A tightly coupled particle–fluid model for DNA-laden flows in complex microscale geometries

D. Trebotich[a,*], G.H. Miller[b,c], P. Colella[c], D.T. Graves[c], D.F. Martin[c], P.O. Schwartz[c]

[a] *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, PO Box 808, L-560, Livermore, CA 94551, USA*
[b] *Department of Applied Science, University of California, One Shields Avenue, Davis, CA 95616, USA*
[c] *Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA*

Abstract

We present a stable and convergent method for the computation of flows of DNA-laden fluids in microchannels with complex geometry. The numerical strategy combines a ball–rod model representation for polymers coupled tightly with a projection method for incompressible viscous flow. We use Cartesian grid embedded boundary methods to discretize the fluid equations in the presence of complex domain boundaries. A sample calculation is presented showing flow through a packed array microchannel in two dimensions.

Keywords: Incompressible Navier–Stokes; Stochastic particle dynamics; DNA; Microfluidics; Embedded boundary method

1. Introduction

Modeling complex biological fluids is a challenge because these types of flows are not well understood and the constitutive behavior of these types of fluids is not easily represented. Modeling is complicated further when restricted to the microscale due to the presence of large particles in the fluid whose molecular lengths are comparable to the flow geometry. For example, a highly concentrated solution of suspended polymer molecules may be represented at large scales with a continuum Oldroyd-B constitutive model [1]. However, when the geometry length scales are comparable to the inter-polymer spacing, a continuum approximation is no longer appropriate. Additionally, when the length scale of the geometry is comparable to the length of an individual polymer macromolecule, new physical behavior may be observed. Here, we are concerned with this dilute microscale limit, which finds application in microfluidic biomedical processing and sensor technology. Our model will consider discrete polymers or macromolecules suspended in an incompressible viscous solvent.

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

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

$$\nabla\cdot\mathbf{u} = 0. \tag{2}$$

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

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

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

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

* Corresponding author. Tel.: +1 925 423 2873; Fax: +1 925 422 2873; E-mail: treb@llnl.gov

Computational Fluid and Solid Mechanics 2005
K.J. Bathe (Editor) **Paper no. 234-196**

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

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

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

$$\sigma_\alpha = \sqrt{2m_\alpha \gamma k_B T}, \tag{7}$$

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

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

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

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

In addition to the incompressibility condition in Eq. (2), we have two additional constraints: (i) that interparticle spacing is constant

$$\left\| \mathbf{x}_\alpha - \mathbf{x}_\beta \right\| = a \tag{9}$$

if particles α and β represent adjacent nodes in a ball–rod polymer representation and (ii) that particles cannot pass through a physical boundary

$$\mathbf{x}_\alpha \in \Omega \tag{10}$$

2. Numerical method

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. The various operators 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. This approach has been used as the basis for second-order accurate methods for elliptic, parabolic, and hyperbolic partial differential equations (PDE) in two and three dimensions [2–5]. These methods also have been combined using the predictor–corrector approach [6] to provide a second-order accurate method for the incompressible Navier–Stokes equations for problems in irregular geometries [7], which is the underlying fluids algorithm for the present work.

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. In the following, the discrete divergence $\nabla\cdot$, discrete gradient ∇, and discrete Laplacian Δ operators use standard symmetric second-order discretizations in interior regions of the flow. These operators are modified by the presence of boundaries, as described above.

A tilde ($\sim$) will be used to denote quantities computed in the predictor step of our predictor–corrector strategy; no tilde is used for the corrector. Superscripts * and † will denote a 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. Where it provides clarity, the subscripts *cc* and *ec* will be used to denote cell- and edge-centered quantities, respectively.

Our approach to solving Eqs (1) and (2) is a projection method based on Bell et al. [6]. This fluid solver is coupled tightly to the particle solver with a predictor–corrector strategy. To advance the solution through a single time step Δt consists of the following four steps, in sequence:

2.1. Step 1: particle predictor

We base our solution to the particle equations on $\mathcal{O}(\Delta t^{2.5})$ – accurate Ito–Taylor [8] expansions of the Langevin equations for variables x_α and $e^{\gamma t}\, v_\alpha$.

