in order to resolve curves such as the one illustrated here.

with $y_b \le y_{j-2} < y_{j+1} \le y_t$ that contains the 3×3 block B_{ij} (see Figure 4 again), and that the interface must satisfy the constraint on the curvature in (3). This precludes the interface from folding back upon itself on scales that are $O(h)$.

Theorem 10 (A well-resolved interface has two column sums that are exact to $O(h)$). *Consider the* 3×3 *block of square cells* B_{ij}, *each with side* h, *centered on the cell* C_{ij} *through which the interface* $\mathbf{z}(s)$ *passes. Assume that in some domain* $D = [x_{i-2}, x_{i+1}] \times [y_b, y_t] \subseteq \Omega$ *with* $y_b \le y_{j-2} < y_{j+1} \le y_t$ (*resp.* $D = [x_b, x_t] \times [y_{j-2}, y_{j+1}] \subseteq \Omega$ *with* $x_b \le x_{i-2} < x_{i+1} \le x_t$) *that contains the* 3×3 *block of cells* B_{ij} *the interface* $\mathbf{z}(s)$ *can be represented as a function* $y = g(x)$ (*resp.* $x = G(y)$) *with* $g \in C^2[x_{i-2}, x_{i+1}]$ (*resp.* $G \in C^2[y_{j-2}, y_{j+1}]$). *Furthermore, assume that the interface* $\mathbf{z}(s)$ *satisfies the constraint on the curvature in Equation* (3). *Then in one of the standard orientations of the grid* (*that is, rotation of the block by* 0, 90, 180, *or* 270 *degrees and/or interchanging the arc length parameter* s *with* $s' = -s$) *the interface has at least two column sums that are either exact or exact to* $O(h)$.

The remainder of Section 3 is concerned with proving Theorem 10 via a sequence of lemmas and theorems. In proving this theorem I will use symmetry arguments such as the one demonstrated in Figure 5. In the following *symmetry lemma*, I show that when the constraint on the curvature in Equation (3) holds there are only four canonical ways the interface can enter the 3×3 block of cells B_{ij}, pass through the center cell C_{ij} and then exit B_{ij}. In the remainder of the lemmas and theorems in this section I will show that, given the assumptions of Theorem 10, two of these cases are not possible and in the other two cases either there are at least two distinct column sums in B_{ij} that are exact to $O(h)$ or the particular interface configuration is not consistent with the hypotheses of Theorem 10.

The purpose of the symmetry lemma is to avoid having to prove that Theorem 10 holds for every possible way in which the interface can enter the 3×3 block of cells B_{ij}, pass through the center cell C_{ij} and then exit B_{ij}, and reduce all of these possible cases to the four canonical cases mentioned above. In the proof of the symmetry lemma, I will argue that one particular interface configuration is *equivalent* to another, say configuration 1 is equivalent to configuration 2, in the sense that the argument I use to prove Theorem 10 is true for configuration 1 can also be used to prove that the theorem is true for configuration 2. In order to see that configurations 1 and 2 are equivalent I will argue that by

(1) rotating the block B_{ij} by 90, 180, 270 degrees, and/or

(2) interchanging the arc length parameter s with $s' = -s$, and/or

(3) reflecting the block B_{ij} about one of the centerlines $x = x_c = (x_1 + x_2)/2$ or $y_c = (y_1 + y_2)/2$.

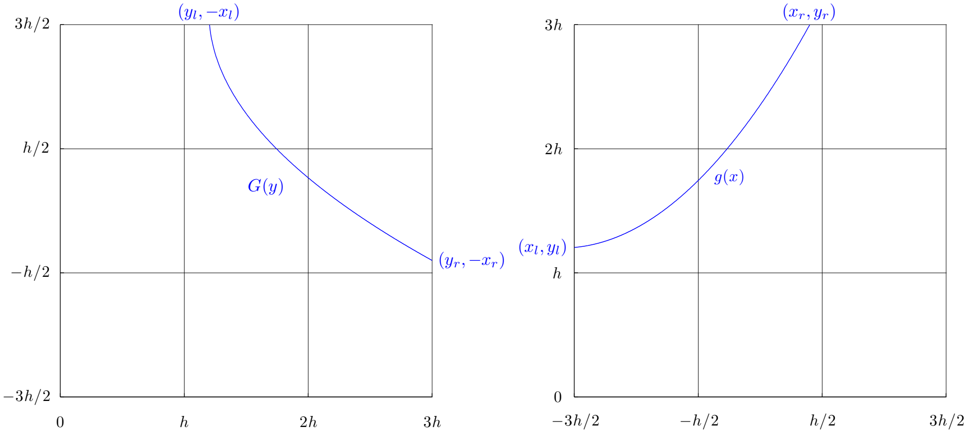


Figure 5. The same interface viewed in two different orientations. Left: In this orientation the interface, written as $-x = G(y)$, has one exact column sum (the third); it enters through the top edge of the center column and exits through the right-hand edge of B_{ij}, so the hypotheses of Theorem 15 do not apply. Right: Upon rotation of the grid clockwise by 270 degrees the interface, now described by $y = g(x)$ (g being the inverse funtion of G, which is strictly monotonic), also has one exact column sum; but here it does satisfy the hypotheses of Theorem 15, so S_i is exact to $O(h)$.

I can use the same proof for configuration 2 as for configuration 1. An example is seen in Figure 5. Note that it is not necessary to reflect the block B_{ij} about either of the centerlines $x = x_c$ or $y = y_c$ in order to determine the approximate slopes m^l_{ij}, m^c_{ij} and m^r_{ij} defined in (12). I only use reflection of the block about one of the lines $x = x_c$ or $y = y_c$ in order to simplify the proof of the symmetry lemma and hence, of Theorem 10.

Symmetry Lemma. *Assume that the hypotheses of Theorem 10 hold. Since the curvature of the interface $\mathbf{z}(s)$ is an intrinsic property of the interface, and hence does not depend on the orientation of the coordinate system that I choose to work in, I only need to prove that the conclusions of Theorem 10 hold in the following four cases*:

I. *The interface $\mathbf{z}$ enters the 3×3 block of cells across its top edge, passes through the center cell and exits the 3×3 block of cells across its top edge.*

II. *The interface $\mathbf{z}$ enters the 3×3 block of cells across its left edge, passes through the center cell and exits the 3×3 block of cells across its right edge.*

III. *The interface* $\mathbf{z}$ *enters the* 3×3 *block of cells across its top edge, passes through the center cell and exits the* 3×3 *block of cells across its bottom edge.*

IV. *The interface* $\mathbf{z}$ *enters the* 3×3 *block of cells across its left hand edge, passes through the center cell and exits the* 3×3 *block of cells across its top edge.*

Proof. As already noted, without loss of generality I may assume that the arc length s has been chosen so that the interface is traversed from left to right as s increases. In particular, this implies that I do not need to consider any case in which the interface enters the 3×3 block of cells across its right edge.

To assist the reader in following the argument that I need only consider cases I–IV, the following is a list of *all* of the ways in which the interface g can enter and exit the 3×3 block of cells together with which of cases I–IV it is equivalent to.

(1) The interface $\mathbf{z}$ enters the 3×3 block of cells across the left edge and exits across:

   (a) The left edge. This violates the assumption that the cell size h is sufficiently small that the interface can be written as a *function* of one of the coordinate variables in terms of the other in the 3×3 block of cells B_{ij}.

   (b) The right edge. This is case II. Since, the interface can be written as a function on the 3×3 block of cells B_{ij} and the *first time* that the interface exits B_{ij} is across the right-hand edge, it has three exact column sums as shown, for example, in Figure 1. Thus, I have just proved that Theorem 10 holds for case II.

   (c) The top edge. This is case IV in the statement of the Symmetry Lemma and is the subject of Lemma 13 and Theorem 15 below. (All of the work in Section 3.2 below is concerned with proving this case when the interface is an increasing, monotonic function of x.)

   (d) The bottom edge. After reflection about the line $y = y_c$ and reversal of the arc length parameter $s \to s' = -s$ this is equivalent to (1c) immediately above and hence falls under case IV in the statement of the Symmetry Lemma.

(2) The interface $\mathbf{z}$ enters the 3×3 block of cells across the top edge and exits across:

   (a) The left edge. Upon reversal of the arc length parameter $s \to s' = -s$ this case is equivalent to case (1c), and hence is equivalent to case IV in the statement of the theorem.

   (b) The right edge. Upon reflection of the 3×3 block of cells about the midline $x = x_c$ this case is equivalent to case (1c), and hence is equivalent to case IV in the statement of the theorem.

(c) The bottom edge. This is case III of the Symmetry Lemma. It has two subcases:

(i) The interface $y = g(x)$ is strictly monotonic in the 3×3 block of cells B_{ij}, and therefore it is invertible. Rotating the 3×3 block of cells 90 degrees counterclockwise yields case (1b) and hence this case is equivalent to case II of the Symmetry Lemma. I have already proven that Theorem 10 holds in this case.

(ii) The interface $\mathbf{z}$ is not strictly monotonic in the 3×3 block of cells B_{ij}. In Lemma 12 I will prove that this case cannot occur.

(d) The top edge. This is case I of the symmetry lemma. In Lemma 11 I will prove that the condition on the maximum curvature $\kappa_{\max}$ in Equation (3) prevents this case from occurring.

(3) The interface $\mathbf{z}$ enters the 3×3 block of cells B_{ij} across the bottom edge and exits across:

(a) The left edge. After rotation of the block B_{ij} clockwise by 90 degrees this case is equivalent to case (1c), and hence is equivalent to case IV of the symmetry lemma.

(b) The right edge. After rotation of the block B_{ij} by 180 degrees and reversal of the arc length parameter $s \to s' = -s$ this case is equivalent to case (1c), and hence is equivalent to case IV of the symmetry lemma.

(c) The bottom edge. After rotation of the block B_{ij} by 180 degrees and reversal of the arc length parameter $s \to s' = -s$ this case is equivalent to (2d) which is case I of the symmetry lemma, which I prove cannot occur.

(d) The top edge. After rotation of the block B_{ij} by 180 degrees and reversal of the arc length parameter $s \to s' = -s$ this case is equivalent to (2c) above.

(4) The interface $\mathbf{z}$ enters the 3×3 block of cells B_{ij} across the right-hand edge and exits across:

(a) The right edge. As in case (1a) above, this violates the assumption that the cell size h is sufficiently small that the interface can be written as a function in the block B_{ij} and hence, this case is not allowed.

(b) The left edge.

(c) The bottom edge.

(d) The top edge.

In each of cases 4(b-d) I can change the parametrization of the interface by interchanging the arc length parameter s with $s' = -s$ so that the interface enters the 3×3 block of cells B_{ij} across its left, bottom, or top edge respectively and exits B_{ij} across its right edge. Therefore, cases 4(b-d) are equivalent to cases 1(b), 3(b), and 2(d), respectively. □

In order to prove that if the interface satisfies the hypotheses of Theorem 10, then it has at least two column sums that are exact to $O(h)$, I will often need to separate the proof into two parts:

A. The interface g is a strictly monotonic function on the interval under consideration.

B. The interface g is not a strictly monotonic function on the interval under consideration.

Recall that a function $g(x)$ is *strictly monotonic* on the interval $[a, b]$ if and only if $x < y \implies g(x) < g(y)$ for all $x, y \in [a, b]$. In the following, when I refer to the interface g as being strictly monotonic or not strictly monotonic, the interval $[a, b]$ is implicitly understood to be $[x_0, x_3]$; that is, the bottom edge of the 3×3 block of cells B_{ij} under consideration.

