

UCRL-JRNL-225206



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

A Hybrid Fluid-Particle Method to Simulate Flow and Transport of DNA in Microdevices

D. Trebotich, G. H. Miller, M. D. Bybee

October 11, 2006

Journal of Nanoscience and Nanotechnology

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

A Penalty Method to Model Particle Interactions in DNA-laden Flows

D. Trebotich *

Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-560, Livermore, CA 94551, USA

G. H. Miller

*Department of Applied Science, University of California, One Shields Avenue, Davis, CA 95616, USA; and
Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA*

M. D. Bybee

Department of Chemical and Biomolecular Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Abstract

We present a hybrid fluid-particle algorithm to simulate flow and transport of DNA-laden fluids in microdevices. Relevant length scales in microfluidic systems range from characteristic channel sizes of millimeters to micron scale geometric variation (e.g., post arrays) to 10 nanometers for the length of a single rod in a bead-rod polymer representation of a biological material such as DNA. The method is based on a previous fluid-particle algorithm in which long molecules are represented as a chain of connected rods, but in which the physically unrealistic behavior of rod crossing occurred. We have extended this algorithm to include screened Coulombic forces between particles by implementing a Debye-Hückel potential acting between rods. In the method an unsteady incompressible Newtonian fluid is discretized with a second-order finite difference method in the interior of the Cartesian grid domain; an embedded boundary volume-of-fluid formulation is used near boundaries. The bead-rod polymer model is fully coupled to the solvent through body forces representing hydrodynamic drag and stochastic thermal fluctuations. While intra-polymer interactions are modeled by a soft potential, polymer-structure interactions are treated as perfectly elastic collisions. We demonstrate this method on flow and transport of a polymer through a post array microchannel in 2D where the polymer incorporates more realistic physical parameters of DNA, and compare to previous simulations where rods are allowed to cross. We also show that the method is capable of simulating 3D flow in a packed bed micro-column.

1 Introduction

Microfluidics is emerging as a vital technology in the design of devices used for fluidic control in biodefense and biomedical applications including pathogen detection, continuous monitoring, and drug delivery. Numerical algorithms that can model flows of complex biological fluids within these devices are needed to facilitate development and optimization. However, the fluid dynamics in these systems is not well understood due to the presence of large particles in the bulk fluid whose sizes are comparable to a length scale in the flow geometry. This is inherently a multiscale problem since continuum flow is strongly controlled by channel geometry and, to a lesser degree, by solvated polymers whose dynamics are sensitive to nanometer-scale properties of the system. For example, in a biomolecular processing microdevice the length scales range from about 1-10 nanometers (the scale at which the surface of a small globular protein is discretized, or the length of a rod in a polymer model for single stranded DNA), to tens or hundreds of microns for typical fluidic processors (average channel width) with the full length of channels extending to a millimeter. The time scales range from tens of nanoseconds (a characteristic time for protein reorientation and short-range interactions of DNA) to a few minutes (a typical time for a complete bioassay). Currently it is not possible to computationally resolve the phenomena of interest over this entire range of length and time scales with any single computational method, necessitating new multiscale schemes.

The overall goal of this paper is to predict the flow, transport, and conformation of individual DNA molecules in microfluidic or biological systems. A highly concentrated or even semi-dilute solution of suspended polymer molecules may be represented at the larger, engineering scales of a microdevice with a continuum constitutive model of a viscoelastic fluid. We have developed such models where DNA in a solvent is represented by the Oldroyd-B constitutive equation, which assumes that the effect of DNA on the fluid is an elastic one, and fundamentally a spring and dashpot fluid element [22]. However, when the geometry length scales are comparable to the inter-polymer spacing, a continuum approximation is no longer appropriate; instead, a hybrid approach with a discrete molecular approximation is needed. Furthermore, if the fluid is not static then there must be full coupling of particles and fluid

* Corresponding author.

Email addresses: trebotich1@llnl.gov (D. Trebotich),
grgmiller@ucdavis.edu (G. H. Miller), bybee@uiuc.edu (M. D. Bybee).

if a model is to be predictive. We have also developed a hybrid fluid-particle method along these lines where a bead-rod polymer representation of DNA is coupled to an incompressible viscous solvent [26]. In this method the polymer and solvent forces – both viscous and stochastic – are fully-coupled such that the fluid can “feel” the effect of the polymer, which is different from, for example, [11], where the coupling is one way and the flow is steady. Additionally in [26], the polymer nodes may experience elastic collisions with domain boundaries; however, there is no model for intra-polymer interactions in that work, that is, a mechanism to prevent rods from crossing.

Our specific interest in this paper is to incorporate more physically realistic behavior of DNA into the bead-rod polymer model as part of a fully-coupled fluid-particle algorithm. For polymer models of DNA, of immediate concern is the non-crossing constraint: a polymer section cannot pass through another polymer section. This constraint is frequently neglected (e.g., [21]). Padding and Briels describe a complex algorithm to detect and prevent bond crossings [19]. Bonds are considered as elastic bands between bonded particles. When any two of these elastic bands make contact, an entanglement point is created which prevents them from crossing. We have since improved the particle algorithm in [26] to include ideas in [19] but with some improvements in order to enhance the fluid coupling. In a novel technique we developed a rigid constraint whereby infinitely thin rods elastically bounce off one another, described initially in [4], then in 2D [25] and in 3D with a new mesoscale model [17]. We have also used a short-ranged Debye-Hückel potential acting between rods based on the algorithm for a repulsive Lennard-Jones potential in [15]. The short-ranged Debye-Hückel potential has also been implemented for rod-surface interactions. The purpose of this paper is two-fold: to extend our previous fluid-particle coupling [26] to include the non-crossing constraint for intra-polymer interactions using the Debye-Hückel potential; and to expand on the fluid modeling with embedded boundary methods for complex microscale geometries. Polymer-surface interactions are elastic as in [26]. We demonstrate the capability on non-trivial irregular microscale flow geometries in 2D and discuss extension to more complicated 3D applications.

2 Equations of Motion

We use the Navier-Stokes equations to model the solvent as a continuum on domain Ω :

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{\rho} \nabla P = \nu \Delta \mathbf{u} + \frac{1}{\rho} \mathbf{F} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0. \quad (2)$$

These equations describe an incompressible fluid of density ρ , pressure P , velocity \mathbf{u} , and Newtonian viscosity ν , subject to an additional body force \mathbf{F} . On the domain boundary $\delta\Omega$ we have the no-slip boundary condition $\mathbf{u} = 0$.