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

$$\tilde{\mathbf{v}}_\alpha^{*,n+1} = \mathbf{u}^n(\mathbf{x}_\alpha^n) + (\mathbf{v}_\alpha^n - \mathbf{u}^n(\mathbf{x}_\alpha^n))\mathrm{e}^{-\gamma\Delta t} + \frac{\sigma}{m_\alpha}\mathbf{R}_{v,\alpha}^n(\Delta t) \tag{11}$$

$$\tilde{\mathbf{x}}_\alpha^{*,n+1} = \mathbf{x}_\alpha^n + (\mathbf{v}_\alpha^n - \mathbf{u}^n(\mathbf{x}_\alpha^n))\frac{1 - \mathrm{e}^{-\gamma\Delta t}}{\gamma} + \mathbf{u}^n(\mathbf{x}_\alpha^n)\Delta t + \frac{\sigma}{\gamma m_\alpha}\mathbf{R}_{x,\alpha}^n(\Delta t) \tag{12}$$

$$\Delta t\, \mathbf{f}_\alpha^n = m_\alpha\left(\tilde{\mathbf{v}}_\alpha^{*,n+1} - \mathbf{v}_\alpha^n\right) \tag{13}$$

$$\mathbf{F}^n(\mathbf{x}) = -\sum_\alpha \mathbf{f}_\alpha^n \delta_\varepsilon(\mathbf{x} - \mathbf{x}_\alpha^n), \tag{14}$$

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

$$\left\langle R_v(\Delta t)^2 \right\rangle = \frac{1}{2\gamma}(1 - \mathrm{e}^{-2\gamma\Delta t}) \tag{15}$$

$$\left\langle R_x(\Delta t)^2 \right\rangle = \frac{1}{2\gamma}\left[2\gamma\Delta t - \mathrm{e}^{-2\gamma\Delta t} + 4\mathrm{e}^{-\gamma\Delta t} - 3\right] \tag{16}$$

$$\langle R_x(\Delta t) R_v(\Delta t) \rangle = \frac{1}{2\gamma}\left(1 - \mathrm{e}^{-\gamma\Delta t}\right)^2 \tag{17}$$

for variables of identical vector index, and all covariances are zero for terms with different vector indices. In Eq. (11), $\mathbf{u}^n(\mathbf{x}_\alpha^n)$ is evaluated by linear interpolation of

the cell-centered discretization $\mathbf{u}^n_{cc}$. The discrete Dirac delta function is represented using a particle-in-cell or cloud-in-cell model[9].

The particle coordinates $\tilde{\mathbf{x}}^*$ do not in general obey the constraint given in Eq. (9). To enforce this condition, we use the Lagrange multiplier technique [10]. This correction consists of iterative solution of a tridiagonal linear system obtained by linearization of Eq. (9). We refer to this corrected state as $\tilde{\mathrm{x}}\dagger$, and

$$\tilde{\mathbf{v}}^{\dagger}_{\alpha} = \tilde{\mathbf{v}}^{n+1}_{\alpha} + \frac{1}{\Delta t}(\tilde{\mathbf{x}}^{\dagger}_{\alpha} - \tilde{\mathbf{x}}^{*,n+1}_{\alpha}) \tag{18}$$

is the corresponding corrected velocity.

The time-linear trajectory $\mathbf{x}^n_{\alpha} \rightarrow \tilde{\mathbf{x}}^{\dagger}_{\alpha}$ may carry particle α across the fluid domain boundary, thereby violating the constraint of Eq. (10). We use a continuous distance function representation of the domain boundary to detect such collisions. If the trajectory contacts the domain at a point $\chi \in \delta\Omega$ at relative time τ, $0 < \tau \leq \Delta t$, then we elastically 'bounce' the particle off the boundary at χ as follows. Let $\mathbf{n}$ be unit normal to the boundary at χ:

$$\tilde{\mathbf{v}}^{n+1} = \tilde{\mathbf{v}}^{\dagger} - 2(\mathbf{n}\cdot\tilde{\mathbf{v}}^{\dagger})\mathbf{n} \tag{19}$$

$$\tilde{\mathbf{x}}^{n+1}_{\alpha} = \mathcal{X} + (\Delta t - \tau)\tilde{\mathbf{v}}^{n+1}. \tag{20}$$

If no collision is indicated, then $\tilde{\mathbf{v}}^{n+1}_{\alpha} = \tilde{\mathbf{v}}^{\dagger}_{\alpha}$, etc.

