INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2009; **60**:633–649
Published online 10 September 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1907

A parallel computational method for simulating two-phase gel dynamics on a staggered grid

Jian Du[1,*,†], Aaron L. Fogelson[1,2] and Grady B. Wright[3]

[1]*Department of Mathematics, University of Utah, Salt Lake City, UT 84112-0090, U.S.A.*
[2]*Department of Bioengineering, University of Utah, Salt Lake City, UT 84112-0090, U.S.A.*
[3]*Department of Mathematics, Boise State University, Boise, ID 83725-1555, U.S.A.*

SUMMARY

We develop a parallel computational algorithm for simulating models of gel dynamics where the gel is described by two phases, a networked polymer and a fluid solvent. The models consist of transport equations for the two phases, two coupled momentum equations, and a volume-averaged incompressibility constraint. Multigrid with Vanka-type box-relaxation scheme is used as preconditioner for the Krylov subspace solver (GMRES) to solve the momentum and incompressibility equations. Through numerical experiments of a model problem, the efficiency, robustness and scalability of the algorithm are illustrated.

Received 3 April 2008; Revised 19 July 2008; Accepted 22 July 2008

KEY WORDS: multiphase flow; multigrid; Vanka relaxation; GMRES; preconditioning

1. INTRODUCTION

Polymer gels are composed of two materials: a polymer network and a solvent. By weight and volume gels are mostly solvent, but the rheology of the gel can range from a very viscous fluid to an elastic solid. In addition to viscoelastic stresses, gels can exhibit chemical stresses, which result in swelling and deswelling behavior.

A commonly used model for the mechanics of gels is the two-phase flow (or two-fluid) model [1–7]. Each phase, network and solvent, is treated as a continuum and moves according to its own velocity field. Each region in space is composed of a mixture of the two phases that is described by the volume fractions of the phases. The equations of motion for the gel consist of two coupled momentum equations and two continuity equations. These models of gels are based on mixture

*Correspondence to: Jian Du, 155 South 1400 East Room 233, Salt Lake City, UT 84112-0090, U.S.A.
†E-mail: du@math.utah.edu

Contract/grant sponsor: NSF; contract/grant number: DMS-0540779
Contract/grant sponsor: Center for High Performance Computing at the University of Utah

theory [8, 9], which has been used for many applications and is becoming increasingly popular in models of biological materials such as tissue [10], tumors [11, 12], cytoplasm [1, 3, 4, 6], and biofilms [2, 13, 14]. While much work has gone into advancing these models, the same is not true for the development of efficient numerical methods for simulating them. Since repeated simulation of models is crucial for validating their usefulness, the need for fast and robust computational methods is essential. A significant step in this direction was described in [15] where a novel serial algorithm for a subdivision of the problems was introduced.

In this paper we extend the work in [15] by proposing a parallel computational methodology for solving the same problems. Both the network and solvent are modeled as viscous fluids. The viscosity of the network is much larger than that of the solvent. The gel feels a chemical pressure that makes the system bistable, that is, it drives the network volume fraction to one of two preferred states. Under appropriate conditions the chemical pressure induces phase separation of the mixture and it is a good test for the robustness of any numerical procedure. Computational tests of the proposed method are carried out for the model in two dimensions, from which the efficiency and robustness of the algorithm are investigated.

In many models of gel dynamics, including our model problem, viscous terms are assumed to dominate so that inertial terms are negligible. This assumption leads to an elliptic system for the coupled momentum and incompressibility equations that resembles Stokes equations. However, there are many key differences, which increase the complexity of the problem. First, since there are two phases, there are two sets of momentum equations. Second, there are off-diagonal terms coupling the two velocity fields (and the individual components of the two fields) together. Third, the Laplacians in Stokes equations are replaced by elliptic operators that involves time-dependent variable coefficients related to the volume fractions of the two phases. Finally, incompressibility is replaced with a 'volume-averaged' incompressibility over the two fluids.

We use second-order finite differences on a marker-and-cell (MAC) grid [16] to discretize this system. As with Stokes (and linearized Navier–Stokes) equations, this gives rise to a large, sparse linear system of saddle point type. In [15], we developed an extension of a multigrid method initially proposed for Stokes and Navier–Stokes equations by Vanka [17], and which has seen considerable development in the past several years (see, for example, [18–23]). The method is characterized by the smoother used and is referred to in literature as box [22, pp. 320–322], coupled [23], or Vanka [19] relaxation. The basic idea of the smoother is to compute updates to the solution by collectively solving for the velocity and pressure in the discrete momentum equations locally, computational cell (or box) by cell. For the gel system, this means solving a $(2^{d+1}+1)$-by-$(2^{d+1}+1)$ saddle point system for each cell, where d is the number of spatial dimensions. Depending on how the cells are processed (like Jacobi or Gauss–Seidel), the method can also be viewed as an additive or multiplicative Schwarz domain decomposition method, where the subdomains consist of a single computational cell [19]. We use the smoother with standard prolongation and restriction operators and do a direct discretization of the equations on the coarse grids, which makes the implementation relatively simple.

When the network and solvent are well mixed, the multigrid box-relaxation method performs quite well as a solver for the model problem considered here. However, as phase separation occurs, the solver performance degrades quite dramatically, and it, in fact, fails in some cases as shown in [15]. One particularly effective way of improving the robustness of a non-optimal (or even non-converging) multigrid method is to combine it with a Krylov subspace method (see, for example, [22, Section 7.8]). In the current paper, a parallel version of the algorithm in [15] is developed,

DOI: 10.1002/fld

which combines the Krylov subspace solver in PETSc [24] with the multigrid box-relaxation scheme as the preconditioner.

In addition to the momentum and incompressibility equations, models for gel dynamics include equations for the transport of the network and solvent. For the transport equations we use a conservative finite-volume discretization in which the advection term is treated explicitly with first-order upwinding and the diffusion term is handled implicitly with backward Euler.

The remainder of this paper is organized as follows. In Section 2, the model problem for describing and testing the computational method is introduced. Section 3 describes the discretization (in space and time) for the 2-D model. In Section 4, the elements of the multigrid box-relaxation scheme are introduced with parallel considerations, followed by a description of its role as a preconditioner for GMRES (generalized minimum residual). Section 5 presents the results of simulations as well as the parallel performance.

2. MODEL PROBLEM

The model problem we consider is for a gel composed of two immiscible materials, a polymer network and a fluid solvent. We assume that the total amount of gel remains constant and that the transport of the network and solvent is governed by the following equations:

$$\frac{\partial \theta^{\mathrm{n}}}{\partial t} + \nabla \cdot (\theta^{\mathrm{n}} \mathbf{u}^{\mathrm{n}}) = 0 \tag{1}$$

$$\frac{\partial \theta^{\mathrm{s}}}{\partial t} + \nabla \cdot (\theta^{\mathrm{s}} \mathbf{u}^{\mathrm{s}}) = 0 \tag{2}$$

where θ^{n} and $\theta^{\mathrm{s}} = 1 - \theta^{\mathrm{n}}$ are the respective volume fractions of the network and solvent, $0 < \theta^{\mathrm{n}} < 1$, and $\mathbf{u}^{\mathrm{n}}$ and $\mathbf{u}^{\mathrm{s}}$ are the respective transport velocities. Adding these two equations and using $\theta^{\mathrm{n}} + \theta^{\mathrm{s}} = 1$ give the incompressibility-type constraint

$$\nabla \cdot (\theta^{\mathrm{n}} \mathbf{u}^{\mathrm{n}} + \theta^{\mathrm{s}} \mathbf{u}^{\mathrm{s}}) = 0 \tag{3}$$

The transport velocities are determined by conservation of momentum. We assume that the network acts as a constant density viscous material and the solvent acts as a Newtonian fluid of much less viscosity. Viscous terms are assumed to dominate so that inertial terms are negligible (similar to zero-Reynolds number flow), i.e. the system responds instantaneously to applied forces.

