Molecular Simulation
Vol. 37, No. 8, July 2011, 738–745

A higher-order accurate fluid-particle algorithm for polymer flows

B. Kallemov[a], G.H. Miller[a]* and D. Trebotich[b]*

[a]*University of California, Davis, CA 95616, USA;* [b]*Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 50A-1148, Berkeley, CA 94720, USA*

(*Received 24 September 2010; final version received 19 November 2010*)

We present a new algorithm for the simulation of polymer-laden flows in microscale environments. Our algorithm is based on a hybridisation of high-order accurate continuum and particle methods. The continuum algorithm provides the basic framework for high-performance computations to resolve device length and time scales. It is coupled to a new particle method with an optimised treatment of particle interactions such that the time step is on the level of the fluid continuum. We demonstrate our simulation capability on the flow of polymers in a contraction microchannel used for single-molecule detection.

Keywords: hybrid algorithm; fluid-particle coupling; polymer flow

1. Introduction

Understanding complex biological flows through advanced algorithmic modelling is critical to several important biomedical applications such as targeted drug delivery coupled with continuous monitoring and diagnostics. These applications will leverage miniaturised technology based on the advancements in microfluidics and nanofluidics. In order for this development to continue towards the design of optimised trustworthy working devices, advanced modelling and simulation tools are needed to understand the fundamental physics and chemistry of biological fluids at much smaller than normal scales. This will enable shorter design and fabrication cycles and ultimately get devices to market more quickly and with less cost.

Modelling complex biological fluids is a challenge, because their non-Newtonian constitutive behaviour is not easily represented. The problem is further complicated when the flow of biological fluids is restricted to the small length scales of state-of-the-art biomedical devices. At these scales, new fluid mechanical and modelling issues arise, because (1) surface-to-volume ratios are extremely large and (2) characteristic lengths of the macromolecules or cells approach those of the flow geometry. For example, a highly concentrated solution of suspended polymer molecules may be represented at large, system-level scales with a continuum viscoelastic constitutive model [1]. However, when the geometry length scales are comparable to the inter-polymer spacing, a continuum approximation is no longer appropriate and a discrete molecular approximation is needed. In addition, when the length scale of the geometry is comparable to the length of an individual polymer macromolecule, new physical behaviour may be observed near surfaces where velocity and concentration gradients tend to be large and macromolecular shear degradation or scission can occur as a result. This dynamic can be beneficial in a DNA-amplification device, e.g., but detrimental in a drug-delivery system. The discrete representation of particles suspended in a fluid is needed in this case to predict the fate of individual molecules.

In a previous work, we had first developed a hybrid fluid-particle algorithm to model freely jointed polymers coupled with a viscous solvent [2]. The algorithm captured qualitatively the behaviour of a polymer molecule in a microarray channel, namely stably enforcing the rod length constraint and bead-surface collisions. We improved this algorithm in regard to overall stability such that the particle time step was on the order of the fluid time step by implementing a velocity constraint (in addition to the position constraint) on the particles [3]. This algorithm was extended to include additional features like a rod non-crossing constraint [4]. A new particle method was developed by introducing an exponential integrator and proved to be stable for long time (i.e. fluid dynamics time scales as opposed to molecular dynamics timescales) in Refs [5,6].

In this paper, we have coupled the new particle method [6] with a high-performance and higher-order accurate fluid dynamics solver for complex geometry [7]. We consider a canonical flow that occurs in microfluidic detection devices: flow of individual DNA polymer strands through an abrupt contraction microchannel. In our flow demonstrations, we introduce two polymers

*Corresponding author. Email: grgmiller@ucdavis.edu; treb@lbl.gov

ISSN 0892-7022 print/ISSN 1029-0435 online

DOI: 10.1080/08927022.2010.543977
http://www.informaworld.com

in a pressure-driven flow through a sudden contraction microchannel. Abrupt contractions in microchannels are common for the purpose of flow control in a microfluidic device [8]. Here, the abrupt contraction is intended to mimic a single-molecule detection component in a larger system [9], where the molecules are threaded through a region for detection using fluid mechanical forces alone. The goal of these simulations is to predict optimal parameters for a flow-through device.