Recall that ξ is called a *critical point* of the function g if and only if $g'(\xi) = 0$. If the function g is a strictly monotonic function on $[x_0, x_3]$, then it *cannot* have a critical point in $[x_0, x_3]$. In the simplest cases, if g is strictly monotonic then, since it is invertible, the 3×3 grid can be rotated by 90 degrees and an interface that has only one or no exact column sums in the original orientation will have two or three exact column sums in the new orientation. However in one case — namely, the one shown in Figure 5 — the lack of a critical point makes it much more difficult to prove that the interface has at least two column sums that are exact to $O(h)$. The existence of a critical point $\xi \in [x_0, x_3]$ greatly simplifies the proof of Lemmas 11–13. In fact, as will become apparent from the proofs of these lemmas, the existence of a critical point $\xi \in [x_0, x_3]$ is sufficient to force the middle column sum S_i to be exact.

Lemma 11 (Case I of the Symmetry Lemma cannot occur). *Let $g \in C^2[x_0, x_3]$ be a nonmonotonic function that satisfies the assumptions of Theorem 10. Then case I of the symmetry lemma cannot occur*; *the interface cannot enter the 3×3 block of cells B_{ij} across its top edge at some point (x_l, y_3), pass through the center cell C_{ij} of B_{ij}, and exit B_{ij} across its top edge at some point (x_r, y_3).*

Proof. Since g is assumed to cross the line $y = y_3$ twice in the interval $[x_0, x_3]$ it is not monotonic, and since g must pass through the center cell of the 3×3 block, it follows that g must have at least one critical point $\xi \in [x_0, x_3]$ such that $g'(\xi) = 0$ and $y_3 - g(\xi) > h$. There are two cases:

A. $x_3 - \xi \leq 3h/2$; that is, ξ lies to the right of the midline $x = x_c$ of the block B_{ij}.

B. $x_3 - \xi > 3h/2$; that is, ξ lies to the left of the midline $x = x_c$ of the block B_{ij}.

I will prove the theorem for case A. I will then indicate the changes one needs to make in the proof of case A in order to prove case B. Consider the *parabolic* *comparison function* p defined by

$$p(x) = a(x-\xi)^2 + g(\xi),$$

where the coefficient a is given by

$$a = \frac{y_3 - g(\xi)}{(\tilde{x}-\xi)^2} \tag{55}$$

and $\tilde{x} = x_3 + h/4$. See Figure 6 for an example. Note that a was chosen so that

$$p(\tilde{x}) = g(x_r) = y_3, \qquad p'(\xi) = g'(\xi) = 0. \tag{56}$$

Since $g(x_r) = y_3$ and p is a monotone increasing function for $x > \xi$, and $\xi < x_r < \tilde{x}$, I must have $g(x_r) > p(x_r)$. Thus, the difference $f(x) = g(x) - p(x)$ between g and p satisfies

$$f(\xi) = g(\xi) - p(\xi) = 0, \quad f'(\xi) = g'(\xi) - p'(\xi) = 0, \quad f(x_r) = g(x_r) - p(x_r) > 0. \tag{57}$$

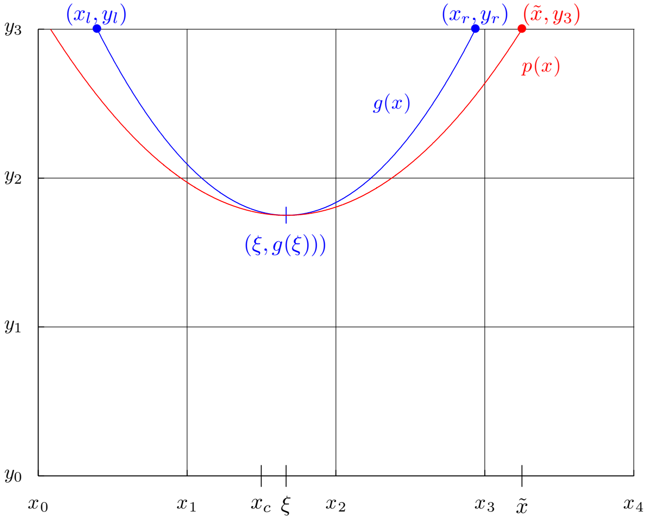


Figure 6. An example in which the interface $g(x)$ enters the top edge of the 3×3 block of cells B_{ij} at the point $(x_l, y_l) = (x_l, y_3)$, passes through the center cell C_{ij} and leaves B_{ij} at the point $(x_r, y_r) = (x_r, y_3)$. The function $p(x)$ is the parabolic comparison function that I use for this particular interface in the proof of case A of Lemma 11. The presence of a critical point $(\xi, g(\xi)) \in B_{ij}$ with $g(\xi) < y_2$ is essential to the successful use of a parabolic comparison function in the proof of Lemma 11.

The first and last of these equations imply there exists $\zeta \in [\xi, x_r]$ such that

$$f'(\zeta) = g'(\zeta) - p'(\zeta) > 0, \tag{58}$$

and this, together with the middle equation in (57), imply there exists $\eta \in [\xi, \zeta]$ such that

$$f''(\eta) = g''(\eta) - p''(\eta) > 0. \tag{59}$$

In other words,

$$g''(\eta) > p''(\eta) = 2a \quad \text{for some } \eta \in [\xi, \zeta]. \tag{60}$$

Since $x_3 - \xi \le 3h/2$, it follows that $\tilde{x} - \xi \le 7h/4$, and hence that

$$\frac{1}{(\tilde{x} - \xi)^2} \ge \frac{16}{49h^2}.$$

This inequality, together with $y_3 - g(\xi) > h$, imply

$$g''(\xi) > 2a = 2\frac{(y_3 - g(\xi))}{(\tilde{x} - \xi)^2} > \frac{32h}{49h^2} > \frac{32}{49h}.$$

From (47), I have

$$\max_{x \in [x_0, x_3]} |g''(x)| \le 8\kappa_{\max},$$

and hence $\kappa^g(\xi) \ge g''(\xi)/8$ where $\kappa^g(x)$ denotes the curvature of the interface $g(x)$ at the point $(x, g(x))$. Thus

$$\kappa^g(\xi) \ge \frac{g''(\xi)}{8} > \frac{4}{49h} > \frac{4}{52h} = \frac{1}{13h}. \tag{61}$$

Since $C_h = \dfrac{\sqrt{2}-1}{4\sqrt{3}} < \dfrac{1}{16}$, it follows from (61) that

$$\kappa^g_{\max} \ge \kappa^g(\xi) > \frac{1}{13h} > \frac{C_h}{h}.$$

Hence, the interface does not satisfy the assumption (3) and thus this interface configuration cannot occur.

In the event that case B holds, replace (x_r, y_3) with (x_l, y_3), set $\tilde{x} = x_0 - h/4$, etc., and the proof that case I of the symmetry lemma cannot occur when $x_3 - \xi > 3h/2$ (case B) is essentially identical to the proof when $x_3 - \xi \le 3h/2$ (case A). □

Recall that in the proof of the Symmetry Lemma, I showed that case II will always have three exact column sums. Hence case II has already been proved. Therefore, I must now consider case III of the Symmetry Lemma. In the proof of that case, I showed that when the interface function g is strictly monotonic it is equivalent to case II of the Symmetry Lemma, so it also has three exact column sums. Therefore, I only need to consider the nonmonotonic version of case III.

Lemma 12 (Nonmonotonic version of case III of the Symmetry Lemma). *Let $g \in C^2[x_0, x_3]$ be a nonmonotonic function satisfying the assumptions of Theorem 10. Then case III of the Symmetry Lemma cannot occur; that is, the interface cannot enter the 3×3 block of cells B_{ij} across its top edge at some point (x_l, y_3), pass through the center cell C_{ij} of B_{ij}, and exit B_{ij} across its bottom edge at some point (x_r, y_0) with $x_0 \le x_l < x_r \le x_3$.*

Proof. I will show that if the interface g enters the 3×3 block of cells B_{ij} across its top edge, passes through the center cell C_{ij} of B_{ij}, and exits B_{ij} across its bottom edge, then it cannot satisfy

$$\kappa^g_{\max} \le C_h h^{-1} \tag{62}$$

and hence it fails to satisfy the first constraint in (3).

First note that since g is nonmonotonic there is at least one point $\xi \in [x_0, x_3]$ such that $g'(\xi) = 0$. As in the proof of Lemma 11 there are two cases: A and B. However, in this proof I must also consider two subcases of each of these cases:

A. The points ξ and x_3 satisfy $x_3 - \xi \le 3h/2$ and one of the following two conditions hold:

(i) $y_3 - g(\xi) > h$ (ii) $y_3 - g(\xi) \le h$

B. The points ξ and x_3 satisfy $x_3 - \xi > 3h/2$ and one of the following two conditions hold:

(i) $y_3 - g(\xi) > h$ (ii) $y_3 - g(\xi) \le h$

I will prove the lemma for case B(i). The proofs of the other three cases are nearly identical.

Therefore, assume that $x_3 - \xi > 3h/2$ and $y_3 - g(\xi) > h$ both hold and consider the parabolic comparison function

$$p(x) = a(x - \xi)^2 + g(\xi)$$

where the coefficient a is defined by

$$a = \frac{3h - g(\xi)}{(\tilde{x} - \xi)^2} \tag{63}$$

and $\tilde{x}$ is defined by $\tilde{x} = x_0 - h/4$. Note that a was chosen so that

$$p(\tilde{x}) = y_3 = 3h, \quad p'(\xi) = g'(\xi) = 0. \tag{64}$$

Since $\tilde{x} < x_l < \xi$ and p is a monotone decreasing function for $x < \xi$, I must have $g(x_l) > p(x_l)$ as shown in Figure 7. Thus, the difference $f(x) = g(x) - p(x)$ between g and p satisfies

$$f(\xi) = g(\xi) - p(\xi) = 0, \quad f'(\xi) = g'(\xi) - p'(\xi) = 0, \quad f(x_l) = g(x_l) - p(x_l) > 0. \tag{65}$$

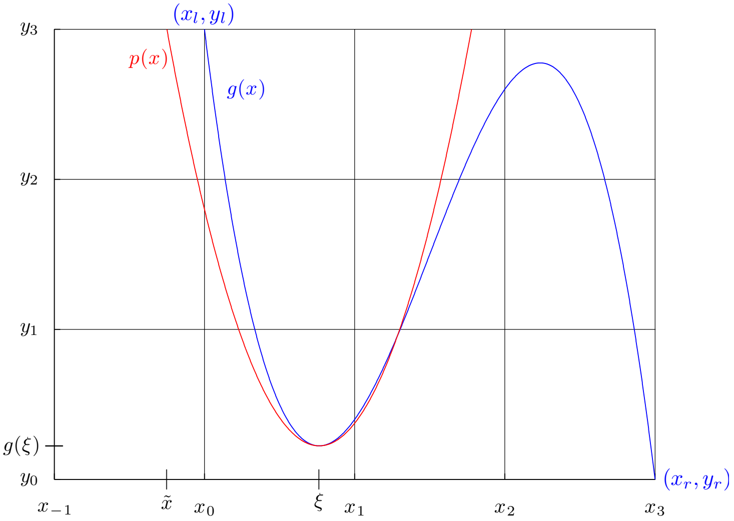


Figure 7. An example in which the interface $g(x)$ enters the 3×3 block of cells B_{ij} at its upper left corner $(x_l, y_l) = (x_0, y_3)$. It then passes through the center cell and leaves B_{ij} at its lower right corner $(x_r, y_r) = (x_3, y_3)$. The function p is the parabolic comparison function used in the proof of case B(1) of Lemma 12.