The network and solvent are each subject to a number of intraphase stresses. We assume that the viscous stress tensors $\underline{\underline{\sigma}}^{\mathrm{n}}$ and $\underline{\underline{\sigma}}^{\mathrm{s}}$ for the network and solvent are proportional to the respective gradient of the network and solvent velocities, i.e.

$$\underline{\underline{\sigma}}^{\mathrm{n}} = \mu_{\mathrm{n}} (\nabla \mathbf{u}^{\mathrm{n}} + \nabla \mathbf{u}^{\mathrm{n}\mathrm{T}}) + \lambda_n \delta_{ij} \nabla \cdot \mathbf{u}^{\mathrm{n}} \tag{4}$$

$$\underline{\underline{\sigma}}^{\mathrm{s}} = \mu_{\mathrm{s}} (\nabla \mathbf{u}^{\mathrm{s}} + \nabla \mathbf{u}^{\mathrm{s}\mathrm{T}}) + \lambda_s \delta_{ij} \nabla \cdot \mathbf{u}^{\mathrm{s}} \tag{5}$$

where $\mu_{\mathrm{n,s}}$ are shear viscosities and $\lambda_{\mathrm{n,s}} + 2\mu_{\mathrm{n,s}}/d$ are the bulk viscosities of the network and solvent (d is the dimension). The network and solvent are also subject to a frictional drag since the motion of the solvent influences the network. We model this by $\beta \theta^{\mathrm{n}} \theta^{\mathrm{s}} (\mathbf{u}^{\mathrm{n}} - \mathbf{u}^{\mathrm{s}})$, where $\beta > 0$ is the drag coefficient. The third force on each phase is due to hydrostatic pressure. Since the polymer

DOI: 10.1002/fld

is assumed to be chemically active within the gel and the solvent is assumed to be chemically neutral, the final force is generated by a chemical pressure, $\Psi(\theta^{\mathrm{n}})$, and acts only on the network. The form of $\Psi(\theta^{\mathrm{n}})$ used in this paper is described below.

Balancing the above forces on the network and solvent yields the following equations:

$$\nabla\cdot(\theta^{\mathrm{n}}\underline{\underline{\sigma}}^{\mathrm{n}})-\beta\theta^{\mathrm{n}}\theta^{\mathrm{s}}(\mathbf{u}^{\mathrm{n}}-\mathbf{u}^{\mathrm{s}})-\theta^{\mathrm{n}}\nabla p=\nabla\Psi(\theta^{\mathrm{n}}) \tag{6}$$

$$\nabla\cdot(\theta^{\mathrm{s}}\underline{\underline{\sigma}}^{\mathrm{s}})-\beta\theta^{\mathrm{n}}\theta^{\mathrm{s}}(\mathbf{u}^{\mathrm{s}}-\mathbf{u}^{\mathrm{n}})-\theta^{\mathrm{s}}\nabla p=0 \tag{7}$$

where p is the hydrostatic pressure. These two equations, combined with (1) and (3), and subject to suitable boundary conditions, govern the gel dynamics and are the same as used in [1, 3, 4, 6, 15]. For our test problem we assume no-slip boundary conditions for the network and solvent velocities, and no-flux boundary conditions for θ^{n} and θ^{s}.

The chemical pressure includes osmotic pressure, but it may also include active, contractile stresses such as in the actomyosin gels of cytoplasm. In this paper we are not concerned with the origins of the chemical stress and refer to it simply as the osmotic pressure. We assume that the osmotic pressure is of the form

$$\Psi(\theta^{\mathrm{n}})=\gamma\theta^{\mathrm{n}}(\theta^{\mathrm{n}}-\theta_0^{\mathrm{n}})(\theta^{\mathrm{n}}-\theta_*^{\mathrm{n}}) \tag{8}$$

where $\gamma>0$, $0<\theta_0^{\mathrm{n}}<\theta_*^{\mathrm{n}}<1$; see Figure 1 for a plot of $\Psi(\theta^{\mathrm{n}})$ with parameters used in the numerical experiments that follow. This functional form is chosen since it can produce phase separation, or channeling. In regions of space where $\Psi'(\theta^{\mathrm{n}})>0$ the mixture is stable, but where $\Psi'(\theta^{\mathrm{n}})<0$ the mixture tends to phase separate; see Cogan and Keener [2] for an analysis. This form of Ψ does not allow the phases to separate completely. Phase separation is generally observed in gels [7] and is vital for locomotion and for transporting nutrients in some amoeboid cells [25].

Finally, we assume that there is a small amount of diffusion between the network and solvent in the gel. This is mathematically and computationally useful since it means that the transition

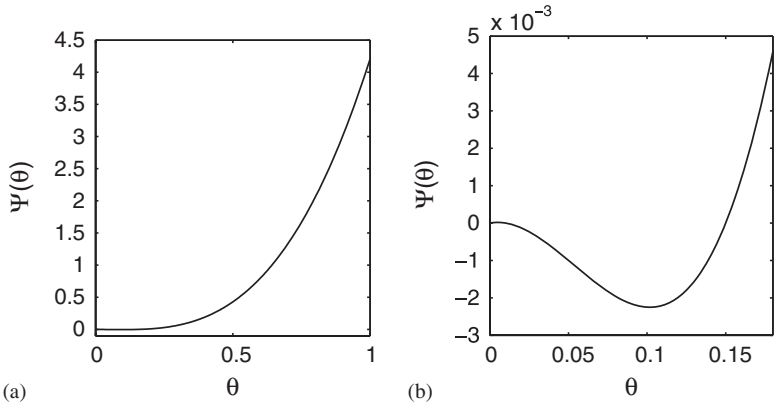


Figure 1. (a) Osmotic pressure function (8) from the model problem with parameters $\gamma=5$, $\theta_0^{\mathrm{n}}=0.01$ and $\theta_*^{\mathrm{n}}=0.15$ used in the numerical experiments. (b) A detailed plot of this function near the origin.

DOI: 10.1002/fld

between the network and solvent will be smooth even in areas where the two phases have separated. This additional assumption changes (1) and (2) to

$$\frac{\partial \theta^{\mathrm{n}}}{\partial t}+\nabla\cdot(\theta^{\mathrm{n}}\mathbf{u}^{\mathrm{n}})=\kappa\nabla^2\theta^{\mathrm{n}} \tag{9}$$

$$\frac{\partial \theta^{\mathrm{s}}}{\partial t}+\nabla\cdot(\theta^{\mathrm{s}}\mathbf{u}^{\mathrm{s}})=\kappa\nabla^2\theta^{\mathrm{s}} \tag{10}$$

where $\kappa\geqslant 0$ is the diffusion constant. We use relatively small diffusion coefficients so that this modification does not change the qualitative features of the model, but ensures the solution is continuous.