2. Model

We model a polymer as a collection of coupled point masses, each subject to the Langevin equation of motion [10] (as in [2–6,11]),

$$\frac{\partial \mathbf{x}_i}{\partial t} = \mathbf{v}_i, \tag{1}$$

$$\frac{\partial \mathbf{v}_i}{\partial t} = \gamma\big(\mathbf{u}(\mathbf{x}_i, t) - \mathbf{v}_i\big) + \mathbf{f}_i + \sigma\xi_i, \tag{2}$$

subject to the constraint

$$\|\mathbf{x}_i - \mathbf{x}_{i+1}\|_2 = a. \tag{3}$$

Here, $\mathbf{x}_i(t)$ is the position of the ith particle with mass m_i, $\mathbf{u}$ is the fluid velocity, $\mathbf{f}_i$ is the interparticle force acting on particle i, $\gamma > 0$ is the friction coefficient and $\xi(t)$ is a white noise representing stochastic thermal bombardment by the solvent. This assumes that the fluid velocity is decomposed into a smooth part $\mathbf{u}$ and a rapidly varying component ξ. The smooth part is influenced by stochastic motion of the particles, however, and is therefore formally a stochastic process. The constant σ is given by

$$\sigma = \sqrt{\frac{2\gamma k_{\mathrm{B}} T}{m_i}}, \tag{4}$$

where k_{B} is the Boltzmann constant and T is the temperature.

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

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \frac{1}{\rho}\nabla p = \nu\Delta\mathbf{u} + \frac{1}{\rho}\mathbf{f}, \tag{5}$$

$$\nabla\cdot\mathbf{u} = 0. \tag{6}$$

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

We consider the Navier–Stokes equations to be applicable to length scales that are large enough that thermal perturbations are averaged out, so $\mathbf{u}$ is smooth. Thus, in our analysis, the stochastic dependence of $\mathbf{u}$ in the Langevin equations is through the particle position $\mathbf{x}$ only: $\mathbf{u} = \mathbf{u}(t, \mathbf{x}(t))$. In the Navier–Stokes equations, $\mathbf{f}$ is

$$\mathbf{f}(\mathbf{x}, t) = \sum_i m_i\gamma\big(\mathbf{v}_i(t) - \mathbf{u}(\mathbf{x}_i(t), t)\big)\delta(\mathbf{x}(t) - \mathbf{x}_i(t)), \tag{7}$$

where the sum is over all particles. This force is stochastic through the position and velocities of the particles.

3. Numerical method

3.1 Particle solver

The numerical method for the integration of Equation (2) for particles was given in detail by Kallemov and Miller [6]. Here, we summarise the algorithm.

The rearrangement

$$\mathbf{z} = \mathbf{v}\mathrm{e}^{\gamma t}, \tag{8a}$$

$$\frac{\partial \mathbf{z}_i}{\partial t} = \mathrm{e}^{\gamma t}\big(\gamma\mathbf{u}(\mathbf{x}_i, t) + \mathbf{f}_i + \sigma\xi_i\big) \tag{8b}$$

permits the velocity solution

$$\mathbf{z}_i(t) = \mathbf{z}_i(0) + \int_0^t \big(\gamma\mathbf{u}(\mathbf{x}_i(s), s) + \mathbf{f}_i + \sigma\xi_i(s)\big)\mathrm{e}^{\gamma s}\mathrm{d}s. \tag{9}$$