The first and last of these equations imply that there exists $\zeta \in [x_l, \xi]$ such that

$$f'(\zeta) = g'(\zeta) - p'(\zeta) < 0, \tag{66}$$

and this, together with the middle equation in (65), implies there exists $\eta \in [\zeta, \xi]$ such that

$$f''(\eta) = g''(\eta) - p''(\eta) > 0. \tag{67}$$

In other words,

$$g''(\eta) > p''(\eta) = 2a \quad \text{for some } \eta \in [\zeta, \xi]. \tag{68}$$

Note that $\xi - x_0 \leq 3h/2$ implies that $\xi - \tilde{x} \leq 7h/4$. This inequality, together with $y_3 - g(\xi) > h$, implies

$$g''(\eta) > p''(\eta) = 2a = 2\frac{y_3 - g(\xi)}{(\tilde{x} - \xi)^2} > \frac{32h}{49h^2}.$$

As in the proof of Lemma 11, it follows from (47) that $\kappa^g(\xi) > g''(\xi)/8$; hence

$$\kappa^g(\xi) \geq \frac{g''(\xi)}{8} > \frac{4}{49h} > \frac{4}{52h} > \frac{1}{13h}. \tag{69}$$

Consequently,

$$\kappa^g_{\max} \geq \kappa^g(\xi) > \frac{1}{13h} > \frac{C_h}{h},$$

whereby g fails to satisfy (62), and hence the constraint in (3) as claimed. □

Lemma 13 (Case IV of the Symmetry Lemma). *Let $g \in C^2[x_0, x_3]$ be a function that satisfies the assumptions of Theorem 10. Assume also that the interface g enters the 3×3 block of cells B_{ij} across its left edge at the point $(x_l, y_l) = (x_0, y_l)$, passes through the center cell $C_{ij} = [x_1, x_2] \times [y_1, y_2]$, and exits B_{ij} across its top edge at $(x_r, y_r) = (x_r, y_3)$ with $x_1 < x_r \leq x_3$. Then the interface has at least two column sums in B_{ij} that are either exact or exact to $O(h)$.*

Proof. I will proceed by dividing the problem into two major divisions: (1) the case in which the interface is strictly monotonic and (2) the case in which it is not. The examples in which the center column sum is not exact in any of the four standard orientations of the block B_{ij} — as shown, for example, in Figures 3, 5, 9 and 10 — are in the strictly monotonic category of case IV; the first of these two major divisions.

In order to make the argument as clear as possible, I have enumerated the proof of case IV into its various subdivisions here.

(1) The interface g is strictly monotonically increasing.

 (a) The ordinate y_l of the point (x_0, y_l) satisfies $y_0 \leq y_l \leq y_1$. Since g is strictly monotonic, it is invertible. Therefore it can be written as a function $x = g^{-1}(x)$ on the interval $[y_0, y_3]$. Furthermore, since it must pass through the center cell $C_{ij} = [x_1, x_2] \times [y_1, y_2]$ before exiting the block B_{ij} across its top edge, rotation of the block clockwise by 90 degrees will yield an orientation in which the second and third column sums are exact. Thus, this particular case of the lemma is proved.

 (b) The ordinate y_l of the point (x_0, y_l) satisfies $y_1 < y_l < y_2$. There are two subdivisions of this case:

  (i) The abscissa x_r of the point (x_r, y_3) at which the interface exits B_{ij} satisfies $x_2 \leq x_r \leq x_3$. In this case the column sums S_{i-1} and S_i are both exact and the lemma is again proved.

  (ii) The abscissa x_r of the point (x_r, y_3) at which the interface exits B_{ij} is strictly less than right-hand edge $x = x_2$ of the second column. Since the interface is assumed to be a function $y = g(x)$ on the interval $[x_0, x_3]$, and since it must pass through the center cell $C_{ij} = [x_1, x_2] \times [y_1, y_2]$, I have $x_1 < x_r < x_2$. In this case the first column sum S_{i-1} is exact and, since the interface satisfies the constraint $\kappa_{\max} \leq (\sqrt{h})^{-1}$ in (3), the second column sum S_i is exact to $O(h)$. I will prove this latter statement in Theorem 15 below.

(c) The ordinate y_l of the point (x_0, y_l) at which g enters B_{ij} satisfies $y_2 \leq y_l \leq y_3$. Since the interface is strictly monotonically increasing, it cannot enter the center cell $C_{ij} = [x_1, x_2] \times [y_1, y_2]$ if $y_l \geq y_2$. This contradicts the basic assumption that the interface passes through C_{ij}. Therefore this case must be excluded.

(2) The interface is not strictly monotonically increasing.

(a) The abscissa x_r of the point (x_r, y_3) at which the interface exits the block satisfies $x_2 \leq x_r \leq x_3$. In this case the column sums S_{i-1} and S_i are exact and once again the lemma is proved.

(b) The abscissa x_r of the point (x_r, y_3) at which the interface g exits B_{ij} is less than right-hand edge of the second column; that is, $x_r < x_2$. In this case, since g is not strictly monotonic, and since it must pass through the center cell $C_{ij} = [x_1, x_2] \times [y_1, y_2]$, g must have a critical point $(\xi, g(\xi))$ with $y_3 - g(\xi) > h$ which is also a local minimum of g. An example appears in Figure 8. I will now prove that this is inconsistent with

$$\kappa_{\max}^{g} \leq \frac{C_h}{h}, \tag{70}$$

and hence with the constraint in (3).

Proof of case (2b). Assume that the conditions listed in (2b) above hold and recall that the point $(\xi, g(\xi))$ is a local minimum of g. I form a comparison function p of the form

$$p(x) = a(x-\xi)^2 + g(\xi), \tag{71}$$

where the coefficient a is defined by

$$a = \frac{y_3 - g(\xi)}{(\xi - x_2)^2}. \tag{72}$$

Note that a was chosen so that

$$p(x_2) = y_3 = 3h, \quad p'(\xi) = g'(\xi). \tag{73}$$

Since p is a monotone increasing function for $\xi < x$ and $\xi < x_r$ I must have $g(x_r) > p(x_r)$ as shown, for example, in Figure 8.

Thus, the difference $f(x) = g(x) - p(x)$ between g and p satisfies

$$f(\xi) = g(\xi) - p(\xi) = 0, \quad f'(\xi) = g'(\xi) - p'(\xi) = 0, \quad f(x_r) = g(x_r) - p(x_r) > 0. \tag{74}$$

The first and last of these equations imply there exists $\zeta \in [\xi, x_r]$ such that

$$f'(\zeta) = g'(\zeta) - p'(\zeta) > 0, \tag{75}$$

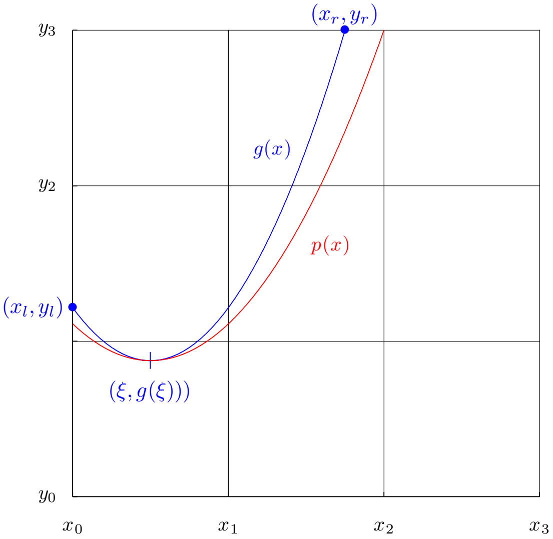


Figure 8. An example in which a nonmonotonic interface $g(x)$ enters the left edge of the 3×3 block B_{ij} at the point $(x_l, y_l) = (x_0, y_l)$ with $y_1 < y_l < y_2$. It then passes through the center cell C_{ij} and leaves B_{ij} at the point $(x_r, y_r) = (x_r, y_3)$ on its top edge with $x_0 < x_r < x_2$. The function $p(x)$ is the parabolic comparison function used in the proof of case (2b) of Lemma 13 to prove that this case cannot occur whenever the interface g satisfies the condition in (70); that is, the first of the two constraints in (3). The presence of a critical point $(\xi, g(\xi)) \in B_{ij}$ is essential to the success of the arguments in which I use a parabolic comparison function p.

and this, together with the middle equation in (74), implies that there exists $\eta \in [\xi, \zeta]$ such that

$$f''(\eta) = g''(\eta) - p''(\eta) > 0. \tag{76}$$

In other words,

$$g''(\eta) > p''(\eta) = 2a \quad \text{for some } \eta \in [\xi, \zeta]. \tag{77}$$

Since $x_2 - \xi < 2h$ and $y_3 - g(\xi) > h$ it follows that

$$g''(\xi) > 2a = 2\frac{(y_3 - g(\xi))}{(x_2 - \xi)^2} = \frac{(2h)}{(x_2 - \xi)^2} > \frac{2h}{4h^2} = \frac{1}{2h}.$$

As in the proof of Lemma 11 I have $\kappa^g(\xi) \geq g''(\xi)/8$ and hence

$$\kappa^g(\xi) \geq \frac{g''(\xi)}{8} > \frac{1}{16h} > \frac{C_h}{h}. \tag{78}$$

Consequently, $\kappa_{\max}^{g} \geq \kappa^{g}(\xi) > C_h/h$, whereby g fails to satisfy (70) and hence, the constraint on $\kappa_{\max}$ in (3) as claimed. □

3.2. ***The comparison circle*** $\tilde{\mathbf{z}}(s)$. All that remains is to prove (ii) from case (1b) in the preceding proof. This is the case in which the center column sum is not exact in each of the four standard orientations of the block B_{ij} as shown in the examples in Figures 3 and 5. The remainder of this section is devoted to proving this result, which is stated explicitly in Theorem 15 below.

Notation. In what follows it will be convenient to translate the coordinate system so that the origin coincides with the point (x_0, y_1). This results in the following relations, which I will use in several of the proofs below: $(x_0, y_1) = (0, 0)$, $(x_1, y_2) = (h, h)$, and $(x_2, y_3) = (2h, 2h)$, where $x_0, \ldots, x_3$ and $y_0, \ldots, y_3$ are the coordinates of the grid lines as shown, for example, in Figure 9.

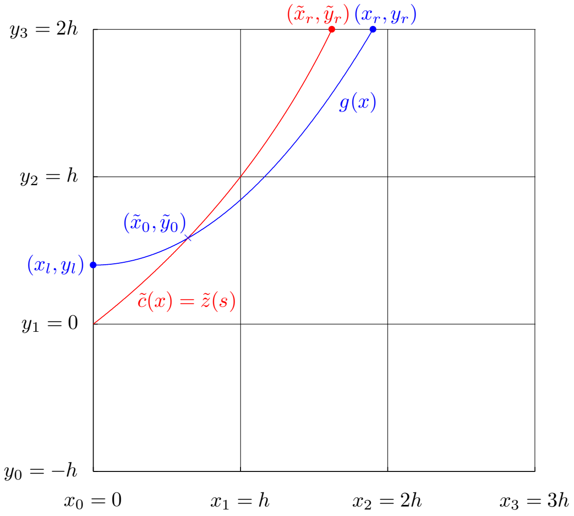


Figure 9. In this figure g is an arbitrary strictly monotonically increasing function that enters the 3×3 block B_{ij} through its left edge at the point (x_l, y_l) with $y_1 \leq y_l < y_2$, passes through the center cell C_{ij}, and exits B_{ij} through the top of its center column S_i at the point (x_r, y_r) with $x_1 < x_r < x_2$. Lemma 16 says that if g satisfies $\kappa_{\max} \leq (\sqrt{h})^{-1}$, the distance $x_2 - x_r$ is $O(h^{3/2})$. In order to prove this, I form a comparison function $\tilde{\mathbf{z}}(s)$ which is a circle that has curvature $\tilde{\kappa} = (\sqrt{h})^{-1}$ and passes through (x_0, y_1) and (x_1, y_2). In the circle comparison theorem (Theorem 14) I prove that g must eventually lie below the graph of $\tilde{\mathbf{z}}$, thereby implying that $\tilde{x}_r < x_r$. Then, in Lemma 17, I prove that $x_2 - \tilde{x}_r$ is $O(h^{3/2})$.