3. DISCRETIZATION

Let $\mathbf{u}^{\mathrm{n}}=(u^{\mathrm{n}},v^{\mathrm{n}})^{\mathrm{T}}$ and $\mathbf{u}^{\mathrm{s}}=(u^{\mathrm{s}},v^{\mathrm{s}})^{\mathrm{T}}$, where u^{n}, v^{n} and u^{s}, v^{s} are the respective network and solvent velocity components in the x and y directions, and let the spatial domain be $\Omega=\{(x,y)|0\leqslant x\leqslant a, 0\leqslant y\leqslant b\}$. Then the momentum equations (6)–(7) and volume-averaged incompressibility (3) are given in matrix–vector form as

$$\begin{bmatrix} \mathscr{L}_{\mathrm{n}}-\mathscr{C} & \mathscr{C} & -\mathscr{G}_{\mathrm{n}} \\ \mathscr{C} & \mathscr{L}_{\mathrm{s}}-\mathscr{C} & -\mathscr{G}_{\mathrm{s}} \\ -\mathscr{D}_{\mathrm{n}}^{\mathrm{T}} & -\mathscr{D}_{\mathrm{s}}^{\mathrm{T}} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{u}^{\mathrm{n}} \\ \mathbf{u}^{\mathrm{s}} \\ p \end{bmatrix}=\begin{bmatrix} \nabla\Psi(\theta^{\mathrm{n}}) \\ 0 \\ 0 \end{bmatrix} \tag{11}$$

where

$$\mathscr{L}_{\mathrm{n,s}}=\begin{bmatrix} \alpha_{\mathrm{n,s}}\partial_x(\theta^{\mathrm{n,s}}\partial_x)+\mu_{\mathrm{n,s}}\partial_y(\theta^{\mathrm{n,s}}\partial_y) & \mu_{\mathrm{n,s}}\partial_y(\theta^{\mathrm{n,s}}\partial_x)+\lambda_{\mathrm{n,s}}\partial_x(\theta^{\mathrm{n,s}}\partial_y) \\ \mu_{\mathrm{n,s}}\partial_x(\theta^{\mathrm{n,s}}\partial_y)+\lambda_{\mathrm{n,s}}\partial_y(\theta^{\mathrm{n,s}}\partial_x) & \alpha_{\mathrm{n,s}}\partial_y(\theta^{\mathrm{n,s}}\partial_y)+\mu_{\mathrm{n,s}}\partial_x(\theta^{\mathrm{n,s}}\partial_x) \end{bmatrix}$$

$$\mathscr{C}=\begin{bmatrix} \beta\theta^{\mathrm{n}}\theta^{\mathrm{s}} & 0 \\ 0 & \beta\theta^{\mathrm{n}}\theta^{\mathrm{s}} \end{bmatrix},\quad \mathscr{G}_{\mathrm{n,s}}=\begin{bmatrix} \theta^{\mathrm{n,s}}\partial_x \\ \theta^{\mathrm{n,s}}\partial_y \end{bmatrix},\quad \mathscr{D}_{\mathrm{n,s}}=\begin{bmatrix} \partial_x\theta^{\mathrm{n,s}} \\ \partial_y\theta^{\mathrm{n,s}} \end{bmatrix}$$

and $\alpha_{n,s}=(2\mu_{n,s}+\lambda_{n,s})$. Since $\theta^{\mathrm{s}}=1-\theta^{\mathrm{n}}$, we need only an equation for the transport of θ^{n}, which is again given by (9). We use no-slip boundary conditions for (11) and no-flux for (9). Finally, the cubic function (8) is used for modeling the osmotic pressure.

For the spatial discretization, we use a MAC grid where the positions of the unknowns are indicated in Figure 2. The mesh-spacing in the x and y direction is set equal and is given by h.

All equations in (11) are discretized using second-order, centered finite differences, which leads to the following approximation of the first row of (11) at the interior point $(x_{i+1/2,j}, y_{i+1/2,j})$:

$$\begin{aligned} &\frac{\alpha_{\mathrm{n}}}{h^2}[\theta^{\mathrm{n}}_{i+1,j}(u^{\mathrm{n}}_{i+3/2,j}-u^{\mathrm{n}}_{i+1/2,j})-\theta^{\mathrm{n}}_{i,j}(u^{\mathrm{n}}_{i+1/2,j}-u^{\mathrm{n}}_{i-1/2,j})] \\ &\quad+\frac{\mu_{\mathrm{n}}}{h^2}[\overline{\theta}^{\mathrm{n}}_{i+1/2,j+1/2}(u^{\mathrm{n}}_{i+1/2,j+1}-u^{\mathrm{n}}_{i+1/2,j})-\overline{\theta}^{\mathrm{n}}_{i+1/2,j-1/2}(u^{\mathrm{n}}_{i+1/2,j}-u^{\mathrm{n}}_{i+1/2,j-1})] \\ &\quad+\frac{\mu_{\mathrm{n}}}{h^2}[\overline{\theta}^{\mathrm{n}}_{i+1/2,j+1/2}(v^{\mathrm{n}}_{i+1,j+1/2}-v^{\mathrm{n}}_{i,j+1/2})-\overline{\theta}^{\mathrm{n}}_{i+1/2,j-1/2}(v^{\mathrm{n}}_{i+1,j-1/2}-v^{\mathrm{n}}_{i,j-1/2})] \end{aligned}$$

DOI: 10.1002/fld

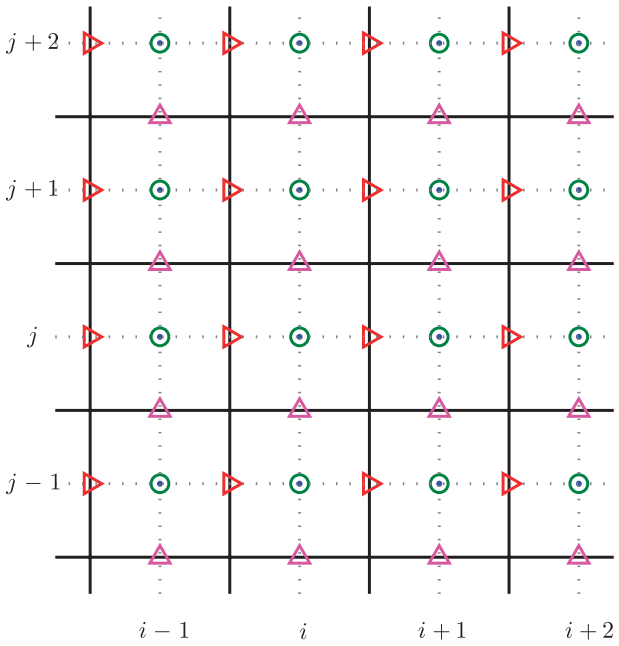


Figure 2. Location of the unknowns in the MAC grid for the 2-D gel model: $\triangleright$ = network/solvent horizontal velocity, $\triangle$ = network/solvent vertical velocity, $\bullet$ = pressure, and $\bigcirc$ = network/solvent volume fractions.

$$+\frac{\lambda_n}{h^2}[\theta^n_{i+1,j}(v^n_{i+1,j+1/2}-v^n_{i+1,j-1/2})-\theta^n_{i,j}(v^n_{i,j+1/2}-v^n_{i,j-1/2})]$$
$$-\beta\overline{\theta}^n_{i+1/2,j}\overline{\theta}^s_{i+1/2,j}(u^n_{i+1/2,j}-u^s_{i+1/2,j})-\overline{\theta}^n_{i+1/2,j}\frac{p_{i+1,j}-p_{i,j}}{h}=\frac{\Psi(\theta^n_{i+1,j})-\Psi(\theta^n_{i,j})}{h} \tag{12}$$

while the approximation to the second row at the interior point $(x_{i,j+1/2}, y_{i,j+1/2})$ is given by

$$\frac{\mu_n}{h^2}[\overline{\theta}^n_{i+1/2,j+1/2}(u^n_{i+1/2,j+1}-u^n_{i+1/2,j})-\overline{\theta}^n_{i-1/2,j+1/2}(u^n_{i-1/2,j+1}-u^n_{i-1/2,j})]$$
$$+\frac{\lambda_n}{h^2}[\theta^n_{i,j+1}(u^n_{i+1/2,j+1}-u^n_{i-1/2,j+1})-\theta^n_{i,j}(u^n_{i+1/2,j}-u^n_{i-1/2,j})]$$
$$+\frac{\alpha_n}{h^2}[\theta^n_{i,j+1}(v^n_{i,j+3/2}-v^n_{i,j+1/2})-\theta^n_{i,j}(v^n_{i,j+1/2}-v^n_{i,j-1/2})]$$
$$+\frac{\mu_n}{h^2}[\overline{\theta}^n_{i+1/2,j+1/2}(v^n_{i+1,j+1/2}-v^n_{i,j+1/2})-\overline{\theta}^n_{i-1/2,j+1/2}(v^n_{i,j+1/2}-v^n_{i-1,j+1/2})]$$
$$-\beta\overline{\theta}^n_{i,j+1/2}\overline{\theta}^s_{i,j+1/2}(v^n_{i,j+1/2}-v^s_{i,j+1/2})-\overline{\theta}^n_{i,j+1/2}\frac{p_{i,j+1}-p_{i,j}}{h}=\frac{\Psi(\theta^n_{i,j+1})-\Psi(\theta^n_{i,j})}{h} \tag{13}$$