The polymer solute is represented as a collection of point masses each subject to Newton's second law of motion

$$m_\alpha \frac{d^2 \mathbf{x}_\alpha}{dt^2} = m_\alpha \frac{d\mathbf{v}_\alpha}{dt} = \mathbf{f}_\alpha. \quad (3)$$

Here m_α is the mass of the α^{th} particle, \mathbf{x}_α is its coordinate, and \mathbf{v}_α is its velocity. The particle is subject to a force \mathbf{f}_α which combines a Stokes drag term with a stochastic (Brownian) perturbation,

$$\mathbf{f}_\alpha = m_\alpha \gamma (\mathbf{u}(\mathbf{x}_\alpha) - \mathbf{v}_\alpha) + \mathcal{F}_{\mathbf{B}\alpha}. \quad (4)$$

Here, $1/\gamma$ is a phenomenological relaxation time ($m\gamma = 6\pi\mu b$ for a Stokes sphere of radius b), and $\mathcal{F}_{\mathbf{B}}$ is the stochastic force

$$\langle \mathcal{F}_{\mathbf{B}\alpha}(t) \rangle = 0 \quad (5)$$

$$\langle \mathcal{F}_{\mathbf{B}\alpha}(t) \mathcal{F}_{\mathbf{B}\alpha}(t') \rangle = \sigma_\alpha^2 I \delta(t - t') \quad (6)$$

$$\sigma_\alpha = \sqrt{2m_\alpha \gamma k_B T}, \quad (7)$$

with k_B being Boltzmann's constant and T the temperature.

The force \mathbf{F} acting on the fluid is

$$\mathbf{F}(\mathbf{x}) = - \sum_\alpha \mathbf{f}_\alpha \delta_\epsilon(\mathbf{x} - \mathbf{x}_\alpha) \quad (8)$$

where δ_ϵ represents a smoothed Dirac delta function with length scale ϵ .

In addition to the incompressibility condition (2) we have three additional constraints on the particles: (i) interparticle spacing is constant

$$\|\mathbf{x}_\alpha - \mathbf{x}_\beta\| = a \quad (9)$$

if particles α and β represent adjacent nodes in a bead-rod polymer representation; (ii) particles cannot pass through a physical boundary,

$$\mathbf{x}_\alpha \in \Omega; \quad (10)$$

and (iii) rods cannot cross.

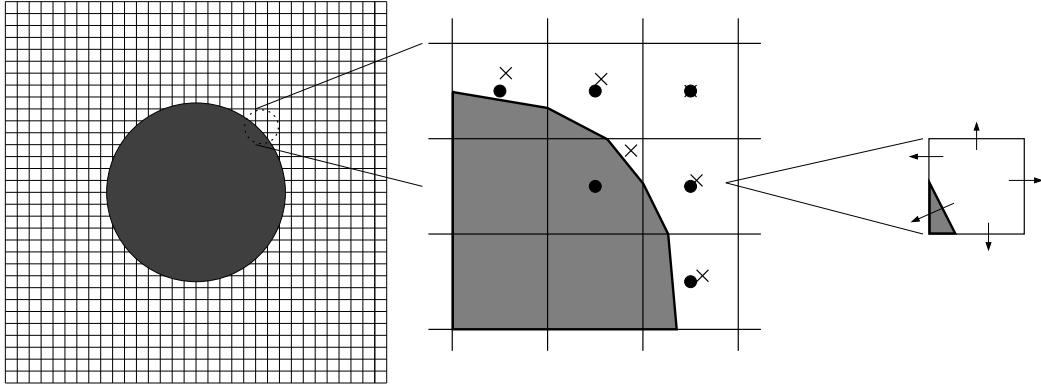


Fig. 1. Example of an irregular geometry on a Cartesian grid (left). Close-up view of embedded boundaries “cutting” regular cells (middle). Single irregular cut cell showing boundary fluxes (right). Shaded area represents volume of cells excluded from domain. Dots represent cell-centers. X’s represent centroids.

3 Embedded Boundary Methods for Incompressible Viscous Flow

We use a Cartesian grid embedded boundary method to discretize the fluid equations in the presence of irregular boundaries. In this approach, the irregular domain is discretized as a collection of control volumes formed by the intersection of the problem domain with the cubic Cartesian grid cells as in a “cookie-cutter” (see Figure 1). The various operators – the discrete divergence $\nabla \cdot$, discrete gradient ∇ , and discrete Laplacian Δ – are approximated using finite volume differences on the irregular control volumes, with the fluxes computed using the primary discretized dependent variables, which approximate the solution evaluated at the centers of the original Cartesian cells.

For example, the Laplacian operator, $\Delta\phi = \nabla \cdot \nabla\phi$, is a divergence of a flux and can be calculated in a finite volume (such as the cut cell in Figure 1b) by applying the divergence theorem which converts a volume integral to a surface integral so that fluxes can be simply summed around the perimeter of the cut cell along normals:

$$\nabla \cdot \vec{F} \approx \frac{1}{\kappa h^d} \int_{\Omega} \nabla \cdot \vec{F} d\Omega = \frac{1}{\kappa h} \int_{\partial\Omega} \vec{F} \cdot \vec{n} dS = \frac{1}{\kappa h} \sum_s \alpha_s \vec{F}_s + \alpha_b \vec{F}_b \quad (11)$$

where κ is the volume fraction of the cell, α is the area fraction of a cell edge, h is the grid spacing and subscripts s and b indicate cell edges and the embedded boundary, respectively.