Now consider the circle $\tilde{\mathbf{z}}(s) = (\tilde{x}(s), \tilde{x}(s))$ defined by

$$\tilde{x}(s) = R\sin\Big(\phi_0 + \frac{s}{R}\Big) - R\sin\phi_0, \quad \tilde{y}(s) = -R\cos\Big(\phi_0 + \frac{s}{R}\Big) + R\cos\phi_0, \tag{79}$$

together with the parameters

$$\phi_0 = \frac{\pi}{4} - \sin^{-1}\frac{R}{\sqrt{2}} = \frac{\pi}{4} - \frac{s_1}{2R}, \tag{80}$$

$$s_1 = 2R\sin^{-1}\frac{R}{\sqrt{2}}, \quad s_2 = R\cos^{-1}(\cos\phi_0 - 2R) - R\phi_0. \tag{81}$$

It is relatively straightforward to check the equalities

$$\tilde{\mathbf{z}}(0) = (x_0, y_1) = (0,0), \quad \tilde{\mathbf{z}}(s_1) = (x_1, y_2) = (h,h), \quad \tilde{\mathbf{z}}(s_2) = (\tilde{x}_r, y_3) = (\tilde{x}_r, 2h). \tag{82}$$

Note that the variable $\tilde{x}_r$ in the last of these equations plays the same role with respect to the function $\tilde{\mathbf{z}}(s)$ as the variable x_r plays with respect to the interface $\mathbf{z}(s) = (x, g(x))$. Namely, $\tilde{x}_r$ is the x-coordinate at which the graph of $\tilde{\mathbf{z}}(s)$ exits the top of the 3×3 block B_{ij}. This is illustrated in Figure 9. In what follows I will often use $(x, \tilde{c}(x))$ to denote the graph of $\tilde{\mathbf{z}}(s)$ reparametrized as a function of x just as I use $(x, g(x))$ to denote the graph of the interface $\mathbf{z}(s)$.

3.3. ***The circle comparison theorem.*** Suppose that the interface $(x, g(x))$ satisfies $\kappa_{\max} \le (\sqrt{h})^{-1}$. In the following theorem I prove that once $g(x) < \tilde{c}(x)$ for some $x \in (x_0, x_2)$, then $g(x)$ must remain below $\tilde{c}(x)$ for all $x \in (\tilde{x}_0, \tilde{x}_r)$, where $(\tilde{x}_0, \tilde{y}_0)$ is the point at which g initially crosses $\tilde{c}$ as shown in Figure 9. An immediate consequence of this fact is that $\tilde{x}_r \le x_r$. Consequently, if $x_r < x_2$, then $\tilde{x}_r \le x_r < x_2$ and hence $|x_2 - x_r| \le |x_2 - \tilde{x}_r|$. Since I have constructed the comparison function c so that I can easily show that $|x_2 - \tilde{x}_r|$ is $O(h^{3/2})$, it follows that $|x_2 - x_r|$ is $O(h^{3/2})$. This, together with the fact that $g'(x) \le \sqrt{3}$ from (46), is sufficient to show that the error in the second column sum associated with g is $O(h)$.

Theorem 14 (The circle comparison theorem). *Assume that $R = \sqrt{h}$ and let $g \in C^2[x_0, x_3]$ be a strictly monotonic function that satisfies*

$$\kappa_{\max} \le (\sqrt{h})^{-1}. \tag{83}$$

Furthermore, assume that g enters the 3×3 block of cells B_{ij} on its left edge at the point (x_l, y_l) with $y_1 < y_l < y_2$, passes through the center cell C_{ij}, and exits B_{ij} through the top of its center column at the point $(x_r, y_r) = (x_r, y_3)$ with $x_1 < x_r < x_2$. Let $(\tilde{x}_0, \tilde{y}_0)$ denote the first point at which the graph of g crosses the graph of $\tilde{c}$ as shown in, for example, Figure 9. Then

$$g(x) < \tilde{c}(x) \quad \textit{for all } x \in (x_0, \tilde{x}_r]. \tag{84}$$

Proof. First note that since $\tilde{c}$ is a circle, the curvature of $\tilde{c}$ is constant: $\kappa^{\tilde{c}} = (\sqrt{h})^{-1}$. Hence, by (83),

$$\kappa^{g}(x) \le \kappa^{\tilde{c}}(x) \quad \text{for all } x \in [x_0, \tilde{x}_r].$$

To prove that (84) is true I start by assuming that

$$g(\xi) = \tilde{c}(\xi) \quad \text{for some } \xi \in (x_0, \tilde{x}_r], \tag{85}$$

and then show that this implies that the maximum curvature $\kappa_{\max}$ of g in $(\tilde{x}_0, \tilde{x}_r)$ must exceed $(\sqrt{h})^{-1}$, thereby contradicting (83).

Since $g(x) > \tilde{c}(x)$ for $x_0 < x < \tilde{x}_0$ and $g(x) < \tilde{c}(x)$ for $\tilde{x}_0 < x < \xi$ it follows that

$$g'(\tilde{x}_0) < \tilde{c}'(\tilde{x}_0). \tag{86}$$

However, since by (85) $g(\xi) = \tilde{c}(\xi)$ for some $\xi > \tilde{x}_0$ it must be the case that eventually $g'(x) \ge \tilde{c}'(x)$. Therefore let $x^* \in (\tilde{x}_0, \xi)$ be the first x such that $g'(x^*) = \tilde{c}'(x^*)$. I have

$$g'(x^*) = g'(\tilde{x}_0) + \int_{\tilde{x}_0}^{x^*} g''(x)\,dx = \tilde{c}'(\tilde{x}_0) + \int_{\tilde{x}_0}^{x^*} \tilde{c}''(x)\,dx = \tilde{c}'(x^*),$$

which, by virtue of (86), can only be true if $g''(x) > \tilde{c}''(x)$ on some subinterval of $(\tilde{x}_0, x^*)$. So in particular $g''(\eta) > \tilde{c}''(\eta)$ for some $\eta \in (\tilde{x}_0, x^*)$. Now recall that

(1) g is strictly monotonic and hence $0 < g'(x)$ for all $x \in (x_0, \tilde{x}_r]$.

(2) $0 < g'(x) < \tilde{c}'(x)$ for all $x \in (\tilde{x}_0, x^*)$.

(3) $\kappa^{g}(x) = g''(x)(1 + g'(x)^2)^{-3/2}$ for all x.

Items (1)–(3) imply that

$$\kappa^{g}(\eta) = \frac{g''(\eta)}{(\sqrt{1 + g'(\eta)^2})^3} > \frac{\tilde{c}''(\eta)}{(\sqrt{1 + \tilde{c}'(\eta)^2})^3} = \kappa^{\tilde{c}}(\eta) = \frac{1}{\sqrt{h}},$$

which contradicts (83) as claimed. □

3.4. *The column sum S_i is exact to $O(h)$.*

Theorem 15 (The column sum S_i is exact to $O(h)$). *Assume that the interface $g \in C^2[x_0, x_3]$ and that g is a strictly monotonically increasing function that satisfies*

$$\kappa_{\max} \le (\sqrt{h})^{-1}. \tag{87}$$

Furthermore, assume that the g enters the 3×3 block of cells B_{ij} on its left edge at the point (x_l, y_l) with $y_1 \le y_l \le y_3$, passes through the center cell $C_{i,j} = [x_1, x_2] \times [y_1, y_2]$, and exits B_{ij} through the top of its center column at the point $(x_r, y_r) = (x_r, y_3)$ with $x_1 < x_r < x_2$ as shown, for example, in Figure 10. Then the

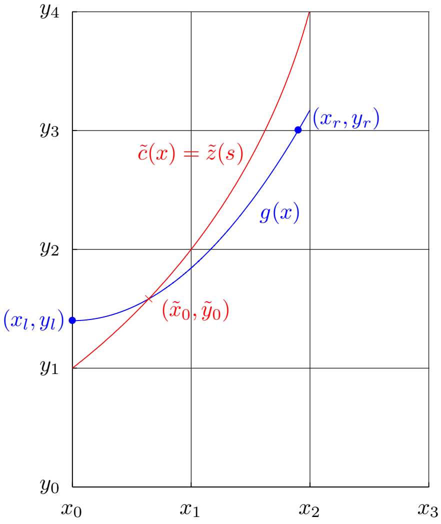


Figure 10. To see the error between the center column sum S_i and the exact volume (area) under the interface $y = g(x)$, I have plotted the row of cells that lie above the standard 3×3 block of cells $B_{i,j}$ centered on the cell $C_{i,j} = [x_1, x_2] \times [y_1, y_2]$ in which the approximation to the interface g will be constructed. I have also plotted the comparison circle $\tilde{c}(x)$ which, in Theorem 14, I prove provides an upper bound on $g(x)$ for all $x \in [\tilde{x}_0, x_2]$ where $(\tilde{x}_0, \tilde{y}_0)$ is the point at which the interface g intersects comparison circle $\tilde{c}$.

error between the column sum S_i and the normalized integral of g over the second column is $O(h)$:

$$\left| S_i - h^{-2} \int_{x_1}^{x2} (g(x) - y_0)\, dx \right| \le C_S h, \tag{88}$$

where

$$C_S = 8\sqrt{3}(2\sqrt{2} - 1)^2. \tag{89}$$

Proof. As one can see from the example shown in Figure 10, the error between the column sum S_i and the exact normalized volume (area) under the interface $y = g(x)$ in the center column is

$$h^{-2} \int_{x_1}^{x2} (g(x) - y_0)\, dx - S_i = h^{-2} \int_{x_r}^{x2} (g(x) - y_3)\, dx,$$

since

$$S_i = h^{-2} \int_{x_1}^{x2} (\min\{g(x), y_3\} - y_0)\, dx,$$

and, by assumption, $\min\limits_{[x_0,x_r]} g(x) \geq y_l \geq y_1$. Thus, it suffices to show that

$$\left| \int_{x_r}^{x2} (g(x) - y_3)\, dx \right| \leq C_S h^3. \tag{90}$$

In other words, I need to show that the volume in the region below the interface $y = g(x)$ that lies in the cell $C_{2,4}$ is $O(h^3)$.