DOI: 10.1002/fld

Bars over θ^{n} and θ^{s} represent arithmetic averages of the values of these variables at nearest neighbor cells with two-point averages when there is a mix of integer and half-integer indices, and four-point averages when there are two half-integer indices. The discretizations of the third and fourth row of (11) are the same, but with variables for the network replaced accordingly by the variables for the solvent. Finally, the last row of (11) is approximated at $(x_{i,j}, y_{i,j})$ by

$$\frac{-\overline{\theta}^{\mathrm{n}}_{i+1/2,j}u^{\mathrm{n}}_{i+1/2,j}+\overline{\theta}^{\mathrm{n}}_{i-1/2,j}u^{\mathrm{n}}_{i-1/2,j}}{h}+\frac{-\overline{\theta}^{\mathrm{n}}_{i,j+1/2}v^{\mathrm{n}}_{i,j+1/2}+\overline{\theta}^{\mathrm{n}}_{i,j-1/2}v^{\mathrm{n}}_{i,j-1/2}}{h}$$
$$+\frac{-\overline{\theta}^{\mathrm{s}}_{i+1/2,j}u^{\mathrm{s}}_{i+1/2,j}+\overline{\theta}^{\mathrm{s}}_{i-1/2,j}u^{\mathrm{s}}_{i-1/2,j}}{h}+\frac{-\overline{\theta}^{\mathrm{s}}_{i,j+1/2}v^{\mathrm{s}}_{i,j+1/2}+\overline{\theta}^{\mathrm{s}}_{i,j-1/2}v^{\mathrm{s}}_{i,j-1/2}}{h}=0 \quad (14)$$

where necessary, we use second-order extrapolation to account for the no-slip boundary conditions for velocity field. As in Figure 2, suppose that $x=x_{i-3/2}$ is the physical domain boundary. Thus, $v^{\mathrm{n}}_{i-2,j+1/2}$ is located on the ghost cell outside the domain and $v^{\mathrm{n}}_{i-3/2,j+1/2}=0$ according to the no-slip boundary conditions. By fitting a quadric polynomial at locations $(i, j+\frac{1}{2})$, $(i-1, j+\frac{1}{2})$ and $(i-\frac{3}{2}, j+\frac{1}{2})$ with the corresponding y-velocity values, it is easy to get $v^{\mathrm{n}}_{i-2,j+1/2}=\frac{1}{3}v^{\mathrm{n}}_{i,j+1/2}-2v^{\mathrm{n}}_{i-1,j+1/2}$ by extrapolation.

For a grid with N cell centers in the x direction and M cell centers in the y direction, the above approximations can be collected in a $(5NM-2(N+M))$-by-$(5NM-2(N+M))$ linear system, which we denote by

$$\underbrace{\begin{bmatrix} \mathscr{L}^h_{\mathrm{n}}-\mathscr{C}^h & \mathscr{C}^h & -\mathscr{G}^h_{\mathrm{n}} \\ \mathscr{C}^h & \mathscr{L}^h_{\mathrm{s}}-\mathscr{C}^h & -\mathscr{G}^h_{\mathrm{s}} \\ \mathscr{G}^{h\mathrm{T}}_{\mathrm{n}} & \mathscr{G}^{h\mathrm{T}}_{\mathrm{s}} & 0 \end{bmatrix}}_{\mathscr{A}^h} \begin{bmatrix} \mathbf{u}^{\mathrm{n}} \\ \mathbf{u}^{\mathrm{s}} \\ p \end{bmatrix} = \begin{bmatrix} \nabla^h\Psi(\theta^{\mathrm{n}}) \\ 0 \\ 0 \end{bmatrix} \quad (15)$$

This forms the discrete approximation to (11). Since we assume $0<\theta^{\mathrm{n}}<1$, it follows, for example, from [26, Theorem 3.6] that the eigenvalues of $\mathscr{A}^h$ have non-positive real part ($\mathscr{A}^h$ is negative semistable), i.e. $Re(\lambda)\leqslant 0$ for all $\lambda\in\sigma(\mathscr{A}^h)$. This can be advantageous for (preconditioned) Krylov subspace methods [26].

For the temporal discretization of (9), we use explicit first-order upwinding for the advective term $\nabla\cdot(\theta^{\mathrm{n}}\mathbf{u}^{\mathrm{n}})$ (specifically we use LeVeque's first-order upwind with transverse wave corrections [27]), and we treat the diffusion implicitly with backward Euler. The linear system that arises because of the implicit discretization is solved in parallel using preconditioned GMRES solver in PETSc with block Jacobi as the preconditioner, and the cost is negligible relative to the cost for solving the momentum equations. For efficiency and stability, we use the adaptive time-stepping procedure described in [15].

The basic strategy we use for simulating the 2-D gel is as follows:

1. For a given θ^{n} at time t, solve the discrete system (15) for $\mathbf{u}^{\mathrm{n}}$, $\mathbf{u}^{\mathrm{s}}$, and p at time t.
2. Solve for θ^{n} (and thus θ^{s}) at time $t+\Delta t$ using upwinding for (9), as discussed in the previous paragraph, with the value of $\mathbf{u}^{\mathrm{n}}$ at time t.
3. Repeat step 1, with the θ^{n} at $t+\Delta t$.

DOI: 10.1002/fld

4. SOLVING THE COUPLED MOMENTUM AND CONTINUITY EQUATIONS

To solve the discrete system (15), we use a collective (box) relaxation scheme within a multigrid procedure [17]. The box-relaxation scheme updates unknowns locally, cell-by-cell, by solving a succession of small linear systems. The updated unknowns are used in a Gauss–Seidel manner as soon as they are available. The key elements of the algorithm are described below.

4.1. Smoother

For the 2-D system (11) on the MAC grid, the box relaxation involves solving the discrete equations (15) locally in each computational cell (or box). For each interior box, this requires solving a 9-by-9 linear system (four equations for the network velocity, four equations for the solvent velocity, and one equation for the pressure) for corrections to the unknowns $\mathbf{u}^{\mathrm{n}}$, $\mathbf{u}^{\mathrm{s}}$, and p. With Dirichlet boundary conditions, boxes at the corners of the domain require solving 5-by-5 linear systems, while boxes on the edges require solving 7-by-7 systems. We update corrections to the unknowns in a Gauss–Seidel-type manner and combine this with under-relaxation with $\omega = 0.675$. Suppose that the linear system on a single box is $L\mathbf{x} = \mathbf{b}$, where L, $\mathbf{x}$ and $\mathbf{b}$ are the coefficient matrix, unknown vector and right-hand side, respectively. Then after the $(k+1)$th box relaxation

$$\mathbf{x}^{k+1} = (1-\omega)\mathbf{x}^k + \omega L^{-1}\mathbf{b} \tag{16}$$

The boxes are processed using red–black ordering. Note that vector values located at cell edges are relaxed twice while scalar values located at cell centers are relaxed once within each iteration. Parallelization similar to [18] is developed using the message passing interface (MPI) for communication on distributed memory computers.