An implicit function for the constraints is obtained by the method of Lagrange multipliers. The Lagrange multipliers λ are obtained as solutions of linear system [6]

$$\mathbf{A}\lambda\mathrm{d}t = \mathbf{b} \tag{10}$$

derived from the constraint condition (3). The matrix $\mathbf{A}$ depends only on $\mathbf{x}$ permitting the implicit function formulation

$$\begin{aligned} \mathrm{d}\mathbf{z}_i = {} & \gamma\mathrm{e}^{\gamma t}\mathbf{u}\mathrm{d}t + \sigma\mathrm{e}^{\gamma t}\mathrm{d}\mathbf{W}_i + \Big[(\Delta_{i-1}\mathbf{x})\mathbf{A}^{-1}_{i-1,j} - (\Delta_i\mathbf{x})\mathbf{A}^{-1}_{i,j}\Big] \\ & \times \mathrm{e}^{\gamma t}\big[\mathrm{e}^{-2\gamma t}(\Delta_j\mathbf{z})\cdot(\Delta_j\mathbf{z})\mathrm{d}t + \gamma(\Delta_j\mathbf{x})\cdot(\Delta_j\mathbf{u})\mathrm{d}t \\ & - \gamma(\Delta_j\mathbf{x})\cdot(\Delta_j\mathbf{u})\mathrm{d}t + \sigma(\Delta_j\mathbf{x})\cdot(\Delta_j\mathrm{d}\mathbf{W})\big], \end{aligned} \tag{11}$$

or, in integral form,

$$\begin{aligned} \mathbf{z}_i(h) = {} & \mathbf{z}_i(0) + \gamma\int_0^h \mathrm{e}^{\gamma s}\mathbf{u}\mathrm{d}s + \sigma\int_0^h \mathrm{e}^{\gamma s}\mathrm{d}\mathbf{W}_i \\ & + \int_0^h \Big[(\Delta_{i-1}\mathbf{x})\mathbf{A}^{-1}_{i-1,j} - (\Delta_i\mathbf{x})\mathbf{A}^{-1}_{i,j}\Big] \\ & \times \mathrm{e}^{\gamma s}\big[\mathrm{e}^{-2\gamma s}(\Delta_j\mathbf{z})\cdot(\Delta_j\mathbf{z})\mathrm{d}s + \gamma(\Delta_j\mathbf{x})\cdot(\Delta_j\mathbf{u})\mathrm{d}s \\ & - \gamma(\Delta_j\mathbf{x})\cdot(\Delta_j\mathbf{u})\mathrm{d}s + \sigma(\Delta_j\mathbf{x})\cdot(\Delta_j\mathrm{d}\mathbf{W})\big]. \end{aligned} \tag{12}$$

With time step *h*, these equations are discretised after the expansion of Wagner and Platen [12],

$$
\begin{aligned}
x_{i^\alpha}(h) = {} & x_{i^\alpha}(0) + v_{i^\alpha}(0)\frac{1 - \mathrm{e}^{-\gamma h}}{\gamma} + \int_0^{h\dagger} (\mathcal{L}' f^x_{i^\alpha})(s_0)\mathrm{d}s_0 \\
& + \frac{1}{\gamma}\left(I_0^0(j^\beta) - \mathrm{e}^{-\gamma h} I_0^+(j^\beta)\right) g^z_{i^\alpha j^\beta}(0),
\end{aligned} \tag{13}
$$