By (46) in I have $|g'(x)| \leq \sqrt{3}$. This implies

$$\left| \int_{x_r}^{x2} (g(x) - y_3)\, dx \right| \leq \left| \int_{x_r}^{x2} l(x)\, dx \right|, \tag{91}$$

where $l(x)$ is the line with slope $\sqrt{3}$ that passes through the point x_r. The region of integration on the right side of (91) is a right triangle with corners (x_r, y_3), (x_2, y_3), and $(x_2, y_3 + \sqrt{3}(x_2 - x_r))$ and the integral is the area of this triangle, namely, $\sqrt{3}(x_2 - x_r)^2/2$. Thus I have

$$\left| \int_{x_r}^{x2} (g(x) - y_3)\, dx \right| \leq \left| \int_{x_r}^{x2} l(x)\, dx \right| \leq \frac{\sqrt{3}}{2}(x_2 - x_r)^2 \leq \frac{\sqrt{3}}{2}\tilde{C}^2 h^3, \tag{92}$$

where the bound $(x_2 - x_r)^2 \leq \tilde{C}^2 h^3$ between the second to last and last terms in (92) follows from the inequality (93) immediately below. Equation (92) implies (90). Equation (88) — and hence the theorem — follows immediately. □

Lemma 16 ($x_2 - x_r$ is $O(h^{3/2})$). *Let $g \in C^2[x_0, x_3]$ be a function that satisfies the assumptions stated in Theorem 14. Then*

$$x_2 - x_r \leq \tilde{C} h^{3/2}, \tag{93}$$

where

$$\tilde{C} = 4(2\sqrt{2} - 1). \tag{94}$$

Proof. By the circle comparison theorem (Theorem 14) there exists a point $\tilde{x}_0 \in [x_0, x_r)$ such that

$$g(x) \leq \tilde{c}(x) \quad \text{for all } x \in [\tilde{x}_0, x_r].$$

This implies that $\tilde{x}_r \leq x_r$. Since by assumption $x_r < x_2$, Equation (93) follows immediately from Equation (95) in Lemma 17 below. □

Lemma 17 ($x_2 - \tilde{x}_r$ is $O(h^{3/2})$). *Let $R = \sqrt{h}$ and let $\tilde{x}_r$ be defined as in (82) above. Then*

$$x_2 - \tilde{x}_r \leq \tilde{C} h^{3/2}, \tag{95}$$

where $\tilde{C}$ is defined in (94).

Proof. Since the coordinate system has been arranged so that the origin is at the point (x_0, y_1) and hence $x_2 = 2h = y_3$ (for example, see Figure 9), I have

$$x_2 = \tilde{y}_r = \tilde{y}(s_2).$$

Thus

$$\begin{aligned} x_2 - \tilde{x}_r &= \tilde{y}(s_2) - \tilde{x}(s_2) \\ &= R\{(\cos\phi_0 - \cos(\phi_0 + s_2/R)) - (-\sin\phi_0 + \sin(\phi_0 + s_2/R))\}, \end{aligned} \tag{96}$$

and since $R = \sqrt{h}$, it suffices to show that the quantity inside the curly braces in (96) is $O(R^2) = O(h)$. I can rewrite (96) as

$$x_2 - \tilde{x}_r = R\{(\cos\phi_0 + \sin\phi_0) - (\cos(\phi_0 + \theta) + \sin(\phi_0 + \theta))\}, \tag{97}$$

where $\theta = s_2/R$. Consider the quantity A defined by dividing (97) by R:

$$A = \{(\cos\phi_0 + \sin\phi_0) - (\cos(\phi_0 + \theta) + \sin(\phi_0 + \theta))\}. \tag{98}$$

Now expand $\cos(\phi_0 + \theta)$ and $\sin(\phi_0 + \theta)$ in a Taylor series about $\cos\phi_0$ and $\sin\phi_0$ to obtain

$$\begin{aligned} A &= (\cos\phi_0 + \sin\phi_0) - \big(\cos(\phi_0 + \theta) + \sin(\phi_0 + \theta)\big) \\ &= -(\cos\phi_0 - \sin\phi_0)\theta + (\cos\phi_0 + \sin\phi_0)\frac{\theta^2}{2!} + (\cos\phi_0 - \sin\phi_0)\frac{\theta^3}{3!} \\ &\quad - (\cos\phi_0 + \sin\phi_0)\frac{\theta^4}{4!} - (\cos\phi_0 - \sin\phi_0)\frac{\theta^5}{5!} + (\cos\phi_0 + \sin\phi_0)\frac{\theta^6}{6!} + \cdots . \end{aligned}$$

After some manipulation one obtains

$$\begin{aligned} A = &-\Big((\cos\phi_0 - \sin\phi_0) - (\cos\phi_0 + \sin\phi_0)\frac{\theta}{2}\Big)\theta \\ &+ \Big((\cos\phi_0 - \sin\phi_0) - (\cos\phi_0 + \sin\phi_0)\frac{\theta}{4}\Big)\frac{\theta^3}{3!} \\ &- \Big((\cos\phi_0 - \sin\phi_0) - (\cos\phi_0 + \sin\phi_0)\frac{\theta}{6}\Big)\frac{\theta^5}{5!} + \cdots . \end{aligned} \tag{99}$$

The first term in this series is $O(R^2) = O(h)$. To see this note that by Lemma 19 below $\cos\phi_0 - \sin\phi_0 = R$ and $\cos\phi_0 + \sin\phi_0 = \sqrt{2 - R^2}$ so that the series for A in (99) becomes

$$A = -\Big(R - \frac{\theta}{2}\sqrt{2 - R^2}\Big)\theta + \Big(R - \frac{\theta}{4}\sqrt{2 - R^2}\Big)\frac{\theta^3}{3!} - \Big(R - \frac{\theta}{6}\sqrt{2 - R^2}\Big)\frac{\theta^5}{5!} + \cdots . \tag{100}$$

The first term is positive, because $R = \sqrt{h}$, $\theta = s_2/R$, and $s_2 \geq h$ (see (102) below). Thus,

$$\Big(\frac{\theta}{2}\sqrt{2 - R^2} - R\Big)\theta \geq \Big(\frac{h}{R}\sqrt{2 - R^2} - R\Big)\frac{h}{R} \geq \big(\sqrt{2 - R^2} - 1\big)R^2 > 0,$$

for all $0 < h \le 1$, and hence all $0 < R \le 1$. Similarly, since $s_2 \le 4h$ (see Lemma 18 again), it follows that

$$\Big(\frac{\theta}{2}\sqrt{2-R^2} - R\Big)\theta \le \big(2R\sqrt{2-R^2} - R\big)4R = 4\big(2\sqrt{2-R^2} - 1\big)R^2, \tag{101}$$

for all $0 < h \le 1$, or equivalently all $0 < R \le 1$. Combining equations (97), (98), (100), and (101) yields

$$x_2 - \tilde{x}_r \le 4(2\sqrt{2-R^2} - 1)R^3 + O(R^5) \le 4(2\sqrt{2} - 1)R^3 + O(R^5).$$

It is possible to show — for example by plotting it with MATLAB — that the coefficient $(R - \theta\sqrt{2-R^2})/4$ of the second term in the expansion of A in terms of R in (100) is negative for $0 < h \le 1$ and that furthermore, the tail of the series is bounded by this term. Equation (95) follows immediately. □

Lemma 18 ($s_2 = O(h)$). *Assume that $h \le 1$ and let s_2 be defined as in (81). Then*

$$h \le s_2 \le 4h. \tag{102}$$

Proof. First, note that I am only interested in functions g that exit the 3×3 block of cells at the point (x_r, y_3) when $x_r < x_2$ as shown for example in Figure 3. For otherwise the first and second column sums would be exact and I would be done.

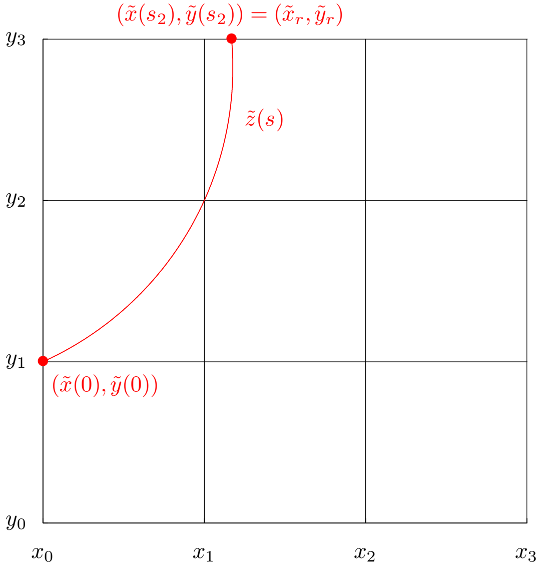

$(\tilde{x}(s_2), \tilde{y}(s_2)) = (\tilde{x}_r, \tilde{y}_r)$
$\tilde{z}(s)$
$(\tilde{x}(0), \tilde{y}(0))$
y_3
y_2
y_1
y_0
x_0
x_1
x_2
x_3

Figure 11. In this figure $h = \frac{1}{4}$ and hence the comparison circle $\tilde{\mathbf{z}}(s)$ has radius $R = \sqrt{h} = 2h$.

Since a consequence of Theorem 14 is that $\tilde{x}_r = \tilde{x}(s_2) \leq x_r$, it follows that I am only interested in values of $R = \sqrt{h}$ and s_2 such that $\tilde{x}_r < x_2$.

To obtain the lower bound on s_2 in (102) note that s_2 is an arc of the circle $\tilde{\mathbf{z}}$ and that when $h = 1$ the radius of $\tilde{\mathbf{z}}$ is $R = \sqrt{h} = h$. In this case the point $(\tilde{x}_r, \tilde{y}_r) = (x_0, y_3)$ and hence $\tilde{x}_r = x_0$. Since this is half the circumference of the circle with center (x_0, y_2) and radius h, $s_2 = \pi$ when $h = 1$. Since s_2 will always be greater than the length of the chord connecting the points (x_0, y_1) and $(\tilde{x}_r, \tilde{y}_r)$ and since this particular chord is the diameter of $\tilde{\mathbf{z}}$ all other chords of $\tilde{\mathbf{z}}$ will be smaller. In particular, since the radius of $\tilde{\mathbf{z}}$ $R = \sqrt{h} \to 0$ as $h \to 0$, all chords connecting (x_0, y_1) and $(\tilde{x}_r, \tilde{y}_r)$ will be smaller than this one. The lower bound on s_2 in (102) follows immediately.

In order to write s_2 in the form $s_2 = Ch$ where C is a constant independent of h note that since $h \leq 1$ and $\tilde{x}(s_2) < x_2$, the arc of the circle that connects the points $(\tilde{x}(0), \tilde{y}(0))$ and $(\tilde{x}(s_2), \tilde{y}(s_2))$ always lies entirely within the triangle with vertices (x_0, y_1), (x_2, y_1) and (x_2, y_3), as shown in Figure 11, for example. Hence, the arc length s_2 will always be bounded above by the sum of the lengths of the two perpendicular sides of this right triangle; namely,

$$s_2 < 4h.$$

This is the upper bound on s_2 in (102). □

In order to prove that $x_2 - \tilde{x}_r = O(h^{3/2})$ in Lemma 17, I expanded $x_2 - \tilde{x}_r$ in a Taylor series about the point ϕ_0. As we saw in Lemma 17 the coefficient of the first nonzero term in this expansion is $\cos\phi_0 - \sin\phi_0$. Hence, the fact that $\cos\phi_0 - \sin\phi_0 = R$ is a crucial part of the proof that $|x_2 - \tilde{x}_r| = O(h^{3/2})$. The purpose of the following lemma is to prove this fact and also to establish the value of $\cos\phi_0 + \sin\phi_0$.