2.2. Step 2: fluid predictor

We begin the fluid calculation by estimating edge- and time-centered velocities $\mathbf{u}^{*,n+\frac{1}{2}}_{\mathrm{ec}}$ e.g. $\mathrm{u}^{*,n+\frac{1}{2}}_{i+\frac{1}{2},j,k}$ using an upwind Taylor series expansion of Eq. (1), including a transverse velocity correction [11], an explicit determination of the viscous source term, and the explicit source $\mathbf{F}^n$, but omitting the pressure. These provisional edge states are then made divergence-free with a Marker-and-cell-stencil Hodge projection,

$$\mathbf{u}^{n+\frac{1}{2}}_{ec} = (I - \nabla\Delta^{-1}\nabla\cdot)\mathbf{u}^{*,n+\frac{1}{2}}_{ec} \tag{21}$$

The edge states $\mathbf{u}^{n+\frac{1}{2}}_{ec}$ are used to estimate the term $(\mathbf{u}\cdot\nabla)\mathbf{u}$ appearing below. Then,

$$\frac{\tilde{\mathbf{u}}^{n+1}_{cc} - \mathbf{u}^n_{cc}}{\Delta t} = -\frac{1}{\rho}(\nabla\pi^{n-\frac{1}{2}}_{cc}) - [(\mathbf{u}\cdot\nabla)\mathbf{u}]^{n+\frac{1}{2}}_{cc} + \frac{1}{\rho}F^n_{cc} + \nu\Delta\tilde{\mathbf{u}}^{n+1}_{cc} \tag{22}$$

is solved implicitly for the time$-(n+1)$ cell-centered velocity field $\tilde{\mathbf{u}}^{n+1}_{cc}$. Here, π is a cell-centered pressure estimate carried over from a previous time step (see Eq. (31)). One would normally center the viscous source in time, but in complex geometries a Crank–Nicholson discretization is not stable [3,12].

2.3. Step 3: particle corrector

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

$$\bar{\mathbf{u}}_{\alpha} = \frac{\mathbf{u}^n(\mathbf{x}^n_{\alpha}) + \tilde{\mathbf{u}}^{n+1}(\tilde{\mathbf{x}}^{n+1}_{\alpha})}{2} \tag{23}$$

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

$$\bar{\mathbf{u}}_{\alpha} = \frac{\tau}{2\Delta t}\mathbf{u}^n(\mathbf{x}^n_{\alpha}) + \frac{(\Delta t - \tau)}{2\Delta t}(I - 2\mathbf{n}\mathbf{n}^T)\tilde{\mathbf{u}}^{n+1}(\tilde{\mathbf{x}}^{n+1}_{\alpha}) \tag{24}$$

if it was predicted to have bounced. Eq. (24) 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

$$\mathbf{v}^{*,n+1}_{\alpha} = \bar{\mathbf{u}}_{\alpha} + (\mathbf{v}^n_{\alpha} - \bar{\mathbf{u}}_{\alpha})\,\mathrm{e}^{-\gamma\Delta t} + \frac{\sigma}{m_{\alpha}}\mathbf{R}^n_{v,\alpha}(\Delta t) \tag{25}$$

$$\mathbf{x}^{*,n+1}_{\alpha} = \mathbf{x}^n_{\alpha} + (\mathbf{v}^n_{\alpha} - \bar{\mathbf{u}}_{\alpha})\frac{1 - \mathrm{e}^{-\gamma\Delta t}}{\gamma} + \bar{\mathbf{u}}_{\alpha}\Delta t + \frac{\sigma}{\gamma m_{\alpha}}\mathbf{R}^n_{x,\alpha}(\Delta t) \tag{26}$$

and

$$\Delta t\,\mathbf{f}^{n+1}_{\alpha} = m_{\alpha}\left(\mathbf{v}^{*,n+1}_{\alpha} - \mathbf{v}^n_{\alpha}\right) \tag{27}$$

$$\mathbf{F}^{n+1}(\mathbf{x}) = -\sum_{\alpha}\mathbf{f}^{n+1}_{\alpha}\delta_{\varepsilon}(\mathbf{x} - \tilde{\mathbf{x}}^{n+1}_{\alpha}) \tag{28}$$

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 $\mathbf{x}^{*,n+1}_{\alpha}$, $\mathbf{v}^{*,n+1}_{\alpha}$ are corrected to enforce the constraints in Eqs (9) and (10) following the procedures used in the particle predictor step.