The parallel box relaxation works as follows. Suppose we have a portion of our 2-D staggered grid on a specific processor as in Figure 3, with one layer of buffer cells marked by dashed lines.

At the beginning of the sweep, the values stored in the buffer cells are current. Let (i, j) be the global cell index, then cells with $(i+j) \bmod 2 = 0$ are marked as red (white squares in the figure) and those with $(i+j) \bmod 2 = 1$ are marked as black (shaded squares in the figure). During the red sweep, the red cells in the subdomain are updated in lexicographic order. The updated velocity components (small rectangles) and pressures (circles) are shown in Figure 3(a). The red cells within the buffer are also simultaneously relaxed on neighboring processors but their state values are not yet available to this processor. After the red sweep, a two-stage communication process is carried out with adjacent processors to update the buffer states. The first stage exchanges local state values along the left and right subdomain boundaries with the corresponding neighboring processors. The updated states after the communication are cross-hatched in Figure 3(b). In order to distinguish the red and black relaxations, the states updated by the red sweep are indicated above or to the right of the cell edges and those updated by the black sweep are shown below or to the left of the edges. Updating the states in the upper and lower buffers is achieved through a similar communication step. Note that unlike the left-right communication, states on the corner cells such as those within the dashed circles in Figure 3(c) are also sent/received in order to interpolate corrections to the next finer grid for the subdomain. Explicit communications with diagonal neighbors are avoided since all the corner states have already been updated in the left-right communication step of the upper/lower processors. The black box relaxation is done in the same manner, as shown in Figure 3(d–f). It is clear that after the final communication step, all velocity components have been relaxed twice and pressure components have been relaxed once. It is also clear that all buffer cells hold

DOI: 10.1002/fld

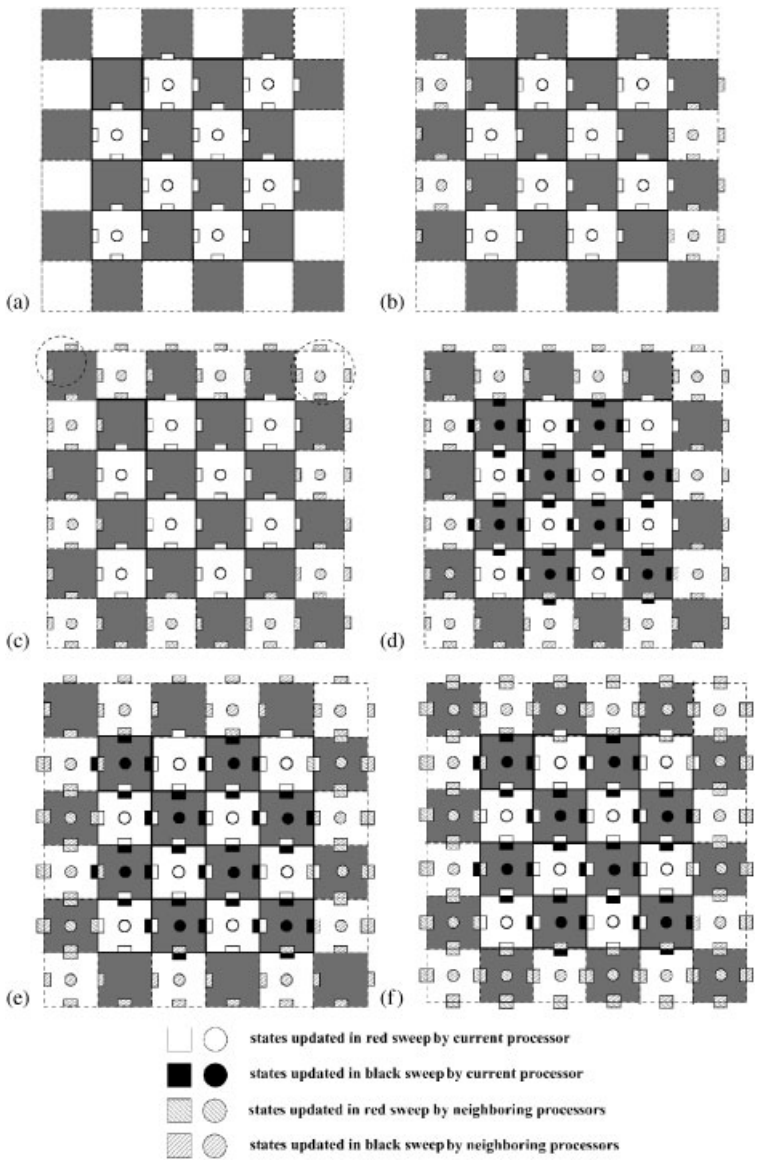


Figure 3. Progression of red–black Gauss–Seidel box-relaxation smoothing sweep. (a) After completion of the red sweep, pressures at red cell centers and all velocities on red cell edges have been updated. (b) Updated red cell values are communicated to neighboring processors to the right and left to update red cell values in subdomain buffer cells. (c) Updated red cell values are communicated to neighboring processors above and below to update red cell values in subdomain buffer cells. (d) After completion of the black sweep, pressures at black cell centers and all velocities on black cell edges have been updated. (e) Updated black cell values are communicated to neighboring processors to the right and left to update black cell values in subdomain buffer cells. (f) Updated black cell values are communicated to neighboring processors above and below to update black cell values in subdomain buffer cells.

DOI: 10.1002/fld

updated information and so the information to carry out another smoothing sweep is in place. Note also that the current values of all variables needed to compute residuals at subdomain cells are available at the end of each sweep.

4.2. Transfer operators

Suppose that the mesh-spacing in both the x and y directions is given by h, with $1/h$ a power of 2. We define a sequence of coarser grids, where each grid is a factor of two coarser than the previous in both the x and y direction. After one iteration of box relaxation, residuals r^h are calculated at the same locations on the MAC grid as the corresponding unknowns, and the right-hand sides f^{2h} of the defect equations on the next coarser grid are formed through restriction with standard full-weighting. As shown in Figure 4(a), scalar quantities f^{2h} (pressure or volume fraction) at a coarse grid center F are calculated as the arithmetic average of the related r^h values at the fine grid centers $1, 2, 3$ and 4. Six-point weighting is used for vector values at a coarse grid edge center E:

$$f^{2h}(F) = \tfrac{1}{4}[r^h(1) + r^h(2) + r^h(3) + r^h(4)] \tag{17}$$

$$f^{2h}(E) = \tfrac{1}{8}[r^h(a) + r^h(c) + r^h(d) + r^h(f) + 2r^h(b) + 2r^h(e)] \tag{18}$$

Note that communication is needed to provide current fine grid residual values with the fine grid buffers (dashed cells) such as at points a and d, before the restriction is carried out.