Lemma 19 ($\cos\phi_0 - \sin\phi_0 = R$). *Let ϕ_0 be defined as in (80):*

$$\phi_0 = \frac{\pi}{4} - \sin^{-1}\frac{R}{\sqrt{2}}.$$

Then

$$\cos\phi_0 - \sin\phi_0 = R, \quad \cos\phi_0 + \sin\phi_0 = \sqrt{2 - R^2}. \tag{103}$$

Proof. Define θ by $\sin\theta = R/\sqrt{2}$, so that

$$\phi_0 = \frac{\pi}{4} - \sin^{-1}\frac{R}{\sqrt{2}} = \frac{\pi}{4} - \theta.$$

The first equation in (103) follows from writing ϕ_0 as $\pi/4 - \theta$ and applying the trigonometric identities for the sine and cosine of the difference of two angles:

$$\cos\phi_0 - \sin\phi_0 = \sqrt{2}\sin\theta = R.$$

To prove the second equality in (103) I again use the trigonometric identities for the sine and cosine of the difference of two angles, together with the trigonometric identity $\cos(\arcsin x) = \sqrt{1-x^2}$, to obtain

$$\cos\phi_0 + \sin\phi_0 = \sqrt{2}\cos\theta = \sqrt{2}\cos\Bigl(\sin^{-1}\frac{R}{\sqrt{2}}\Bigr) = \sqrt{2-R^2}. \qquad \square$$

4. Second-order accuracy in the max norm

In this section I will assume the coordinate system has been arranged so that the bottom edge of the 3×3 block of cells B_{ij} lies along the line $y=0$ and that the vertical line $x=x_c$ which passes through the center of the center cell is $x=0$ as shown in Figure 12. In particular, note that the origin is at the center of the bottom edge of the 3×3 block and the center of C_{ij} is $(0, 3h/2)$ as shown in the figure.

I will also denote the interval that forms the bottom of the 3×3 block B_{ij} by I, and the intervals $[x_{i-2}, x_{i-1}]$, $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$ that are associated with the three columns of B_{ij} by $I_{i+\alpha}$ for $\alpha = -1, 0, 1$. Thus, $I = [-3h/2, 3h/2]$ and

$$I_{i+\alpha} \equiv \begin{cases} [-3h/2, -h/2] & \text{if } \alpha = -1, \\ [-h/2, h/2] & \text{if } \alpha = 0, \\ [h/2, 3h/2] & \text{if } \alpha = 1. \end{cases}$$

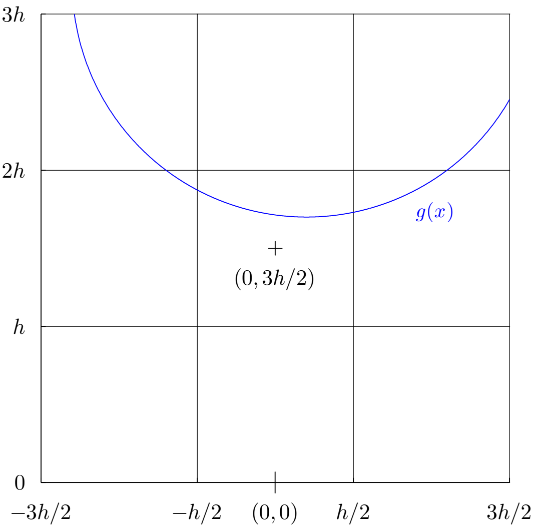


Figure 12. In this section I will work with the coordinate system shown here. The origin is at the center of the bottom of the 3×3 block B_{ij} so that the center of the center cell C_{ij} is $(0, 3h/2)$ as shown in the figure. This latter point corresponds to the point labeled (x_c, y_c) in some of the other figures.

Given an arbitrary integrable function $g(x)$ on the interval $I = [-3h/2, 3h/2]$, let $\Lambda_{i,j}(g)$ denote the volume fraction due to g in the center cell

$$\Lambda_{i,j}(g) = h^{-2} \int_{I_i} \theta_j(g(x))\,dx. \tag{104}$$

where $\theta_j(g)$ is defined by

$$\theta_j(g) \equiv (g(x) - (j-1)h)_+ - (g(x) - jh)_+ \tag{105}$$

and

$$x_+ = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{if } x \le 0, \end{cases} \tag{106}$$

is the ramp function. I will denote the volume fractions in the other cells similarly; that is, I will use $\Lambda_{i',j'}(g)$ for $i' = i-1, i, i+1$ and $j' = j-1, j, j+1$ to denote the volume fraction in the (i', j')-th cell. When the function g under consideration is apparent, I will simply write $\Lambda_{i',j'}$ or equivalently $\Lambda_{i+\alpha,j+\beta}$ for some $\alpha, \beta = -1, 0, 1$.

In the following lemma I make the implicit assumption that the 3×3 block of cells B_{ij} has been arranged so that the volume fraction $\Lambda_{i,j}(g)$ is the volume (area) of *dark fluid* in the center cell. In other words, if one assumes that the block B_{ij} has been rotated so that the interface $\mathbf{z}$ can be represented as a function $g(x)$ on the interval $I = [-3h/2, 3h/2]$, then there are two possibilities:

(1) $\Lambda_{i,j}(g) = h^{-2} \int_{I_i} \theta_j(g)\,dx$ is the volume of dark fluid in C_{ij},

(2) $\Lambda_{i,j}(g) = h^{-2} \int_{I_i} \theta_j(g)\,dx$ is the volume of light fluid in C_{ij}.

In the event that (2) holds, one can reflect the 3×3 block B_{ij} about the line $y = y_c$, where $y_c = (y_j + y_{j+1})/2$ is the line that divides the block $B_{i,j}$ in half horizontally, to ensure that case (1) holds. This is necessary because when I write the piecewise linear approximation $\tilde{g}_{i,j}(x) = m_{i,j}x + b_{i,j}$ to $g(x)$ in $C_{i,j}$ I am implicitly assuming that

$$\Lambda_{i,j}(\tilde{g}_{i,j}) = h^{-2} \int_{I_i} \theta_j(\tilde{g}_{i,j}(x))\,dx$$

is the volume of dark fluid in $C_{i,j}$. It is necessary to be consistent about which fluid is represented by the volume fraction $\Lambda_{i,j}$ in order to prove the following lemma.

Lemma 20 (Equal volume fractions ensure that $\tilde{g}$ intersects g in the center cell $C_{i,j}$). *Let $g(x)$ be a continuous function on the interval $I_i \equiv [-h/2, h/2]$ and assume that a portion of the interface $g(x)$ passes through the center cell*

$$C_{i,j} = [-h/2, h/2] \times [h, 2h].$$

Furthermore, assume that the 3×3 *block of cells* B_{ij} *centered on* $C_{i,j}$ *has been arranged so that*

$$\Lambda_{i,j}(g) = h^{-2} \int_{I_i} \theta_j(g(x))\, dx \tag{107}$$

is the (*nonzero*) *volume fraction of dark fluid in* $C_{i,j}$. *Let*

$$\tilde{g}(x) = mx + b \tag{108}$$

be a piecewise linear approximation to g *that passes through the center cell* $C_{i,j}$ *and assume that* g *and* $\tilde{g}$ *have the same volume fraction*

$$0 < \Lambda_{i,j}(g) = \Lambda_{i,j}(\tilde{g}) < 1$$

in C_{ij}. *Then there exists a point* $x^* \in I_i = [-h/2, h/2]$ *such that*

$$g(x^*) = \tilde{g}(x^*).$$

Proof. Consider

$$h^{-2} \int_{I_i} \big[\theta_j(g(x)) - \theta_j(\tilde{g}(x))\big]\, dx = \Lambda_{i,j}(g) - \Lambda_{i,j}(\tilde{g}) = 0,$$

and note that $\theta_j(g)$ defined in (105) is a strictly monotonically increasing function of $g(x)$:

$$g(x) < \tilde{g}(x) \Rightarrow \theta_j(g(x)) < \theta_j(\tilde{g}(x)).$$

Therefore, in order for $\Lambda_{i,j}(g) = \Lambda_{i,j}(\tilde{g})$ to hold, there are two possibilities. The first is that $g(x) = \tilde{g}(x)$ for all $x \in I_i$, in which case the theorem is true and x^* is any point in I_i.

The second possibility is that there exists a point $x_- \in I_i$ with $g(x_-) < \tilde{g}(x_-)$ and there also exists a point $x_+ \in I_i$ where $g(x_+) > \tilde{g}(x_+)$. Thus, since both $g(x)$ and $\tilde{g}(x)$ are continuous, there must be a point x^* between x_- and x_+ where $g(x^*) = \tilde{g}(x^*)$. To see this, consider the function $f(x) = g(x) - \tilde{g}(x)$. The function f is continuous and furthermore,

$$f(x_+) = g(x_+) - \tilde{g}(x_+) > 0, \quad f(x_-) = g(x_-) - \tilde{g}(x_-) < 0.$$

Hence, if $x_- < x_+$, then there must exist an $x^* \in (x_-, x_+) \subset I_i$ (or, if $x_+ < x_-$, then $x^* \in (x_+, x_-) \subset I_i$) such that $f(x^*) = 0$, or equivalently, $g(x^*) = \tilde{g}(x^*)$, as claimed. □

An immediate consequence of this lemma is that the piecewise constant volume-of-fluid interface reconstruction algorithm as defined below must be first-order accurate. The details are as follows.

Definition 21. The *piecewise constant VOF interface reconstruction algorithm* is defined by

$$\tilde{g}(x) = y_{j-1} + h\Lambda_{i,j}(g) \quad \text{for all } x \in I_i = [x_{i-1}, x_i], \tag{109}$$

where, as usual, I have assumed that the 3 × 3 block B_{ij} centered about the cell C_{ij} in which I want to reconstruct the interface has been rotated so that the interface can be written as a single valued function $g(x)$ on the interval $I = [-3h/2, 3h/2]$ and

$$\Lambda_{i,j}(g) = h^{-2} \int_{I_i} \theta_j(g)\, dx$$

is the volume of dark fluid in C_{ij}.

Corollary 22 (The piecewise constant VOF interface reconstruction algorithm is first-order). *Suppose that the interface passes through a portion of the cell $C_{i,j}$ and that it can be represented as a C^2 function on the interval $I = [-3h/2, 3h/2]$. Then the piecewise constant interface reconstruction algorithm defined in (109) produces a first-order accurate approximation $\tilde{g}$ to the exact interface g in $C_{i,j}$:*

$$|g(x) - \tilde{g}(x)| \le C_P h \quad \textit{for all } x \in I_i = [-h/2, h/2],$$

where $C_P = \sqrt{3}$.

Proof. By assumption the interface g is continuous and passes through the center cell C_{ij}. Furthermore, the piecewise constant interface reconstruction algorithm defined in (109) is a member of the class of piecewise linear approximations to g. Therefore, Lemma 20 applies, and hence there exists a point $x^* \in I_i$ such that $y_{j-1} \le g(x^*) \le y_j$ and

$$g(x^*) = \tilde{g}(x^*).$$

The assumption[10] that $g \in C^2[I]$ allows me to apply Theorem 6 to obtain (see Equation (46))

$$\left|g'(x)\right| \le \sqrt{3} \quad \text{for all } x \in [-h/2, h/2]. \tag{110}$$

Thus, applying the Taylor remainder theorem [29] to $g(x)$, I find that for all $x \in [-h/2, h/2]$

$$|g(x) - \tilde{g}(x)| = |g(x^*) + g'(\xi)(x - x^*) - \tilde{g}(x^*)| \le |g'(\xi)h| \le \sqrt{3}h,$$

since $\xi = \xi(x)$ is some number between x and x^* (that is, $\xi \in [-h/2, h/2]$) and hence, (110) applies. □

[10] Actually, I only need the interface g to be one times continuously differentiable on I_i; that is, $g \in C^1[I_i]$. I have assumed $g \in C^2[I]$ here so that I will not have to prove a special version of Theorem 6 in order to obtain the bound in (46) on $g'(x)$.