To obtain a flux at an embedded boundary when only cell-centered data exists, which is the case when a homogeneous Dirichlet boundary condition is needed for no-slip of the fluid velocity at solid walls, a special extrapolation procedure is needed. Instead of applying the advanced stencil for Dirichlet boundary

conditions described in [12] we assume a lower-order truncation error stencil based on least squares [23,20], but one that maintains second-order solution error, to interpolate the flux φ at an irregular boundary, B . The gradient $\nabla\varphi$ is obtained from the system:

$$\mathcal{A} \cdot \nabla\varphi = \delta\varphi \tag{12}$$

where

$$\mathcal{A} = (\delta\vec{x}_1, \delta\vec{x}_2, \dots, \delta\vec{x}_p)^T \tag{13}$$

$$\delta\varphi = (\delta\varphi_1, \delta\varphi_2, \dots, \delta\varphi_p)^T \tag{14}$$

$$\delta\vec{x}_m = \vec{x}_m - \vec{x}_B \tag{15}$$

$$\delta\varphi_m = \varphi_m - \varphi_B. \tag{16}$$

The stencil ($m = 1, 2, \dots, p$) is determined by the normal of the embedded boundary. In 2D the stencil includes up, side and corner cells (see Figure 1b), with $p = 3$; in 3D, the normal points to an octant where $p = 7$. In 2D, for example, there are two equations and three unknowns; a least squares fit is applied to obtain the gradients $\nabla\varphi$.

The embedded boundary / volume-of-fluid approach has been used as the basis for second-order accurate methods for elliptic, parabolic, and hyperbolic PDEs in two and three dimensions [12,16,8,20]. These methods also have been combined using the predictor-corrector approach in [2] to provide a second-order accurate projection method for the incompressible Navier-Stokes equations for problems in irregular geometries [1]. This is the underlying algorithm for solving the fluid equations (1,2). In addition, the embedded boundary volume-of-fluid approach to complex geometry is compatible with a fast and accurate surface extraction technique to go from raw image data to direct simulation without loss in geometric detail [9,24].

4 Hybrid Fluid-Particle Algorithm with Short Range Forces

The particle solver is tightly coupled to the fluid solver through an interwoven predictor-corrector strategy as in [26]. In the previous algorithm [26] the particle position is advanced and then the rod length constraint is applied in both particle predictor and corrector steps. Here, the rod-rod crossing constraints are added in both the predictor and corrector with the bead-surface constraint applied last in the hierarchy.

We discretize time in steps Δt , with $t^n = t^{n-1} + \Delta t$; and we discretize space with a rectangular Cartesian grid, $\mathbf{x}_{i,j,k} = h(i, j, k)$, regardless of the geometry

of the fluid domain Ω . The domain boundary $\delta\Omega$ is given indirectly by assigning to each rectangular grid cell a set of volume and area fractions, which describe the intersection of the cell with the fluid boundary. Away from boundaries, the discrete divergence $\nabla\cdot$, discrete gradient ∇ , and discrete Laplacian Δ operators use standard symmetric second-order discretizations of the fluid. These operators are modified by the presence of boundaries as described above.

Following [26], to advance the coupled fluid-particle system in time consists of the following four steps, in sequence. A tilde is used to denote quantities computed in the predictor step of our predictor-corrector strategy; no tilde is used for the corrector. Superscripts $*$ and \dagger denote provisional quantities; e.g., \mathbf{u}^* is a fluid velocity subject to divergence cleaning, and \mathbf{x}^* , \mathbf{v}^* and \mathbf{x}^\dagger , \mathbf{v}^\dagger are particle coordinates and velocities subject to correction by appropriate constraints. All quantities are assumed to be cell-centered unless otherwise noted, either explicitly or by use of cell edge indices (e.g., $i + \frac{1}{2}$).

Step 1: Particle Predictor

We base our solution to the particle equations on $\mathcal{O}(\Delta t^{2.5})$ -accurate Ito-Taylor [14] expansions of the Langevin equations for variables \mathbf{x}_α and $\mathbf{e}^{\gamma t}\mathbf{v}_\alpha$.

The predictor is derived using time- n quantities only to estimate the time- $(n+1)$ state of the particles:

$$\tilde{\mathbf{v}}_\alpha^{*,n+1} = \mathbf{u}^n(\mathbf{x}_\alpha^n) + (\mathbf{v}_\alpha^n - \mathbf{u}^n(\mathbf{x}_\alpha^n))\mathbf{e}^{-\gamma\Delta t} + \frac{\sigma}{m_\alpha}\mathbf{R}_{v,\alpha}^n(\Delta t) \quad (17)$$

$$\tilde{\mathbf{x}}_\alpha^{*,n+1} = \mathbf{x}_\alpha^n + (\mathbf{v}_\alpha^n - \mathbf{u}^n(\mathbf{x}_\alpha^n))\frac{1 - \mathbf{e}^{-\gamma\Delta t}}{\gamma} + \mathbf{u}^n(\mathbf{x}_\alpha^n)\Delta t + \frac{\sigma}{\gamma m_\alpha}\mathbf{R}_{x,\alpha}^n(\Delta t) \quad (18)$$

$$\Delta t \mathbf{f}_\alpha^n = m_\alpha (\tilde{\mathbf{v}}_\alpha^{*,n+1} - \mathbf{v}_\alpha^n) \quad (19)$$

$$\mathbf{F}^n(\mathbf{x}) = -\sum_\alpha \mathbf{f}_\alpha^n \delta_\epsilon(\mathbf{x} - \mathbf{x}_\alpha^n), \quad (20)$$

with $\mathbf{R}_{v,\alpha}^n(\Delta t)$ and $\mathbf{R}_{x,\alpha}^n(\Delta t)$ vectors of independent random numbers drawn from Gaussian distributions with zero mean and variances

$$\langle R_v(\Delta t)^2 \rangle = \frac{1}{2\gamma} (1 - \mathbf{e}^{-2\gamma\Delta t}) \quad (21)$$

$$\langle R_x(\Delta t)^2 \rangle = \frac{1}{2\gamma} [2\gamma\Delta t - \mathbf{e}^{-2\gamma\Delta t} + 4\mathbf{e}^{-\gamma\Delta t} - 3] \quad (22)$$

$$\langle R_x(\Delta t)R_v(\Delta t) \rangle = \frac{1}{2\gamma} (1 - \mathbf{e}^{-\gamma\Delta t})^2 \quad (23)$$

for variables of identical vector index, and all covariances are zero for terms with different vector indices. In Eq. (17) $\mathbf{u}^n(\mathbf{x}_\alpha^n)$ is evaluated by linear interpo-

lation of the cell-centered discretization \mathbf{u}_{cc}^n . The discrete Dirac delta function is represented using a PIC or cloud-in-cell model [3].