$$
\begin{aligned}
v_{i^\alpha}(h) = {} & \mathrm{e}^{-\gamma h} v_{i^\alpha}(0) + \mathrm{e}^{-\gamma h}\int_0^{h\dagger} f^z_{i^\alpha}(s_0)\mathrm{d}s_0 \\
& + \mathrm{e}^{-\gamma h} I_0^+(j^\beta) g^z_{i^\alpha j^\beta}(0) + \mathrm{e}^{-\gamma h}\int_0^{h\dagger} (\mathcal{L}' f^z_{i^\alpha})(s_0)\mathrm{d}s_0 \\
& + \frac{\mathrm{e}^{-\gamma h}}{\gamma}\left(I_0^+(j^\beta) - I_0^0(j^\beta)\right)(\mathcal{L}' g^z_{i^\alpha j^\beta})(0) \\
& + \mathrm{e}^{-\gamma h}\Big\{\Big[(\Delta_{i-1} x_\alpha(0))\mathbf{A}^{-1}_{i-1,k} - (\Delta_i x_\alpha(0))\mathbf{A}^{-1}_{i,k}\Big] \\
& \times \Big[2(\Delta_k v_\omega(0))\cdot(\Delta_k \mathbf{g}^z_{k^\omega j^\beta}(0)) I_{00}^{+-}(0, j^\beta) \\
& - \gamma(\Delta_k x_\omega(0))\cdot(\Delta_k \mathbf{g}^z_{k^\omega j^\beta}(0)) I_{00}^{+0}(0, j^\beta)\Big]\Big\} \\
& + \frac{\mathrm{e}^{-\gamma h}}{\gamma}\left(I_{00}^{0+}(j^\beta, k^\omega) - I_{00}^{+0}(j^\beta, k^\omega)\right)(\mathcal{G}_{k^\omega}\mathcal{L}' g^z_{i^\alpha j^\beta})(0) \\
& + \frac{\mathrm{e}^{-\gamma h}}{\gamma}\left[I_{00}^{+0}(j^\beta, k^\omega) - \mathrm{e}^{-\gamma h} I_{00}^{++}(j^\beta, k^\omega)\right] \\
& \times (\mathcal{G}_{k^\omega}\mathcal{G}_{j^\beta} f^z_{i^\alpha})(0),
\end{aligned} \tag{14}
$$

with

$$
\begin{aligned}
\mathcal{L} = {} & \frac{\partial}{\partial t} + \sum_{i=1}^{m} \mathbf{f}_i \cdot \frac{\partial}{\partial \mathbf{y}_i} \\
& + \frac{1}{2}\sum_{i,j,k=1}^{m}\ \sum_{\alpha,\beta,\omega=1}^{D} g_{i^\alpha k^\omega} g_{j^\beta k^\omega} \frac{\partial^2}{\partial y_{i^\alpha}\partial y_{j^\beta}},
\end{aligned} \tag{15a}
$$

$$
\mathcal{G}_{j^\alpha} = \sum_{i=1}^{m}\sum_{\beta=1}^{D} g_{i^\beta j^\alpha} \frac{\partial}{\partial y_{i^\beta}}. \tag{15b}
$$

$$
f^x_{i^\alpha} = \mathrm{e}^{-\gamma t} z_{i^\alpha}, \tag{15c}
$$

$$
g^x_{i^\alpha j^\beta} = 0, \tag{15d}
$$

$$
\begin{aligned}
f^z_{i^\alpha} = {} & \gamma \mathrm{e}^{\gamma t} u_{i^\alpha} + \sum_{j=1}^{N-1}\Big[(\Delta_{i-1}x_\alpha)\mathbf{A}^{-1}_{i-1,j} - (\Delta_i x_\alpha)\mathbf{A}^{-1}_{ij}\Big] \\
& \times \left[\mathrm{e}^{-\gamma t}(\Delta_j \mathbf{z})\cdot(\Delta_j \mathbf{z}) + \gamma \mathrm{e}^{\gamma t}(\Delta_j \mathbf{x})\cdot(\Delta_j \mathbf{u}) - \gamma(\Delta_j \mathbf{x})\cdot(\Delta_j \mathbf{z})\right],
\end{aligned} \tag{15e}
$$

$$
\begin{aligned}
g^z_{i^\alpha j^\beta} = {} & \sigma \mathrm{e}^{\gamma t}\Big[\delta_{ij}\delta_{\alpha\beta} + (\Delta_{i-1}x_\alpha)\Big[\mathbf{A}^{-1}_{i-1,j-1}(\Delta_{j-1}x_\beta) \\
& - \mathbf{A}^{-1}_{i-1,j}(\Delta_j x_\beta)\Big] - (\Delta_i x_\alpha)\Big[\mathbf{A}^{-1}_{i,j-1}(\Delta_{j-1}x_\beta) - \mathbf{A}^{-1}_{i,j}(\Delta_j x_\beta)\Big]\Big].
\end{aligned} \tag{15f}
$$