After reaching the coarsest grid, corrections on the next finer grid are calculated by linear/bilinear interpolation and box relaxation is carried out again with the corrected state values. In Figure 4(b), point 2 is a fine grid center, and points 3 and 4 are centers of fine grid cell edges. The corrections

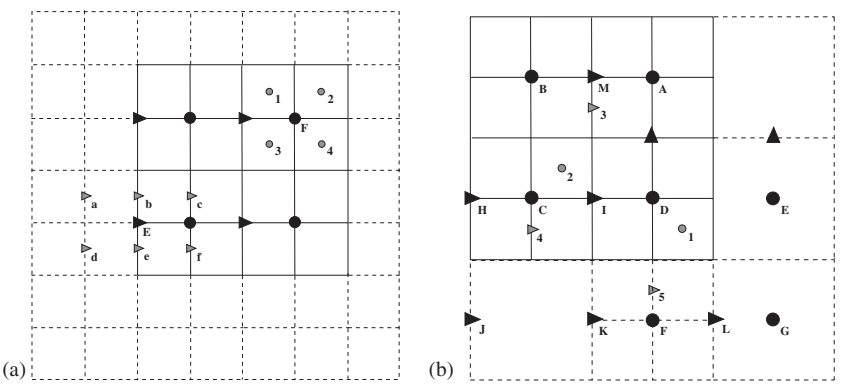


Figure 4. Multigrid transfer operators: (a) Restriction: For variables at cell centers (F), restriction is the arithmetic average of values from the neighboring fine grid points (1,2,3,4). For variables at cell edges (E), restriction uses a weighted average of values at the neighboring edges (a–f). (b) Interpolation: The value of a variable at a fine grid cell center (2) is calculated by bilinear interpolation from neighboring coarse grid cell centers (A, B, C, D). The value of variables at fine grid cell edges such as points 3 or 4 are interpolated by linear interpolation from the neighboring two (I and M) coarse grid cell edges or by bilinear interpolation from the four neighboring coarse grid cell edges (H, I, J, K).

DOI: 10.1002/fld

for the pressure (Cp_h) and the velocity (Cv_h) are calculated from the related coarse grid values as

$$Cp_h(2) = \tfrac{1}{16}[9P_{2h}(C) + 3P_{2h}(B) + 3P_{2h}(D) + P_{2h}(A)] \tag{19}$$

$$Cv_h(3) = \tfrac{1}{4}[3V_{2h}(M) + V_{2h}(I)] \tag{20}$$

$$Cv_h(4) = \tfrac{1}{8}[3V_{2h}(H) + 3V_{2h}(I) + V_{2h}(J) + V_{2h}(K)] \tag{21}$$

where P_{2h} and V_{2h} are the pressure and the velocity on coarse grid. Note that it is also necessary to interpolate corrections to those fine grid points in the buffer (e.g. point 5) which is within the processor's subdomain on the finer grid. Obviously, buffer state values at points such as E, F, G and J, K, L are needed to interpolate corrections at points 1, 4 and 5. Since these values are already available through the communication step of box relaxation, no additional communications are required for interpolation.

4.3. Parallel multigrid on coarsest grid

The relative communication overhead increases as the grid gets coarser during the multigrid cycle, and it is known that this may result in a significant loss of efficiency for the parallel application. We use the agglomeration method to overcome the problem. Specifically, in our implementation, if the grid is coarse enough so that there is only one grid cell in the x or y direction for any of the processors, the residuals and volume fractions restricted from the next finer grid are gathered and distributed globally onto every processor through an all-to-all communication procedure. Then each processor solves (or approximately solves) the same system of equations on the global coarsest grid and updates the data within its own subdomain. Two different coarsest grid system solvers are considered: direct solve by Gaussian elimination with pivoting and approximate solve by a prescribed small number (5) of box-relaxation iterations. For the test problems considered in this paper, the number of iterations needed to reach the same tolerance is very close for the two methods. The two methods have similar overall time cost up to 16 processors, while the approximate solve version is generally faster for simulations with 32 processors. This presumably reflects faster than linear increase in the cost of the direct solve as the size of the coarsest grid system grows with the number of processors. In our computational experiments reported below, the parallel simulations used the direct solve for 16 or fewer processors and the approximate solve for 32 processors.

4.4. Multigrid as preconditioner for GMRES

It is shown in [15] that multigrid used on its own as the solver for the gel system may not be very robust, especially when phase separation occurs. The reason is the appearance of a few large isolated eigenvalues of the multigrid iteration matrix. It is also shown in [15] that these large eigenvalues are well captured by a Krylov acceleration technique [28]. Specifically, the multigrid procedure just described is used as a (right) preconditioner for GMRES solution of system (15). The multigrid-preconditioned GMRES procedure is robust and efficient even when sharp gradients in volume fraction develop during the gel separation process. We follow the same strategy for the parallel algorithm by using the multigrid box-relaxation scheme as a preconditioner for the parallel GMRES solver in PETSc [24]. The Krylov method is applicable to non-symmetric matrices and non-positive definite preconditioners, both of which are found for the gel system. One multigrid sweep is carried out per Krylov iteration.

DOI: 10.1002/fld

If we let $y^{\mathrm{T}}=[(u^{\mathrm{n}})^{\mathrm{T}}\ \ (u^{\mathrm{s}})^{\mathrm{T}}\ \ p^{\mathrm{T}}]$ and f be the right-hand side of (15), the preconditioned system of equations to solve with GMRES is given by

$$\mathscr{A}^h(\mathscr{M}^h)^{-1}z=f \tag{22}$$

where $z=\mathscr{M}^h y$, and $\mathscr{M}^h$ represents the preconditioning matrix from the multigrid box-relaxation scheme. As shown in [29], the $(m+1)$th iteration of the multigrid procedure can be written in the form

$$y^{(m+1)}=(I-(\mathscr{M}^h)^{-1}\mathscr{A}^h)y^{(m)}+(\mathscr{M}^h)^{-1}f \tag{23}$$

Applying this once with zero initial guess gives

$$y^{(1)}=(\mathscr{M}^h)^{-1}f$$

This is exactly what we do when applying multigrid once as a preconditioning matrix to vector f. Therefore, the multigrid preconditioner always takes zero as the initial guess, the input vector as the right-hand side, and sets the updated state variables after one multigrid iteration to be the output vector. Function *PCShellSetApply*(*pc*, (∗*apply*)(*void*∗, *Vec*, *Vec*)) in PETSc provides a convenient way to implement a user-defined parallel preconditioner, where *pc* is the preconditioner context and *apply* is the user-provided preconditioning routine. The first input argument of function *apply* is a void pointer and can be cast to whatever the user has set the application-defined context to be and the other two arguments are global input and output vectors [24].

We consider system (15) at time t 'solved' when the residual of the ℓth iterate of the respective method satisfies

$$\frac{\|r^{(\ell)}\|_2}{\|f\|_2}\leqslant 10^{-6} \tag{24}$$

where $\widetilde{r}^{(\ell)}=f-\mathscr{A}^h y^{(\ell)}$. Unless otherwise specified, for all time steps but the initial one, we use the previous time step's values for the network and solvent velocities and pressure as initial guesses for the iterative methods. After solving system (15), updated values of the network velocity are communicated to the appropriate buffer cells. These values are needed at the beginning of the next time step in order to evaluate the discrete advection terms in approximating the continuity equation (9) for θ^{n}.