Rod length constraint. The particle coordinates $\tilde{\mathbf{x}}^{*,n+1}$ do not in general obey the rod length constraint (9). To enforce this condition we use the Lagrange multiplier technique described in [6]. This correction consists of iterative solution of a tridiagonal linear system obtained by linearization of (9). We refer to this corrected state as $\tilde{\mathbf{x}}^\dagger$, and

$$\tilde{\mathbf{v}}_\alpha^\dagger = \tilde{\mathbf{v}}_\alpha^{*,n+1} + \frac{1}{\Delta t}(\tilde{\mathbf{x}}_\alpha^\dagger - \tilde{\mathbf{x}}_\alpha^{*,n+1}) \quad (24)$$

is the corresponding corrected velocity. See also [17] for additional details.

Rod-rod repulsion. The time-linear trajectory $\mathbf{x}_\alpha^n \rightarrow \tilde{\mathbf{x}}_\alpha^\dagger$ may cause rods to cross, thereby violating the rod crossing constraint. This is prevented by implementing a short range potential between rods. The algorithm for a repulsive potential interaction between rods is taken from the work of Kumar and Larson [15]. The shortest vector between two rods is calculated, and a repulsive force is applied to both rods along that vector. While Kumar and Larson explored the use of an exponential potential and a Lennard-Jones potential, this work uses a short-ranged Debye-Hückel potential

$$U^{\text{DH}} = A \frac{e^{-\kappa r}}{r}.$$

It should be noted that while the mathematically correct introduction of these forces is directly into the Langevin equation, we initially introduced these forces after the rod length constraint to avoid complexities in the Ito-Taylor expansion of [26]. For details of the hard constraint for rod crossing see [25] in 2D and [17] in 3D.

Bead-surface boundary. Application of the repulsive potential may lead a particle α across the fluid domain boundary, thereby violating constraint (10). We use a continuous distance function representation of the domain boundary to detect collisions due to such crossings. If the trajectory contacts the domain at a point $\chi \in \delta\Omega$ at relative time τ , $0 < \tau \leq \Delta t$, we elastically “bounce” the particle off the boundary at χ as follows. Let \mathbf{n} be unit normal to the boundary at χ :

$$\tilde{\mathbf{v}}^{n+1} = \tilde{\mathbf{v}}^\ddagger - 2(\mathbf{n} \cdot \tilde{\mathbf{v}}^\ddagger)\mathbf{n} \quad (25)$$

$$\tilde{\mathbf{x}}_\alpha^{n+1} = \chi + (\Delta t - \tau)\tilde{\mathbf{v}}^{n+1}. \quad (26)$$

If no collision is indicated, then $\tilde{\mathbf{v}}_\alpha^{n+1} = \tilde{\mathbf{v}}_\alpha^\ddagger$, etc. We have also implemented a short range repulsive potential in addition to the elastic collision model to

prevent particles from leaving the fluid domain. The same algorithm as in rod-rod repulsion using the shortest vector between a rod and the level set boundary defining a surface can be applied here, too.

Step 2: Fluid Predictor

The main feature of the fluid predictor is a numerical estimate of the convective derivative using a second-order Godunov method [7]. (For details of the embedded boundary discretization for this advection step see [8] and [1].) We calculate $(\mathbf{u} \cdot \nabla)\mathbf{u}^{n+\frac{1}{2}}$ in several steps, first, by approximating edge- and time-centered velocities, $\hat{\mathbf{u}}^{n+\frac{1}{2}}$, using an upwind Taylor series expansion of \mathbf{u}^n and the PDE (1), including a viscous stability correction to the slopes [18], an explicit determination of the viscous source term, and the explicit source \mathbf{F}^n , but omitting the pressure. For example, the extrapolated states at plus and minus edges from a cell center in 2D are

$$\hat{\mathbf{u}}^{x,+} = \mathbf{u}^n + \frac{1}{2} \min \left[\left(1 - u^n \frac{\Delta t}{\Delta x} \right), 1 \right] \Delta_x \mathbf{u}^n - \frac{\Delta t}{2\Delta y} v^n \Delta_y \mathbf{u}^n - \frac{\nu \Delta t}{2} \Delta \mathbf{u}^n \quad (27)$$

$$\hat{\mathbf{u}}^{x,-} = \mathbf{u}^n - \frac{1}{2} \min \left[\left(1 + u^n \frac{\Delta t}{\Delta x} \right), 1 \right] \Delta_x \mathbf{u}^n + \frac{\Delta t}{2\Delta y} v^n \Delta_y \mathbf{u}^n + \frac{\nu \Delta t}{2} \Delta \mathbf{u}^n \quad (28)$$

$$\hat{\mathbf{u}}^{y,+} = \mathbf{u}^n + \frac{1}{2} \min \left[\left(1 - v^n \frac{\Delta t}{\Delta y} \right), 1 \right] \Delta_y \mathbf{u}^n - \frac{\Delta t}{2\Delta x} u^n \Delta_x \mathbf{u}^n - \frac{\nu \Delta t}{2} \Delta \mathbf{u}^n \quad (29)$$

$$\hat{\mathbf{u}}^{y,-} = \mathbf{u}^n - \frac{1}{2} \min \left[\left(1 + v^n \frac{\Delta t}{\Delta y} \right), 1 \right] \Delta_y \mathbf{u}^n + \frac{\Delta t}{2\Delta x} u^n \Delta_x \mathbf{u}^n + \frac{\nu \Delta t}{2} \Delta \mathbf{u}^n, \quad (30)$$

where monotonized 2nd-order slopes are used

$$(\Delta_x \mathbf{u})^n = \begin{cases} (\Delta_x \mathbf{u})^{vL} & \text{if } (\mathbf{u}_{i+1,j}^n - \mathbf{u}_{i,j}^n)(\mathbf{u}_{i,j}^n - \mathbf{u}_{i-1,j}^n) > 0 \\ 0 & \text{if } (\mathbf{u}_{i+1,j}^n - \mathbf{u}_{i,j}^n)(\mathbf{u}_{i,j}^n - \mathbf{u}_{i-1,j}^n) \leq 0, \end{cases} \quad (31)$$