Theorem 23 (The approximation to g' is first-order accurate). *Assume that the interface $g \in C^2[I]$ where $I = [-3h/2, 3h/2]$ and that at least two distinct column sums $S_{i+\alpha}$ and $S_{i+\beta}$ with $\alpha, \beta = 1, 0, -1$ and $\alpha \neq \beta$ are exact to $O(h)$:*

$$\left| S_{i+\alpha} - h^{-2} \int_{I_{i+\alpha}} g(x)\,dx \right| \leq C_S h, \tag{111}$$

$$\left| S_{i+\beta} - h^{-2} \int_{I_{i+\beta}} g(x)\,dx \right| \leq C_S h, \tag{112}$$

where

$$C_S = 8\sqrt{3}(2\sqrt{2} - 1)^2 \tag{113}$$

is the constant obtained in Theorem 15. Then the slope defined by

$$m = \frac{S_{i+\alpha} - S_{i+\beta}}{\alpha - \beta} \quad \textit{for } \alpha, \beta = 1, 0, -1 \textit{ with } \alpha \neq \beta$$

of the piecewise linear approximation $\tilde{g}(x) = mx + b$ to the exact interface g satisfies

$$|m - g'(0)| \leq \left(\tfrac{26}{3}\kappa_{\max} + C_S\right)h. \tag{114}$$

Proof. Note that during the course of proving Theorem 10, I have shown that the only column sum that may not be exact is the middle one, S_i; for example, see the list in the proof of the Symmetry Lemma. Therefore, I may assume that

$$S_{i+\alpha} = h^{-2} \int_{I_{i+\alpha}} g(x)\,dx \quad \text{if } \alpha = 1 \text{ or } -1. \tag{115}$$

Now note that the inequality in (88) can be rewritten in the following way. If (88) holds for the i-th column sum S_i, then there exists $\epsilon_i > 0$ with $|\epsilon_i| \leq C_S h$ such that

$$h^{-2} \int_{I_i} g(x)\,dx = S_i + \epsilon_i \qquad \text{if } \alpha = 0. \tag{116}$$

In other words, if the column sum S_i is not exact, then ϵ_i is the area of the region bounded by the horizontal line $y = y_3$, the vertical line $x = x_2$, and the graph of the interface $y = g(x)$ as shown in Figure 10. Otherwise, the column sum S_i is exact and $\epsilon_i = 0$.

By the Taylor remainder theorem

$$g(x) = g(0) + g'(0)x + \tfrac{1}{2}g''(\xi)x^2, \tag{117}$$

for some $\xi \in (-x, x)$.[11] Applying (117) to g and performing the integration in equations (115) and (116) for each $\alpha = -1, 0, 1$ yields

$$S_{i-1} = g(0)h^{-1} - g'(0) + \tfrac{13}{24} g''(\xi_{-1})h, \tag{118}$$

$$S_i = g(0)h^{-1} + \tfrac{1}{24} g''(\xi_0)h - \epsilon_i, \tag{119}$$

$$S_{i+1} = g(0)h^{-1} + g'(0) + \tfrac{13}{24} g''(\xi_1)h, \tag{120}$$

where the term with $g'(0)$ has dropped out of the expression for S_i, since $g'(0)x$ is an odd function of x and the interval $I_i = [-h/2, h/2]$ is centered about $x = 0$.

Subtracting the expression in (118) from the expression in (120) and dividing by 2 yields the centered difference approximation to the derivative $g'(0)$ plus error terms:

$$\frac{S_{i+1} - S_{i-1}}{2} = g'(0) + \tfrac{13}{24}\big(g''(\xi_1) + g''(\xi_{-1})\big)h. \tag{121}$$

Rearranging the terms in (121) and using (47) yields

$$\left|\frac{S_{i+1} - S_{i-1}}{2} - g'(0)\right| = \left|\tfrac{13}{24}\big(g''(\xi_1) + g''(\xi_{-1})\big)h\right| \le \tfrac{26}{3}\kappa_{\max}h \le \big(\tfrac{26}{3}\kappa_{\max} + C_S\big)h.$$

Similarly, subtracting S_{i-1} from S_i and S_i from S_{i+1} yield the two one-sided difference approximations to $g'(0)$,

$$|(S_i - S_{i-1}) - g'(0)| \le \big(\tfrac{14}{3}\kappa_{\max} + C_S\big)h,$$
$$|(S_{i+1} - S_i) - g'(0)| \le \big(\tfrac{14}{3}\kappa_{\max} + C_S\big)h.$$

The inequality in (114) follows immediately. □

The following theorem is the main result of this paper.

Theorem 24. *Assume the interface $g \in C^2[I]$ where $I = [-3h/2, 3h/2]$ and that at least two of the column sums $S_{i+\alpha}$ and $S_{i+\beta}$ for $\alpha, \beta = 1, 0, -1$ with $\alpha \neq \beta$ are exact to $O(h)$. Let*

$$\tilde{g}(x) = mx + b$$

be a piecewise linear approximation to $g(x)$ in $I_i = [-h/2, h/2]$ with

$$m = \frac{S_{i+\alpha} - S_{i+\beta}}{\alpha - \beta}, \tag{122}$$

and assume that $g(x)$ and $\tilde{g}(x)$ have the same volume fraction in the center cell:

$$\Lambda_{i,j}(g) = \Lambda_{i,j}(\tilde{g}).$$

[11] Technically speaking, if $x > 0$, then $\xi \in (0, x)$, while if $x < 0$, then $\xi \in (x, 0)$. My intention is for the notation $\xi \in (-x, x)$ to cover both cases.

Then $\tilde{g}(x)$ is a second-order accurate approximation to $g(x)$ in I_i:

$$|g(x)-\tilde{g}(x)| \le \bigl(\tfrac{50}{3}\kappa_{\max}+C_S\bigr)h^2 \quad \textit{for all } x\in I_i=[-h/2,h/2]$$

where

$$C_S = 8\sqrt{3}(2\sqrt{2}-1)^2. \tag{123}$$

Proof. By Lemma 20 I know that there exists $x^* \in I_i = [-h/2, h/2]$ such that $g(x^*) = \tilde{g}(x^*)$. Let $x \in I_i$ be arbitrary, but fixed. By the Taylor remainder theorem I know that there exists $\xi = \xi(x) \in I_i$ such that

$$g(x) = g(x^*) + g'(x^*)(x-x^*) + \tfrac{1}{2}g''(\xi)(x-x^*)^2.$$

Hence,

$$\begin{aligned}|g(x)-\tilde{g}(x)| &= \bigl|g(x^*)+g'(x^*)(x-x^*)+\tfrac{1}{2}g''(\xi)(x-x^*)^2-\tilde{g}(x^*)-m(x-x^*)\bigr|\\ &\le |g'(x^*)-m||x-x^*|+\tfrac{1}{2}|g''(\xi)|(x-x^*)^2\\ &\le |g'(x^*)-m|h+4\kappa_{\max}h^2,\end{aligned}$$

where I have used (47) to bound $g''(\xi)$ and the fact that $x, x^* \in I_i = [-h/2, h/2]$ to obtain $|x-x^*| \le h$. In order to bound $\bigl|g'(x^*)-m\bigr|$ I rewrite this expression as:

$$|g'(x^*)-m| = |g'(x^*)-g'(0)| + |g'(0)-m|. \tag{124}$$

In order to bound the first term on the right side of (124) I expand $g'(x^*)$ in a Taylor series about $x = 0$ and use the Taylor remainder theorem to obtain

$$g'(x^*) = g'(0) + g''(\zeta)(x^*-0),$$

for some $\zeta \in I_i$. From (47) and, since $x^* \in I_i = [-h/2, h/2]$ implies $|x^*| \le h/2$, I have

$$|g'(x^*)-g'(0)| \le |g''(\zeta)||x^*| \le 4\kappa_{\max}h. \tag{125}$$

Finally, using the bound on $|g'(0)-m|$ in (114), I have

$$\begin{aligned}|g(x)-\tilde{g}(x)| &\le (|g'(x^*)-g'(0)|+|g'(0)-m|)h+4\kappa_{\max}h^2\\ &\le \bigl(8+\tfrac{26}{3}\bigr)\kappa_{\max}h^2+C_Sh^2 = \bigl(\tfrac{50}{3}\kappa_{\max}+C_S\bigr)h^2,\end{aligned}$$

as claimed. □

5. Conclusions

Given any C^2 curve $\mathbf{z}(s)$ in $\mathbb{R}^2$ overlaid with a computational grid consisting of square cells, each with (nondimensional) side h, I have proven that for each cell C_{ij} that contains a portion of the curve $\mathbf{z}(s)$ there exist at least two columns or two rows in the 3×3 block of cells B_{ij} centered on the cell C_{ij} whose divided difference is a first-order accurate approximation m_{ij} to the slope of the curve $\mathbf{z}(s)$ in the center cell C_{ij}. This approximation to the slope of $\mathbf{z}$ in C_{ij}, together with the knowledge of the exact volume fraction Λ_{ij} in C_{ij}, is sufficient to construct a line segment $\tilde{g}_{ij}(x)$ that is an $O(h^2)$ approximation to the curve $\mathbf{z}(s) = (x(s), g(x(s))$ it in the max norm in that cell:

$$|g(x) - \tilde{g}_{ij}(x)| \leq C(\kappa_{\max})h^2 \quad \text{for all } x \in [x_i, x_{i+1}]. \tag{126}$$

Here $\kappa_{\max}$ is the maximum curvature of the interface $\mathbf{z}$ in the 3×3 block of cells B_{ij} centered on the cell C_{ij}, $C\,(\kappa_{\max})$ is a constant that depends on $\kappa_{\max}$ but is independent of h, and x_i, x_{i+1} denote the left and right edges, respectively, of the cell C_{ij}.

I *have not* demonstrated a way in which to find these two columns or two rows given the volume fraction information in the 3×3 block of cells B_{ij} centered on the cell C_{ij}. However, there are at least two algorithms currently in use that may provide the user with a way to choose the columns correctly, and hence produce a first-order accurate approximation to the slope of the curve $\mathbf{z}(s)$ in the center cell C_{ij}. These algorithms are the ones named LVIRA and ELVIRA in [23]. However this remains to be proven. Computational studies in [23] show that these algorithms are second-order accurate in the discrete max norm when the results are averaged over many (for example, one thousand) computations. However these algorithms may need to be modified in order to achieve strict second-order accuracy in the max norm without averaging.

In Theorem 24, I have proven that (126) holds provided that the maximum value

$$\kappa_{\max} = \max_{s} |\kappa(s)|$$

of the curvature $\kappa(s)$ of the interface $\mathbf{z}(s)$ in the 3×3 block of cells B_{ij} satisfies

$$\kappa_{\max} \leq C_\kappa \equiv \min\{C_h h^{-1}, (\sqrt{h})^{-1}\}, \tag{127}$$

where C_h is a constant that is independent of h. As $h \to 0$ the second constraint in (127) eventually becomes the condition that must be satisfied; that is, $(\sqrt{h})^{-1} < C_h h^{-1}$ for h small enough. It is natural to ask if this constraint is necessary, since I only need this constraint when the center column sum S_i is not exact; that is, I only use the constraint $\kappa_{\max} \leq (\sqrt{h})^{-1}$ to prove Theorem 15.