2.4. Step 4: fluid corrector

The fluid corrector calculation resembles the predictor and, in particular, uses the same calculation of $(\mathbf{u}\cdot\nabla)\mathbf{u}$; i.e. the forcing used to estimate $\mathrm{u}^{n+\frac{1}{2}}_{ec}$ remains time-centered. We use the so-called 'pressure formulation' projection strategy:

$$\frac{\mathbf{u}^*_{cc} - \mathbf{u}^n_{cc}}{\Delta t} = -\frac{1}{\rho}\nabla\pi^{n-\frac{1}{2}} - [(\mathbf{u}\cdot\nabla)\mathbf{u}]^{n+\frac{1}{2}}_{cc} + \frac{1}{2\rho}(\mathbf{F}^n_{cc} + \mathbf{F}^{n+1}_{cc}) + \nu\Delta\mathbf{u}^* \tag{29}$$

$$\frac{\Delta t}{\rho}\Delta\pi^{n+\frac{1}{2}}_{cc} = \nabla\cdot\left[\mathbf{u}^*_{cc} + \frac{\Delta t}{\rho}\nabla\pi^{n-\frac{1}{2}}_{cc}\right] \tag{30}$$

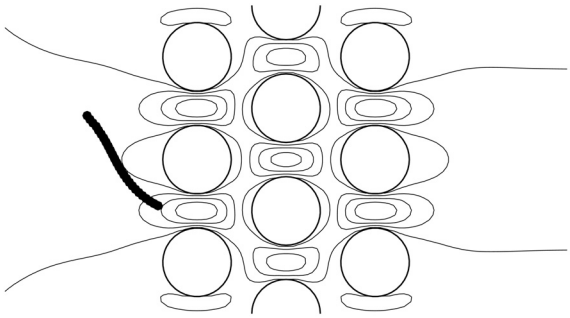

Fig. 1. Contours of horizontal velocity at 0.1, 0.2, 0.3, and 0.4 cm/s. Time 0.0956576 s.

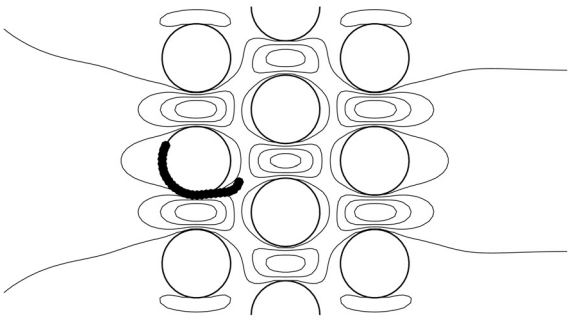

Fig. 2. Contours of horizontal velocity at 0.1, 0.2, 0.3 and 0.4 cm/s. Time 0.381831 s.

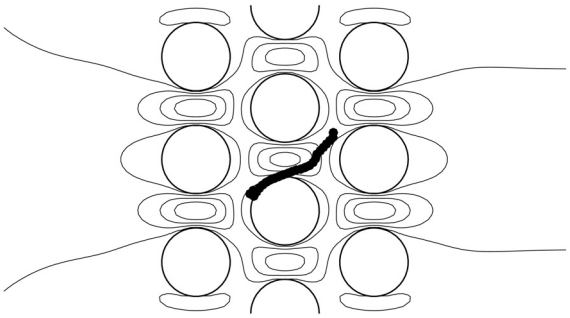

Fig. 3. Time 0.668005 s.

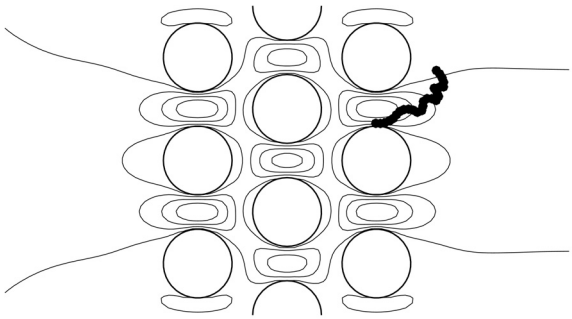

Fig. 4. Time 0.906483 s.

$$\mathbf{u}_{cc}^{n+1} = \mathbf{u}_{cc}^{*} - \frac{\Delta t}{\rho}\nabla\left[\pi_{cc}^{n+\frac{1}{2}} - \pi_{cc}^{n-\frac{1}{2}}\right] \tag{31}$$

A divergence-cleaning projection (see Eqs (30) and (31)) is applied at this step, resulting in the cell- and time-centered pressure estimate π appearing also in Eq. (22).