and

$$\begin{aligned} (\Delta_x \mathbf{u})^{vL} &= \text{sign} \left(\frac{\mathbf{u}_{i+1,j}^n - \mathbf{u}_{i-1,j}^n}{2} \right) \\ &\quad \times \min(2|\mathbf{u}_{i,j}^n - \mathbf{u}_{i-1,j}^n|, 2|\mathbf{u}_{i+1,j}^n - \mathbf{u}_{i,j}^n|, .5|\mathbf{u}_{i+1,j}^n - \mathbf{u}_{i-1,j}^n|) \end{aligned} \quad (32)$$

after [27], with one-sided differences at boundaries; the upwinded transverse slopes include a viscous correction for stability [18]

$$(\Delta_y \mathbf{u})_{i,j}^n = \begin{cases} \mathbf{u}_{i,j+1}^n - \mathbf{u}_{i,j}^n + \frac{\nu \Delta t}{2} (\Delta \mathbf{u}_{i,j+1}^n - \Delta \mathbf{u}_{i,j}^n) & \text{if } v_{i,j}^n < 0 \\ \mathbf{u}_{i,j}^n - \mathbf{u}_{i,j-1}^n + \frac{\nu \Delta t}{2} (\Delta \mathbf{u}_{i,j}^n - \Delta \mathbf{u}_{i,j-1}^n) & \text{if } v_{i,j}^n \geq 0 \end{cases} \quad (33)$$

$$(\Delta_x \mathbf{u})_{i,j}^n = \begin{cases} \mathbf{u}_{i+1,j}^n - \mathbf{u}_{i,j}^n + \frac{\nu \Delta t}{2} (\Delta \mathbf{u}_{i+1,j}^n - \Delta \mathbf{u}_{i,j}^n) & \text{if } u_{i,j}^n < 0 \\ \mathbf{u}_{i,j}^n - \mathbf{u}_{i-1,j}^n + \frac{\nu \Delta t}{2} (\Delta \mathbf{u}_{i,j}^n - \Delta \mathbf{u}_{i-1,j}^n) & \text{if } u_{i,j}^n \geq 0. \end{cases} \quad (34)$$

We then solve a Riemann problem at each cell edge. For the incompressible Navier Stokes equations the exact Riemann problem solution amounts to simple upwinding:

$$\hat{\mathbf{u}}^{n+\frac{1}{2}} = \begin{cases} \hat{\mathbf{u}}_{i,j}^{x,+} & \text{if } \frac{u_{i,j}^n + u_{i+1,j}^n}{2} > 0 \\ \hat{\mathbf{u}}_{i+1,j}^{x,-} & \text{if } \frac{u_{i,j}^n + u_{i+1,j}^n}{2} < 0 \\ \frac{\hat{\mathbf{u}}_{i,j}^{x,+} + \hat{\mathbf{u}}_{i+1,j}^{x,-}}{2} & \text{if } \frac{u_{i,j}^n + u_{i+1,j}^n}{2} = 0. \end{cases} \quad (35)$$

These provisional edge states are made divergence-free (i.e., account for previously omitted pressure in the Taylor extrapolation) with a MAC-stencil Hodge projection,

$$\mathbf{u}^{n+\frac{1}{2}} = (I - \nabla \Delta^{-1} \nabla \cdot) \hat{\mathbf{u}}^{n+\frac{1}{2}}. \quad (36)$$

The edge states $\mathbf{u}^{n+\frac{1}{2}}$ are used to estimate the convective derivative $(\mathbf{u} \cdot \nabla) \mathbf{u}^{n+\frac{1}{2}}$:

$$\begin{aligned} [(\mathbf{u} \cdot \nabla) \mathbf{u}]^{n+\frac{1}{2}} &= \frac{(u_{i+\frac{1}{2},j}^{n+\frac{1}{2}} + u_{i-\frac{1}{2},j}^{n+\frac{1}{2}})}{2} \frac{(\mathbf{u}_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - \mathbf{u}_{i-\frac{1}{2},j}^{n+\frac{1}{2}})}{\Delta x} \\ &\quad + \frac{(v_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} + v_{i,j-\frac{1}{2}}^{n+\frac{1}{2}})}{2} \frac{(\mathbf{u}_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - \mathbf{u}_{i,j-\frac{1}{2}}^{n+\frac{1}{2}})}{\Delta y}. \end{aligned} \quad (37)$$

Then,

$$\frac{\tilde{\mathbf{u}}^{n+1} - \mathbf{u}^n}{\Delta t} = -\frac{1}{\rho} (\nabla \pi^{n-\frac{1}{2}}) - [(\mathbf{u} \cdot \nabla) \mathbf{u}]^{n+\frac{1}{2}} + \frac{1}{\rho} F^n + \nu \Delta \tilde{\mathbf{u}}^{n+1} \quad (38)$$

is solved implicitly for the time- $(n+1)$ cell-centered velocity field $\tilde{\mathbf{u}}^{n+1}$ to yield an estimate to the fluid velocity at time- $n+1$. This velocity is not necessarily divergence-free because $\nabla \pi^{n-\frac{1}{2}}$ is lagged in time (see (48)). Second-order accuracy in time can be obtained here using the approach described in [13,16,1] instead of backward Euler. The time step in this explicit fluid predictor is subject to the CFL condition ($\sigma < 1$)

$$\max_{i,j} [|u_{i,j}|] \Delta t < \sigma \Delta x \quad (39)$$

$$\max_{i,j} [|v_{i,j}|] \Delta t < \sigma \Delta y. \quad (40)$$

In fact, due to the higher frequency particle motions, σ has been reduced by an order of magnitude for particle stability. A stable particle time step that is equivalent to the fluid CFL condition can be achieved using the ideas in [17], providing optimal stability to a multiscale fluid-particle system.