5. NUMERICAL RESULTS

To test the method in 2-D, we let the domain be the unit square and start with an initial distribution of network θ^{n} that is perturbed about the unstable region of the osmotic pressure Ψ (cf. Figure 1):

$$\theta^{\mathrm{n}}=0.08+2.5\cdot 10^{-4}(\cos(6\pi x)+\cos(4\pi y))$$

The model parameters are set to $\mu_{\mathrm{n}}=0.1$, $\lambda_{\mathrm{n}}=0.3$, $\mu_{\mathrm{s}}=0.025$, $\lambda_{\mathrm{s}}=0$, $\beta=1$, and $\kappa=10^{-7}$. The first four of these values make $\alpha_{\mathrm{n}}=0.5$ and $\alpha_{\mathrm{s}}=0.05$. Note that multigrid F-cycle is always used for cycling through the grids. Plots of the network volume fraction and velocity field at various stages of gel phase separation are shown in Figure 6. Since the pressure distribution has very

DOI: 10.1002/fld

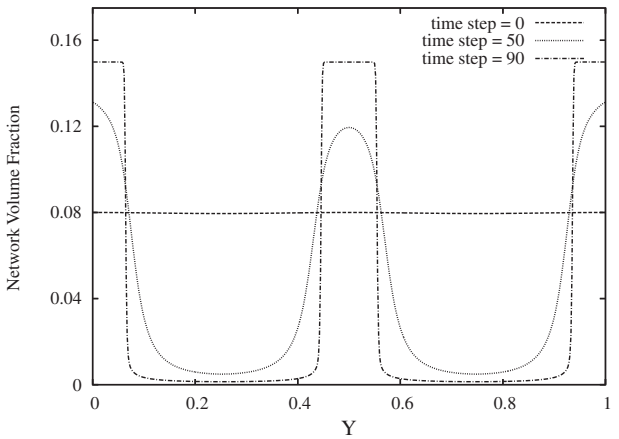


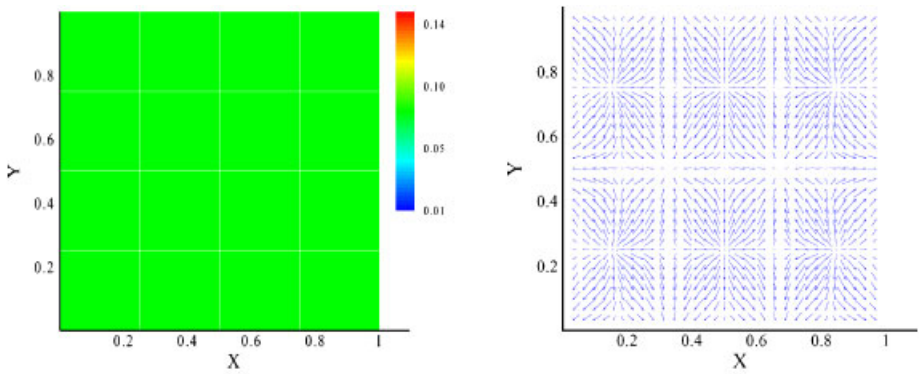
Figure 5. Network volume fraction distribution along $x = 0.5$ at different time steps.

similar patterns as the volume fraction, it is not shown here. Figure 5 shows the network volume fraction θ^{n} distributions along $x = 0.5$ for different time steps. Clearly as time evolves, the network and solvent phases separate and channels with sharp edges form. The simulation was run on 16 processors with a 512×512 grid, using multigrid-preconditioned GMRES as the solver for the momentum equations.

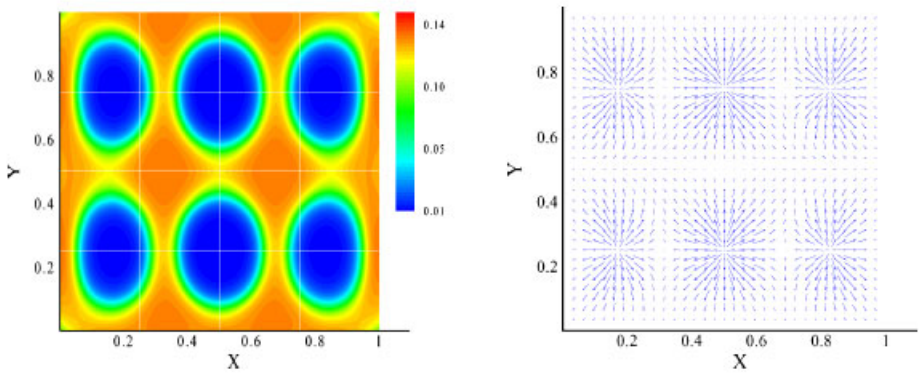
Figures 7 and 8 display the number of iterations needed to satisfy the relative residual condition (24) for multigrid (F-cycle) as a stand-alone solver (MGF) and as preconditioner for GMRES (GMRES-MGF) with mesh size 512×512 at different time steps during the simulation. The results are for serial (1×1) and parallel simulations with 32 processors (8×4 parallel decomposition). As pointed out in [18], since cells of the same color are still coupled with each other, a degradation in the smoothing rate of the scheme may be expected as the number of processors increases. Another source for the degradation may be the reduction in the number of multigrid levels as the number of processors increases, along with the approximate iterative solution of the coarsest grid equations (see Section 4.3). It is clear from the figures that this degradation, increasing the number of iterations by 1–3, is not severe with a moderate number of processors. By comparing with Figure 5, it can be seen that the number of iterations required by the numerical solvers remains approximately constant until channels with sharp edges form. Starting at this stage, the number of iterations for MGF increases sharply. In fact, MGF fails to converge in the later stages of gel separation within 30 iterations. In contrast, GMRES-MGF performs well for all stages of gel separation, showing only a moderate increase of iterations. Therefore, as previously mentioned in [15], using multigrid as the preconditioner for GMRES yields a more robust solver than multigrid alone.

By choosing a time step in which the serial and parallel methods required the same number of iterations, the speedup of the code within one time step of a simulation was determined. This is plotted in Figure 9 for grid sizes of 1024×1024 and 512×512. It is clear that for a sufficiently large problem, the code shows good scalability. Table I shows the detailed information about the wall clock time distribution percentage for parallel runnings with different number of processors. The results for the last column include parallel box relaxation, residual evaluation and

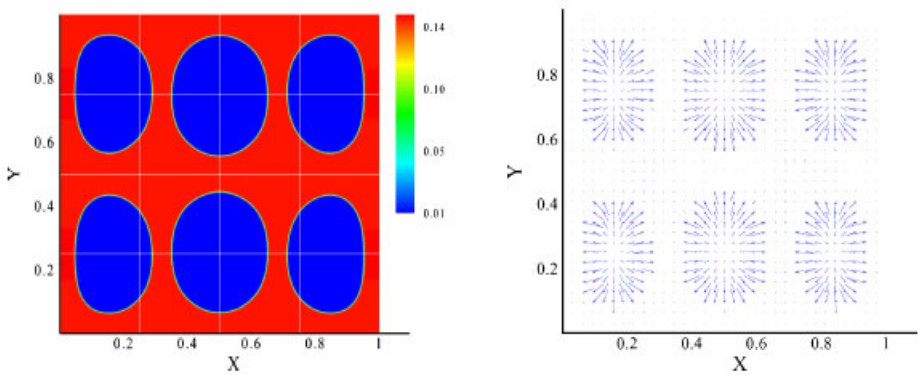
DOI: 10.1002/fld



Step = 1, Time = 39.70, Vmax = 1.51e-05



Step = 50, Time = 153.87, Vmax = 3.69e-3



Step = 110, Time = 401.59, Vmax = 1.76e-4

Figure 6. Network volume fraction (left) and distribution of the network velocity (right). $V_{\max}$ is the largest velocity value in the plot.

DOI: 10.1002/fld

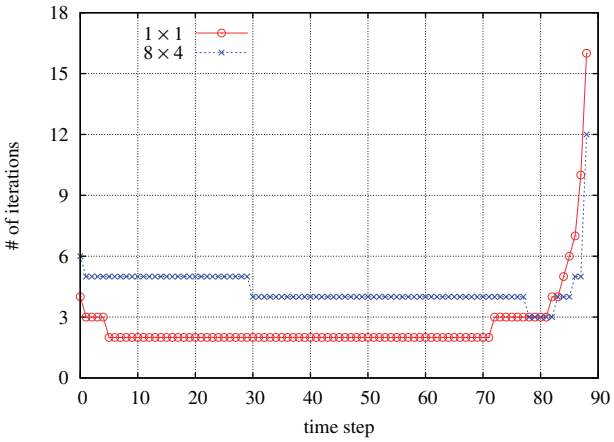


Figure 7. Number of iterations for MGF at different time steps for 1 and 32 processors.