3. Conclusions and discussion

Sample calculations are displayed in Figs 1–4. The run parameters (width 450 μm, $Re = 0.45$, $a = 6.8\,\mu$m, $\gamma = 10^{12}$/s, $m = 1.9 \times 10^{-17}$ g, $\rho = 1$ g/cm^3) approximate DNA in an actual microfluidic device used for extraction. The left boundary condition is plug flow with a velocity of 0.1 cm/s; the right boundary is outflow (homogeneous Neumann); the top and bottom boundaries, and the interior circular boundaries, are solid wall. The polymer is a 26-node approximation of DNA, introduced near the left boundary as an inclined linear array after the fluid flow field reached steady state. The polymer's trajectory causes it to become wrapped around the first circular element, where it remains pinned until the stochastic perturbations work it loose.

The fluid dynamic steps of this method are subject to an advective Courant–Friedrichs–Lewy (CFL) stability condition only. The particle steps, without constraints, are also stable with this CFL timestep. When particles move far from the constraint manifold (Eq. (9)), however, the Lagrange multiplier algorithm of Ciccotti et al. [10] may diverge. We have found that the maximum particle displacement per timestep for which the Ciccotti et al. algorithm is stable may be extended for most systems by centering the constraint force at the conclusion of the time step; rather than at the start. With this modification, a maximum particle displacement of $\mathcal{O}(a/10)$ works well.

The stochastic term $\mathbf{R}_x$ is unbounded, and thus for any CFL-limited hydrodynamic time step Δt there may be particle displacements much greater than $a/10$, resulting in convergence problems. In the current implementation, we limit our stochastic variables $\mathbf{R}$ to lie within ± 3 standard deviations of zero. Statistically, this results in 0.26% of random numbers being truncated. In our numerical tests, this has not yet resulted in discernible negative consequences. An alternative approach that we will explore is to compute particle trajectories with adaptive time steps that are decoupled from the fluid dynamic time steps.

We use a backward Euler time stepping strategy in, for example, (22,290), which is formally first-order accurate. To make the overall method second-order, it will be necessary to replace at least Eq. (29) with a Runge–Kutta time-stepping strategy [13], which has been used in a computational context similar to ours [3].

4. Acknowledgments

This work was performed under the auspices of the US Department of Energy (DOE) by the University of California, Lawrence Livermore National Laboratory under contract no. W-7405-Eng-48. Work at the Lawrence Berkeley National Laboratory was supported by the US DOE Mathematical, Information, and Computer Sciences (MICS) Division under contract number DE-AC03-76SF00098. Work at the University of California, Davis, was partially supported by the US DOE MICS Division under contract number DE-FG02-03ER25579.

References

[1] Trebotich D, Colella P, Miller GH. A stable and convergent scheme for viscoelastic flow in contraction channels. J Comp Phys 2004, Accepted for publication.
[2] Johansen H, Colella, P. A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. J Comp Phys 1998;147:60–85.
[3] McCorquodale P, Colella P, Johansen H. A cartesian grid embedded boundary method for the heat equation on irregular domains. J Comp Phys 2001;173:620–635.
[4] Colella P, Graves DT, Keen B, Modiano, D. A Cartesian grid embedded boundary method for hyperbolic conservation laws. Submitted for publication, 2004.
[5] Schwartz PO, Barad M, Colella P, Ligocki TJ. A Cartesian grid embedded boundary method for Poisson's equation and the heat equation in three dimensions. Submitted for publication, 2004.
[6] Bell JB, Colella P, Glaz HM. A second-order projection method for the incompressible Navier–Stokes equations. J Comp Phys 1989;85:257–283.
[7] Barad M, Colella P, Graves DT, Ligocki TJ, Schwartz PO, Trebotich D. A Cartesian grid embedded boundary method for incompressible flow. In preparation, 2005.
[8] Kloeden PE, Platen E. Numerical Solution of Stochastic Differential Equations. New York: Springer-Verlag, 1980.
[9] Birdsall CK, Langdon AB. Plasma Physics via Computer Simulation. Philadelphia: Institute of Physics, 1991.
[10] Ciccotti G, Ferrario M, Ryckaert J-P. Molecular dynamics of rigid systems in cartesian coordinates: a general formulation. Molec Phys 1982;47:1253–1264.
[11] Minion ML. On the stability of Godunov-projection methods for incompressible flow. J Comp Phys 1996;123:435–449.
[12] Johansen H, Colella P. A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. J Comp Phys 1998;147:60–85.
[13] Twizell EH, Gumel AB, and Arigu MA. Second-order L_0-stable methods for the heat equation with time-dependent boundary conditions. Adv Comp Math 1996;6:333–352.