Step 3: Particle Corrector

The particle update is re-evaluated using a mean fluid velocity $\bar{\mathbf{u}}$,

$$\bar{\mathbf{u}}_\alpha = \frac{\mathbf{u}^n(\mathbf{x}_\alpha^n) + \tilde{\mathbf{u}}^{n+1}(\tilde{\mathbf{x}}_\alpha^{n+1})}{2}, \quad (41)$$

if particle α was not predicted to have bounced off the interface; or,

$$\bar{\mathbf{u}}_\alpha = \frac{\tau}{2\Delta t} \mathbf{u}^n(\mathbf{x}_\alpha^n) + \frac{(\Delta t - \tau)}{2\Delta t} (I - 2\mathbf{nn}^T) \tilde{\mathbf{u}}^{n+1}(\tilde{\mathbf{x}}_\alpha^{n+1}), \quad (42)$$

if it was predicted to have bounced. Eq. (42) is the average field \mathbf{u} on the particle's trajectory, referenced to the particle's original t^n orientation. This expression takes into account the $\mathbf{u} = 0$ no slip boundary condition experienced at relative time τ . For the particle trajectory we then have the $\mathcal{O}(\Delta t^{2.5})$ estimate

$$\mathbf{v}_\alpha^{*,n+1} = \bar{\mathbf{u}}_\alpha + (\mathbf{v}_\alpha^n - \bar{\mathbf{u}}_\alpha) e^{-\gamma \Delta t} + \frac{\sigma}{m_\alpha} \mathbf{R}_{v,\alpha}^n(\Delta t) \quad (43)$$

$$\mathbf{x}_\alpha^{*,n+1} = \mathbf{x}_\alpha^n + (\mathbf{v}_\alpha^n - \bar{\mathbf{u}}_\alpha) \frac{1 - e^{-\gamma \Delta t}}{\gamma} + \bar{\mathbf{u}}_\alpha \Delta t + \frac{\sigma}{\gamma m_\alpha} \mathbf{R}_{x,\alpha}^n(\Delta t), \quad (44)$$

and

$$\Delta t \mathbf{f}_\alpha^{n+1} = m_\alpha (\mathbf{v}_\alpha^{*,n+1} - \mathbf{v}_\alpha^n) \quad (45)$$

$$\mathbf{F}^{n+1}(\mathbf{x}) = - \sum_\alpha \mathbf{f}_\alpha^{n+1} \delta_\epsilon(\mathbf{x} - \tilde{\mathbf{x}}_\alpha^{n+1}) \quad (46)$$

gives the fluid-particle coupling centered at t^{n+1} .

Note that the random variables \mathbf{R} appearing in the corrector are identical to those used in the predictor. The provisional terms $\mathbf{x}_\alpha^{*,n+1}$, $\mathbf{v}_\alpha^{*,n+1}$ are corrected to enforce constraints (9) and (10) following the procedures used in the particle predictor step; the Debye-Hückel forces are re-calculated in between the two constraints to prevent rod crossing as described in the predictor step.

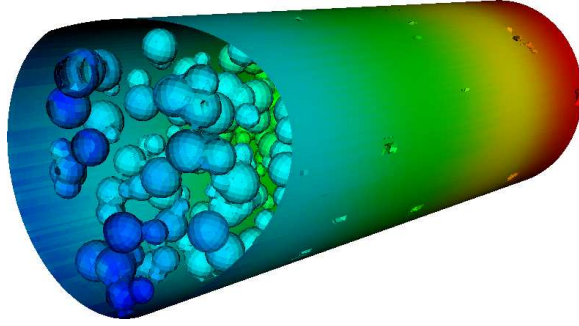


Fig. 2. 3D continuum model of packed bed geometry represented by 200 randomly packed spheres ($30\mu\text{m} < r < 80\mu\text{m}$) in a cylinder ($L = 2\text{cm}$, $R = 500\mu\text{m}$) for a $Re = 0.02$ flow. Grid spacing is $h = 1/64$. Pressure data shown.

Step 4: Fluid Corrector

We update the fluid velocity and pressure gradient using a projection method to enforce the incompressibility constraint [5,2]. This divergence-cleaning resembles the predictor step except that the quantities are now cell-centered:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\frac{1}{\rho}\nabla\pi^{n-\frac{1}{2}} - [(\mathbf{u} \cdot \nabla)\mathbf{u}]^{n+\frac{1}{2}} + \frac{1}{2\rho}(\mathbf{F}^n + \mathbf{F}^{n+1}) + \nu\Delta\mathbf{u}^* \quad (47)$$

$$\frac{\Delta t}{\rho}\Delta\pi^{n+\frac{1}{2}} = \nabla \cdot \left[\mathbf{u}^* + \frac{\Delta t}{\rho}\nabla\pi^{n-\frac{1}{2}} \right] \quad (48)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho}\nabla \left[\pi^{n+\frac{1}{2}} - \pi^{n-\frac{1}{2}} \right] \quad (49)$$

where π also appears in (38).

5 Results and Discussion

The application for the modeling capability described above is flow and transport of DNA in a microfluidic device. We are specifically interested in modeling flows in microchannel configurations used for DNA extraction and size-separation. These processes are typically performed in packed bed columns or post array channels. The latter lends itself to 2D modeling. We can employ the embedded boundary technique to treat these geometries. Figure 2 depicts the pressure drop in a packed bed represented by randomly placed microspheres in a micro-cylinder. The purpose of this result is to demonstrate the 3D computational capability for the macroscale continuum flow. Similarly, Figure 3a depicts a 2D representation of a dense post array microchannel. In these results, the array of obstructions in the flow—both homogeneous in Figure 3b and heterogeneous in Figure 2—create a pressure differential in the transverse direction, as in Figure 3b. This bulk flow effect is different from

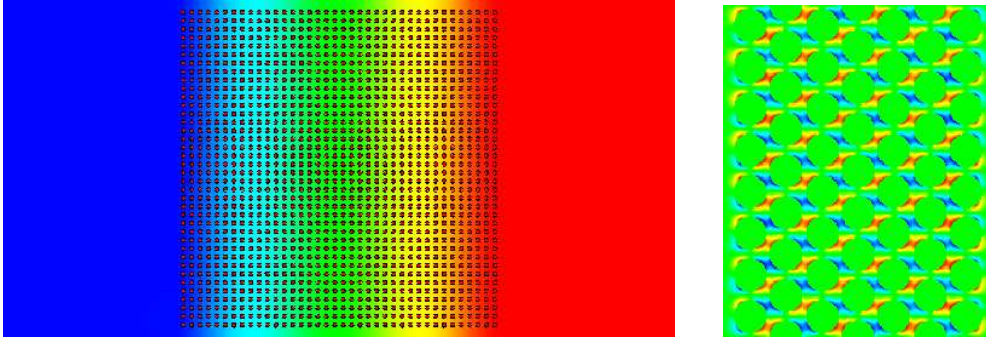


Fig. 3. (a) 2D continuum model of a “pillar chip” channel represented by a 38x38 post ($r = 12.5\mu\text{m}$) ordered array in a channel () for a $Re = 0.02$ flow (left). Pressure data shown. (b) Non-zero pressure gradient in the vertical direction due to channel obstructions (right).