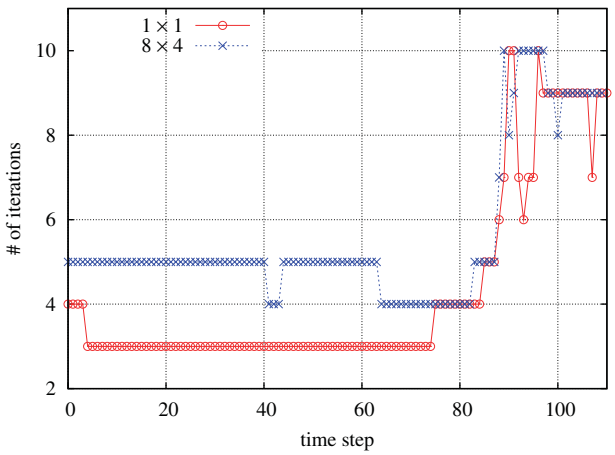


Figure 8. Number of iterations for GMRES-MGF at different time steps for 1 and 32 processors.

intergrid transfers, which are fully parallelizable. Smoothing on coarsest grid only makes use of one processor and forms the serial part of the algorithm.

6. CONCLUSIONS

We have presented a computational methodology for simulating models of two-phase gel dynamics. The main computational challenge of these models is in solving the momentum and incompressibility equations that involve variable-coefficient differential terms and terms coupling the two fluids. Our method of solving this system by using parallel multigrid with a box-type relaxation procedure as a preconditioner for GMRES solver in PETSc package appears to be very

DOI: 10.1002/fld

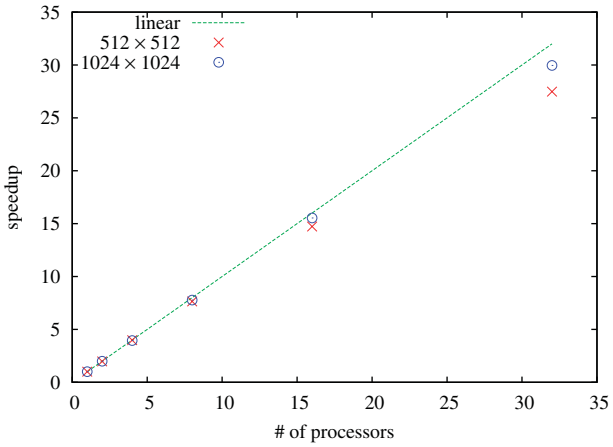


Figure 9. Speedup plot for GMRES-MGF for different sized grids.

Table I. Wall clock time distributions for simulations on 1024×1024 grid.

No. of processors	Communication (%)	Smoothing on coarsest grid (%)	Others (%)
8	2.7	0.8	96.5
16	6.8	2.3	90.9
32	11.1	5.1	83.8

effective with good parallel performance. Numerical results from model problems in two dimensions indicate that the method is both robust and efficient.

ACKNOWLEDGEMENTS

This work was supported in part by NSF grant DMS-0540779. We gratefully acknowledge a grant of computing time from the Center for High Performance Computing at the University of Utah.

REFERENCES

1. Alt W, Dembo M. Cytoplasm dynamics and cell motion: two-phase flow models. *Mathematical Biosciences* 1999; **156**:207–228.
2. Cogan NG, Keener JP. Channel formation in gels. *SIAM Journal on Applied Mathematics* 2005; **65**:1839–1854.
3. Dembo M. Mechanics and control of the cytoskeleton in Amoeba proteus. *Biophysical Journal* 1989; **55**(6): 1053–1080.
4. Dembo M, Harlow F. Cell motion, contractile networks, and the physics of interpenetrating reactive flow. *Biophysical Journal* 1986; **50**(1):109–121.
5. Doi M, Onuki A. Dynamic coupling between stress and composition in polymer solutions and blends. *Journal de Physique II France* 1992; **2**:1631–1656.
6. He XY, Dembo M. On the mechanics of the first cleavage division of the sea urchin egg. *Experimental Cell Research* 1997; **233**(2):252–273.
7. Tanaka H. Viscoelastic phase separation. *Journal of Physics*: *Condensed Matter* 2000; **12**:R207–R264.
8. Drew DA. Mathematical modeling of two-phase flow. *Annual Review of Fluid Mechanics* 1983; **15**:261–291.

DOI: 10.1002/fld

9. Drew DA, Passman SL. Theory of multicomponent fluids. In *Applied Mathematical Sciences*, Marsden JE, Sirovich L (eds). Springer: New York, 1999.
10. Lemon G, King JR, Byrne HM, Jensen OE, Shakesheff KM. Mathematical modelling of engineered tissue growth using a multiphase porous flow mixture theory. *Journal of Mathematical Biology* 2006; **52**(5):571–594.
11. Byrne H, Preziosi L. Modelling solid tumour growth using the theory of mixtures. *Mathematical Medicine and Biology* 2003; **20**(4):341–366.
12. Lubkin SR, Jackson T. Multiphase mechanics of capsule formation in tumors. *Journal of Biomechanical Engineering* 2002; **124**(2):237–243.
13. Alpkvist E, Klapper I. A multidimensional multispecies continuum model for modelling biofilm development. *Bulletin of Mathematical Biology* 2007; **69**:765–789.
14. Cogan NG, Keener JP. The role of the biofilm matrix in structural development. *Mathematical Medicine and Biology* 2004; **21**:147–166.
15. Wright GB, Guy RD, Fogelson AL. An efficient and robust method for simulating two-phase gel dynamics. *SIAM Journal on Scientific Computing* 2008; **30**(5):2535–2565. DOI: 10.1137/070695927.
16. Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surfaces. *Physics of Fluids* 1965; **8**:2182–2189.
17. Vanka SP. Block-implicit multigrid solution of Navier–Stokes equations in primitive variables. *Journal of Computational Physics* 1986; **65**:138–158.
18. Degani AT, Fox GC. Parallel multigrid computation of the unsteady incompressible Navier–Stokes equations. *Journal of Computational Physics* 1996; **128**(1):223–236.
19. Manservisi S. Numerical analysis of Vanka-type solvers for steady Stokes and Navier–Stokes flows. *SIAM Journal on Scientific Computing* 2006; **44**:2025–2056.
20. Sivaloganathan S. The use of local mode analysis in the design and comparison of multigrid methods. *Computer Physics Communications* 1991; **65**:246–252.
21. Thompson MC, Ferziger JH. An adaptive multigrid technique for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 1989; **82**:94–121.
22. Trottenberg U, Oosterlee CW, Schüller A. *Multigrid*. Academic Press: London, 2000.
23. Wesseling P, Oosterlee CW. Geometric multigrid with applications to computational fluid dynamics. *Journal of Computational and Applied Mathematics* 2001; **128**:311–334.
24. Balay S, Gropp W, McInnes L, Smith B. *PETSc Users Manual. ANL-95/11*. Argonne National Laboratory, 2001.
25. Allen RD, Cowden RR. Syneresis in ameboid movement: its localization by interference microscopy and its significance. *The Journal of Cell Biology* 1962; **12**:185–189.
26. Benzi M, Golub GH, Liesen J. Numerical solution of saddle point problems. *Acta Numerica* 2005; **14**:1–137.
27. Leveque RJ. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis* 1996; **33**:627–665.
28. Oosterlee CW, Washio T. An evaluation of parallel multigrid as a solver and a preconditioner for singularly perturbed problems. *SIAM Journal on Scientific Computing* 1998; **19**:87–110.
29. Saad Y. *Iterative Methods for Sparse Linear Systems*. SIAM: Philadelphia, PA, 2003.