I have performed a number of computations in an effort to determine if the first constraint

$$\kappa_{\max} \leq C_h h^{-1}$$

is sufficient to ensure that (126) holds. These computations, together with several theorems I have proven in special cases when the center column sum S_i is not exact,[12] lead me to believe that the second constraint in (127)

$$\kappa_{\max} \le (\sqrt{h})^{-1}$$

is indeed necessary. However this issue requires further study.

In closing, I would like to emphasize that when the interface reconstruction algorithm is coupled to an adaptive mesh refinement algorithm, the parameter

$$H_{\max} = \min\{C_h (\kappa_{\max})^{-1}, (\kappa_{\max})^{-2}\}$$

can be used to develop a criterion for determining when to increase the resolution of the grid. Namely, the computation of the interface in a given cell C_{ij} is under-resolved whenever

$$h > H_{\max},$$

where $\kappa_{\max}$ is the maximum curvature of the interface over the 3×3 block of cells B_{ij} centered on C_{ij}, and hence the grid needs to be refined in a neighborhood of this block.

Acknowledgment

I wish to express my sincere thanks to Professor Greg Miller of the Department of Applied Science at University of California, Davis who first suggested that we collaborate on finding a proof of the convergence of the LVIRA and ELVIRA algorithms and whose ideas form the basis for the results in Section 4.

References

[1] I. Aleinov and E. G. Puckett, *Computing surface tension with high-order kernels*, Proceedings of the 6th International Symposium on Computational Fluid Dynamics (K. Oshima, ed.), American Society of Mechanical Engineers, 1995, pp. 6–13.

[2] J. U. Brackbill, D. B. Kothe, and C. Zemach, *A continuum method for modeling surface tension*, J. Comput. Phys. **100** (1992), no. 2, 335–354. MR 93c:76008 Zbl 0775.76110

[3] A. J. Chorin, *Flame advection and propagation algorithms*, J. Comput. Phys. **35** (1980), no. 1, 1–11. MR 81d:76061 Zbl 0425.76086

[4] ———, *Curvature and solidification*, J. Comput. Phys. **57** (1985), no. 3, 472–490. MR 86d:80001 Zbl 0555.65085

[5] P. Colella, L. F. Henderson, and E. G. Puckett, *A numerical study of shock wave refraction at a gas interface*, Proceedings of the AIAA 9th Computational Fluid Dynamics Conference, 1989, pp. 426–439.

[6] J. J. Helmsen, P. Colella, E. G. Puckett, and M. R. Dorr, *Two new methods for simulating photolithography development in three dimensions*, Proceeedings of the 10th SPIE Optical/Laser Microlithography Conference, vol. 2726, SPIE, 1996, pp. 253–261.

[12] I have not included these theorems in this article.

[7] J. J. Helmsen, P. Colella, and E. G. Puckett, *Non-convex profile evolution in two dimensions using volume of fluids*, Technical report lbnl-40693, Lawrence Berkeley National Laboratory, 1997.

[8] L. F. Henderson, P. Colella, and E. G. Puckett, *On the refraction of shock waves at a slow-fast gas interface*, J. Fluid Mech. **224** (1991), 1–27.

[9] C. W. Hirt and B. D. Nichols, *Volume of fluid (VOF) method for the dynamics of free boundaries*, Journal of Computational Physics **39** (1981), 201–225.

[10] R. M. Hurst, *Numerical approximations to the curvature and normal of a smooth interface using high-order kernels*, MS thesis, University of California, Davis, 1995.

[11] D. R. Korzekwa, D. B. Kothe, K. L. Lam, E. G. Puckett, P. K. Tubesing, and M. W. Williams, *A second-order accurate, linearity-preserving volume tracking algorithm for free surface flows on 3-D unstructured meshes*, Proceedings of the 3rd ASME /JSME Joint Fluids Engineering Conference, American Society of Mechanical Engineers, 1999.

[12] D. B. Kothe, J. R. Baumgardner, S. T. Bennion, J. H. Cerutti, B. J. Daly, and K. S. Torrey, *Pagosa: A massively-parallel, multi-material hydro-dynamics model for three-dimensional high-speed flow and high-rate deformation*, technical report la-ur-92-4306, Los Alamos National Laboratory, 1992.

[13] D. B. Kothe, E. G. Puckett, and M. W. Williams, *Convergence and accuracy of kernel-based continuum surface tension models*, Fluid dynamics at interfaces (W. Shyy and R. Naranyan, eds.), Cambridge University Press, New York, 1999, pp. 347–356.

[14] D. B. Kothe, E. G. Puckett, and M. W. Williams, *Approximating interface topologies with applications to interface tracking algorithms*, Proceedings of the 37th AIAA Aerospace Sciences Meetings, American Institute of Aeronautics and Astronautics, 1999, pp. 1–9.

[15] P. Lax and B. Wendroff, *Systems of conservation laws*, Comm. Pure Appl. Math. **13** (1960), 217–237. MR 22 #11523 Zbl 0152.44802

[16] G. H. Miller and P. Colella, *A conservative three-dimensional Eulerian method for coupled solid-fluid shock capturing*, J. Comput. Phys. **183** (2002), no. 1, 26–82. MR 2003j:76080 Zbl 1057.76558

[17] G. H. Miller and E. G. Puckett, *Edge effects in molybdenum-encapsulated molten silicate shock wave targets*, J. Appl. Phys. **75** (1994), 1426–1434.

[18] ———, *A high-order godunov method for multiple condensed phases*, Journal of Computational Physics **128** (1996), 134–164.

[19] B. D. Nichols, C. W. Hirt, and R. S. Hotchkiss, *SOLA-VOF: A solution algorithm for transient fluid flow with multiple free boundaries*, technical report la-8355, Los Alamos National Laboratory, 1980.

[20] W. F. Noh and P. R. Woodward, *SLIC (Simple line interface calculation)*, technical teport ucrl-77651, Los Alamos National Laboratory, 1976.

[21] ———, *SLIC (Simple line interface calculation)*, Lecture Notes in Physics, no. 59, Springer, New York, 1976.

[22] J. E. Pilliod, *An analysis of piecewise linear interface reconstruction algorithms for volume-of-fluid methods*, MS thesis, University of California, Davis, 1992.

[23] J. E. Pilliod, Jr. and E. G. Puckett, *Second-order accurate volume-of-fluid algorithms for tracking material interfaces*, J. Comput. Phys. **199** (2004), no. 2, 465–502. MR 2005d:65145 Zbl 1126.76347

[24] S. Popinet and S. Zaleski, *A front-tracking algorithm for accurate representation of surface tension*, Int. J. Numer. Methods Fluids **30** (1999), 775–793. Zbl 0940.76047

[25] E. G. Puckett, *A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction*, Proceedings of the 4th International Symposium on Computational Fluid Dynamics (H. Dwyer, ed.), 1991, pp. 933–938.

[26] E. G. Puckett, L. F. Henderson, and P. Colella, *A general theory of anomalous refraction*, Shock Waves at Marseilles (R. Brun and L. Z. Dumitrescu, eds.), vol. 4, Springer, 1995, pp. 139–144.

[27] W. J. Rider and D. B. Kothe, *Reconstructing volume tracking*, J. Comput. Phys. **141** (1998), no. 2, 112–152. MR 99a:65200 Zbl 0933.76069

[28] R. Scardovelli and S. Zaleski, *Direct numerical simulation of free-surface and interfacial flow*, Annu. Rev. Fluid Mech. (1999), no. 31, 567–603. MR 99m:76002

[29] S. K. Stein and A. Barcellos, *Calculus and analytic geometry*, 5th ed., McGraw-Hill, 1992. Zbl 0375.26001

[30] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, *An adaptive level set approach for incompressible two-phase flows*, Proceedings of the 1996 ASME Fluids Engineering Summer Meeting (San Diego, CA), American Society of Mechanical Engineers, 1996, pp. 355–360.

[31] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, *An adaptive level set approach for incompressible two-phase flows*, J. Comput. Phys. **148** (1999), no. 1, 81–124. MR 99m:76098 Zbl 0930.76068

[32] M. Sussman and M. Ohta, *Improvements for calculating two-phase bubble and drop motion using an adaptive sharp interface method*, FDMP Fluid Dyn. Mater. Process. **3** (2007), no. 1, 21–36. MR 2008a:76117 Zbl 1153.76436

[33] M. Sussman and 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), no. 2, 301–337. MR 2001c:76099 Zbl 0977.76071

[34] M. M. Sussman, *An adaptive mesh algorithm for free surface flows in general geometries*, Adaptive method of lines (A. V. Wouwer, P. Saucez, and W. E. Shiesser, eds.), Chapman & Hall/CRC, New York, 2001, pp. 207–213. MR 2002c:65004

[35] M. D. Torrey, L. D. Cloutman, R. C. Mjolsness, and C. W. Hirt, *NASA-VOF2D: A computer program for incompressible flows with free surfaces*, technical report LA-10612-MS, Los Alamos National Laboratory, December 1985.

[36] M. D. Torrey, R. C. Mjolsness, and L. R. Stein, *NASA-VOF3D: A three-dimensonal computer program for incompressible flows with free surfaces*, technical report LA-11009-MS, Los Alamos National Laboratory, 1987.

[37] M. W. Williams, *Numerical methods for tracking interfaces with surface tension in 3-D mold-filling processes*, Ph.D. thesis, University of California, Davis, 2000.

Received June 12, 2009.

ELBRIDGE GERRY PUCKETT: egpuckett@ucdavis.edu
Department of Mathematics, University of California, Davis, CA 95616, United States

Guidelines for Authors

Authors may submit manuscripts in PDF format on-line at the Submission page at pjm.math.berkeley.edu/camcos.

Originality. Submission of a manuscript acknowledges that the manuscript is original and and is not, in whole or in part, published or under consideration for publication elsewhere. It is understood also that the manuscript will not be submitted elsewhere while under consideration for publication in this journal.

Language. Articles in CAMCoS are usually in English, but articles written in other languages are welcome.

Required items. A brief abstract of about 150 words or less must be included. It should be self-contained and not make any reference to the bibliography. If the article is not in English, two versions of the abstract must be included, one in the language of the article and one in English. Also required are keywords and subject classifications for the article, and, for each author, postal address, affiliation (if appropriate), and email address.

Format. Authors are encouraged to use LaTeX but submissions in other varieties of TeX, and exceptionally in other formats, are acceptable. Initial uploads should be in PDF format; after the refereeing process we will ask you to submit all source material.

References. Bibliographical references should be complete, including article titles and page ranges. All references in the bibliography should be cited in the text. The use of BibTeX is preferred but not required. Tags will be converted to the house format, however, for submission you may use the format of your choice. Links will be provided to all literature with known web locations and authors are encouraged to provide their own links in addition to those supplied in the editorial process.

Figures. Figures must be of publication quality. After acceptance, you will need to submit the original source files in vector graphics format for all diagrams in your manuscript: vector EPS or vector PDF files are the most useful.

Most drawing and graphing packages (Mathematica, Adobe Illustrator, Corel Draw, MATLAB, etc.) allow the user to save files in one of these formats. Make sure that what you are saving is vector graphics and not a bitmap. If you need help, please write to graphics@mathscipub.org with details about how your graphics were generated.

White Space. Forced line breaks or page breaks should not be inserted in the document. There is no point in your trying to optimize line and page breaks in the original manuscript. The manuscript will be reformatted to use the journal's preferred fonts and layout.

Proofs. Page proofs will be made available to authors (or to the designated corresponding author) at a Web site in PDF format. Failure to acknowledge the receipt of proofs or to return corrections within the requested deadline may cause publication to be postponed.

Communications in Applied Mathematics and Computational Science

vol. 5 no. 1 2010