pressure-driven flows with no obstructions in the flow where the flow direction is the dominant gradient vector. The flow parameters for the packed bed are a plug inflow velocity of 0.01cm/sec, $Re = 0.02$, cylinder length is 2cm, cylinder width is 1mm and the 200 randomly packed spheres range in radius from 30-80 μm . The flow parameters for the straight post array are a plug inflow velocity of 0.01cm/sec, $Re = 0.02$, channel length is 4mm, width is 2mm and the 38x38 posts in an ordered array each have a radius of 12.5 μm . The staggered post array is a smaller length section of a channel ($L = 400\mu\text{m}$) with larger posts ($r = 50\mu\text{m}$). The idealized geometries in these 2D and 3D results are generated using the following implicit function on the grid:

$$\phi(x) = \min_k (|\vec{x} - \vec{x}_k|^2 - r_k^2) \quad (50)$$

where \vec{x}_k = center of k^{th} sphere, r_k = radius of k^{th} sphere and $\vec{x} : \phi(\vec{x}) = 0$ on the boundary.

The 2D pillar chip model can be restricted to a single molecule traveling through a smaller section of the array. This is the test problem in which we implement the new short range interactions into our algorithm. Figure 4 is a comparison between the hybrid method including the repulsive Debye-Hückel potential between rods (noncrossing) and the hybrid method which does not treat intra-polymer interactions at all (crossing) as in [26]. In the noncrossing simulation there is a clear separation of the polymer from itself at all locations. In the crossing simulation the beads tend to bunch up on top of each other in several locations along the polymer. The rods do cross as depicted by colored particles which are out of sequence. A smooth colormap is used to represent the particle number in the polymer chain. Clumping of particles on top of one another is an artifact of visualization, similar to any particles that cross the boundary. In these simulations the physical parameters of the polymer are meant to closely resemble genomic DNA at constant temperature $T = 300\text{K}$

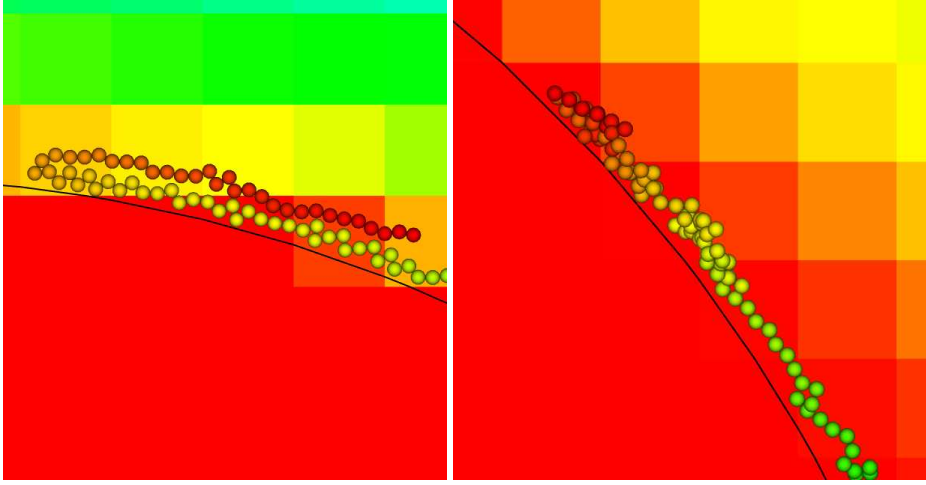


Fig. 4. Bead-rod polymer representation of DNA flowing around a pillar in a 2D post microarray. (left) Hybrid algorithm with repulsive Debye-Hückel potential between rods to prevent crossing. There is a clear separation of the polymer from itself at all locations. (right) Hybrid algorithm with no treatment of intra-polymer interactions. Rods are allowed to cross. The color of the beads represents the number of the bead in the polymer chain using a smooth colormap. Clumping of beads on top of one another is an artifact of visualization. Crossing of rods occurs where the color of a bead is out of order sequentially, particularly in the red, orange and yellow to green bead areas.

and have been corrected according to an assumed Kuhn length for double-stranded DNA of 100nm: 200 nodes at $m = 3 \times 10^{-19} \text{g/node}$, separated by rods of length $a = 100 \text{nm}$ with a Stokes drag coefficient of $m\gamma = 1.0 \times 10^{-6} \text{cm}^2/\text{sec}$. We use a rough estimate to the Debye screening length $k = 10 \text{nm}$ and choose the constant $A = 1.0 \times 10^{-17}$. The flow parameters are a $40 \mu\text{m}$ square channel with a single cylindrical pillar obstruction ($r = 10 \mu\text{m}$) at $Re = 0.4$ with plug inflow velocity 1cm/sec , density $\rho = 1 \text{g/cm}^3$ and dynamic viscosity of water 0.01g/(cm - sec) .

6 Conclusion

We have presented a hybrid fluid-particle algorithm in which the fluid-particle forces are tightly coupled, and which includes short range Debye-Hückel model to enforce the physical rod-crossing constraint. The calculations shown here use a perfectly elastic collision model to enforce the particle-wall constraints, though we have also implemented this using Debye-Hückel elsewhere [4].

By itself, the fluid dynamics part of our hybrid fluid-particle approach is governed by a time scale set by the advective CFL condition. The particle algorithms are stable with a somewhat shorter time step. In this paper, we maintain tight coupling between the fluid and particles by marching both me-

dia at the same reduced time step. The computational cost of a single time step is strongly controlled by the explicit advection step in the fluid, however, so reducing the fluid timestep does increase the cost of the method. Elsewhere, we describe efforts to reduce this cost by implementing new particle algorithms which are subject to significantly less restrictive stability conditions [17].

We present sample calculations spanning several of the length scales relevant to our goal of system-level modeling: from a resolved view of polymers interacting with embedded posts, to a continuum-level result for a packed bed reactor. One of our goals is to integrate these approaches through the adaptive mesh and algorithm refinement technique (AMAR) [10]. In this method, those parts of a domain requiring molecular resolution can be handled with algorithms like dissipative particle dynamics; those parts in which the polymer is dilute, but in which its dynamics can be more crudely approximated, can be simulated with the methods described in this paper; and any parts where a continuum approach is sufficient can be handled with the methods of, e.g., [22]. Through this multiscale hierarchy of algorithms, efficient and predictive simulation of complete microfluidic devices may be attained.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48. The work of G. H. Miller at the University of California, Davis was supported by LLNL IUT subcontracts number B550201 and number B553964, and by DOE MICS contract number DE-FG02-03ER25579. The work of M. D. Bybee at the Lawrence Livermore National Laboratory was supported by the Department of Energy Computational Science Graduate Fellowship Program under grant number DE-FG02-97ER25308.

References

- [1] M. Barad, P. Colella, D. T. Graves, B. van Straalen, and D. Trebotich. An adaptive, Cartesian grid embedded boundary method for the incompressible Navier Stokes equations. *J. Comp. Phys.*, 2006. in preparation.
- [2] J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *J. Comp. Phys.*, 85:257–283, 1989.
- [3] C. K. Birdsall and A. B. Langdon. *Plasma Physics via Computer Simulation*. Institute of Physics, Philadelphia, 1991.

- [4] M. D. Bybee, G. H. Miller, and D. Trebotich. Particle interactions in DNA flows. Technical Report UCRL-TR-217843, LLNL, December 2005.
- [5] A. J. Chorin. Numerical solutions of the Navier-Stokes equations. *Math. Comp.*, 22:745–762, 1968.
- [6] G. Ciccotti, M. Ferrario, and J.-P. Ryckaert. Molecular dynamics of rigid systems in cartesian coordinates: A general formulation. *Molec. Phys.*, 47:1253–1264, 1982.
- [7] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *J. Comput. Phys.*, 87:171–200, 1990.
- [8] P. Colella, D. T. Graves, B. Keen, and D. Modiano. A Cartesian grid embedded boundary method for hyperbolic conservation laws. 211:347–366, 2006.
- [9] T. Deschamps, P. Schwartz, D. Trebotich, P. Colella, D. Saloner, and R. Malladi. Vessel segmentation and blood flow simulation using level-sets and embedded boundary methods. In *International Congress Series*, volume 1268, pages 75–80, June 2004.
- [10] A. L. Garcia, J. B. Bell, W. Y. Crutchfield, and B. J. Alder. Adaptive mesh and algorithm refinement using direct simulation Monte Carlo. *J. Comp. Phys.*, 154:134–155, 1999.
- [11] J. S. Hur and E. G. S. Shaqfeh. Brownian dynamics simulations of single DNA molecules in shear flow. *Journal of Rheology*, 44(4):713–742, 2000.
- [12] H. Johansen and P. Colella. A Cartesian grid embedded boundary method for Poisson’s equation on irregular domains. *J. Comp. Phys.*, 147(2):60–85, December 1998.
- [13] H. Johansen and P. Colella. A Cartesian grid embedded boundary method for Poisson’s equation on irregular domains. *J. Comp. Phys.*, 147:60–85, 1998.
- [14] P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, New York, 1980.
- [15] S. Kumar and R. G. Larson. Brownian dynamics simulations of flexible polymers with spring-spring repulsions. *Journal of Chemical Physics*, 114(15):6937–6941, 2001.
- [16] P. McCorquodale, P. Colella, and H. Johansen. A cartesian grid embedded boundary method for the heat equation on irregular domains. *J. Comp. Phys.*, 173:620–635, 2001.
- [17] G. H. Miller and D. Trebotich. Toward a mesoscale model for the dynamics of polymer solutions. *J. Comput. Theor. Nanosci.*, 2006. to appear.
- [18] M. L. Minion. On the stability of Godunov-projection methods for incompressible flow. *J. Comp. Phys.*, 123(2):435–449, 1996.

- [19] J. T. Padding and W. J. Briels. Uncrossability constraints in mesoscopic polymer melt simulations: Non-Rouse behavior of $C_{120}H_{242}$. *Journal of Chemical Physics*, 115(6):2846–2859, 2001.
- [20] P. O. Schwartz, M. Barad, P. Colella, and T. J. Ligoeki. A Cartesian grid embedded boundary method for Poisson’s equation and the heat equation in three dimensions. 211:531–50, 2006.
- [21] M. Somasi, B. Khomami, N. J. Woo, J. S. Hur, and E. S. G. Shaqfeh. Brownian dynamics simulations of bead-rod and bead-spring chains: numerical algorithms and coarse-graining issues. *Journal of Non-Newtonian Fluid Mechanics*, 108:227–255, 2002.
- [22] D. Trebotich, P. Colella, and G. H. Miller. A stable and convergent scheme for viscoelastic flow in contraction channels. *J. Comp. Phys.*, 205:315–342, 2005.
- [23] D. Trebotich, P. Colella, G. H. Miller, A. Nonaka, T. Marshall, S. Gulati, and D. Liepmann. A numerical algorithm for complex biological flow in irregular microdevice geometries. In *Technical Proceedings of the 2004 Nanotechnology Conference and Trade Show*, volume 2, pages 470–473, 2004.
- [24] D. Trebotich, T. Deschamps, and P. Schwartz. Air-flow simulation in realistic models of the trachea. In *Technical Proceedings of the 26th Annual International Conference IEEE Engineering in Medicine and Biology Society*, pages 3933–3936, September 2004.
- [25] D. Trebotich, G. H. Miller, and M. D. Bybee. A hard constraint algorithm to model particle interactions in DNA-laden flows. *Nanoscale & Microscale Thermophys. Engr.*, 2006. submitted. Also available as LLNL Technical Report UCRL-JRNL-223318.
- [26] D. Trebotich, G. H. Miller, P. Colella, D. T. Graves, D. F. Martin, and P. O. Schwartz. A tightly couple particle-fluid model for DNA-laden flows in complex microscale geometries. *Computational Fluid and Solid Mechanics*, pages 1018–1022, 2005.
- [27] B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method. *J. Comput. Phys.*, 32:101–136, 1979.