Arabic subscripts denote the index of a bead in a polymer chain, from 1 to *M*; and Greek superscripts on the bead indices denote directional components, from 1 to *D*, the spatial dimension. Itô integrals are given by

$$
I_0^0(i^\alpha) = W_{i^\alpha}(t), \tag{16a}
$$

$$
I_0^+(i^\alpha) = \int_0^t \mathrm{e}^{\gamma s_0}\, \mathrm{d}W_{i^\alpha}(s_0), \tag{16b}
$$

$$
I_1^+(i^\alpha) = \int_0^t s_0 \mathrm{e}^{\gamma s_0}\, \mathrm{d}W_{i^\alpha}(s_0), \tag{16c}
$$

$$
I_{00}^{+0}(i^\alpha, j^\beta) = \int_0^t\int_0^{s_0} \mathrm{e}^{\gamma s_1}\, \mathrm{d}W_{j^\beta}(s_1)\mathrm{d}W_{i^\alpha}(s_0), \tag{16d}
$$

$$
I_{00}^{0+}(i^\alpha, j^\beta) = \int_0^t \mathrm{e}^{\gamma s_0}\int_0^{s_0} \mathrm{d}W_{j^\beta}(s_1)\mathrm{d}W_{i^\alpha}(s_0), \tag{16e}
$$

and the three integrals $\int^\dagger$ are to be interpreted in the sense that the integrands are to be expanded as a collection of terms multiplying exponentials $\exp(n\gamma s_0)$, for some integer *n*. The coefficients of these exponentials are evaluated at $s_0 = 0$, so the integration really applies only to these exponential factors.

Each step of the stochastic ordinary differential equation (ODE) integration involves evaluating (13) and (14), then projecting the resulting parameters $\{\mathbf{x}, \mathbf{v}\}$ to obey the constraints $\|\Delta_i \mathbf{x}\| = a$ and $(\Delta_i \mathbf{x})\cdot(\Delta_i)\mathbf{v} = 0$.

By itself, assuming a prescribed fluid motion **u**, this stochastic Langevin algorithm is second-order accurate in the weak and strong senses [6].

3.2 *Fluid solver*

3.2.1 Corrector

If we consider a two-dimensional (2D) cell-centred velocity, $\mathbf{u}_{i,j} = (u, v)_{i,j}$, it would be ideal (for second-order temporal accuracy) to advance the solution to times $(n + 1)\Delta t$ as follows:

$$
\begin{aligned}
\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\bigg(& -[(\mathbf{u}\cdot\nabla)\mathbf{u}]^{n+(1/2)} - \frac{1}{\rho}\nabla p^{n+(1/2)} \\
& + \nu\Delta \mathbf{u}^{n+(1/2)} + \frac{1}{\rho}\mathbf{f}^{n+(1/2)}\bigg).
\end{aligned} \tag{17}
$$

The implicit nature of the velocity and time centring of the pressure gradient are issues in this discretisation. We treat the viscous source term implicitly and estimate an intermediate velocity that is not necessarily divergence free based on a lagged pressure gradient. This temporal integration is performed using a second-order Runge–Kutta method [13] that is known to be stable in L_0 for our finite volume approach to geometry. For parabolic partial differential equations (PDEs)

$$\frac{\partial \varphi}{\partial t} = \mathcal{L}\varphi + \rho(\varphi), \tag{18}$$

it takes the form

$$(\mathbf{I} - \mu_1\mathcal{L})(\mathbf{I} - \mu_2\mathcal{L})\varphi^{n+1} = (\mathbf{I} + \mu_3\mathcal{L})\varphi^n + (\mathbf{I} + \mu_4\mathcal{L})\rho(\varphi^{n+(1/2)}) \tag{19}$$

with $\mu_1 = \mu_2 = (1 - 1/\sqrt{2})\Delta t$, $\mu_3 = (\sqrt{2} - 1)\Delta t$ and $\mu_4 = (\sqrt{2} - 3/2)\Delta t$. This leads to the following discretisation for the momentum equation:

$$\big(\mathbf{I} - \mu_1\nu\Delta\big)\big(\mathbf{I} - \mu_2\nu\Delta\big)\mathbf{u}^* = \big(\mathbf{I} + \mu_3\nu\Delta\big)\mathbf{u}^n + \big(\mathbf{I} + \mu_4\nu\Delta\big) \times \left(-[(\mathbf{u}\cdot\nabla)\mathbf{u}]^{n+(1/2)} - \frac{1}{\rho}\nabla p^{n-(1/2)} + \frac{1}{\rho}\mathbf{f}^{n+(1/2)}\right), \tag{20}$$

where $\mathcal{L} = \nu\Delta$. A projection method [14] is then used to advance the velocity and pressure gradient in pressure formulation form (as opposed to pressure correction) while enforcing incompressibility

$$\mathbf{u}^{n+1} = \mathbf{P}\left(\mathbf{u}^* + \frac{\Delta t}{\rho}\nabla p^{n-(1/2)}\right), \tag{21}$$

$$\nabla p^{n+(1/2)} = \frac{\rho}{\Delta t}\mathbf{Q}\left(\mathbf{u}^* + \frac{\Delta t}{\rho}\nabla p^{n-(1/2)}\right), \tag{22}$$

where

$$\mathbf{Q} = \mathbf{G}\mathbf{L}^{-1}\mathbf{D}, \tag{23a}$$

$$\mathbf{P} = \mathbf{I} - \mathbf{Q}. \tag{23b}$$

D, **G** and **L** are the discrete divergence operator, discrete gradient operator and discrete Laplacian, respectively. In our approach, **P** is an approximate projection operator and $\mathbf{L} \neq \mathbf{DG}$ [15].

3.2.2 *Velocity predictor*

The heat equation (20) contains as a source term the nonlinear convective derivative $\mathbf{u}\cdot\nabla\mathbf{u}^{n+(1/2)}$. To approximate this term, we follow finite volume discretisations of hyperbolic systems based on high-order Godunov methods for advection [16]. We expand the cell-centred velocity to a cell edge and to the half step in time in a Taylor series, for example, in the x-direction,

$$\mathbf{u}^{n+(1/2)}_{i+(1/2),j} = \left(\mathbf{u} + \frac{\Delta x}{2}\frac{\partial \mathbf{u}}{\partial x} + \frac{\Delta t}{2}\frac{\partial \mathbf{u}}{\partial t}\right)^n_{i,j}. \tag{24}$$

Using the PDE to substitute the time derivative, we extrapolate from cell centres to cell edges in the plus and minus directions, omitting the pressure gradient:

$$\mathbf{u}^{x\pm}_{i,j} = \mathbf{u}^n_{i,j} \pm \frac{1}{2}\min\left(1 \mp u_{ij}\frac{\Delta t}{2\Delta x}, 1\right)\delta_x\mathbf{u}^n_{i,j} + \frac{\Delta t}{2}\left(\nu\Delta\mathbf{u}^n_{i,j} + \frac{1}{\rho}\mathbf{f}^n_{i,j}\right). \tag{25}$$

The normal slopes are calculated as follows, e.g.

$$(\delta_x u)^n_{i,j} = \begin{cases} (\delta_x u)^{vL} & \text{if } (u^n_{i+1,j} - u^n_{i,j})(u^n_{i,j} - u^n_{i-1,j}) > 0, \\ 0 & \text{if } (u^n_{i+1,j} - u^n_{i,j})(u^n_{i,j} - u^n_{i-1,j}) \leq 0, \end{cases} \tag{26}$$

where [17]

$$(\delta_x u)^{vL} = \mathrm{sign}\left(\frac{u^n_{i+1,j} - u^n_{i-1,j}}{2}\right) \times \min\left(2\left|u^n_{i,j} - u^n_{i-1,j}\right|, 2\left|u^n_{i+1,j} - u^n_{i,j}\right|, \frac{1}{2}\left|u^n_{i+1,j} - u^n_{i-1,j}\right|\right), \tag{27}$$

with one-sided differences at boundaries. We can now solve a Riemann problem at each cell edge, where there exist two states, having extrapolated from cell centres on both sides of a cell edge, to obtain the state at the edge. For the incompressible Navier–Stokes equations, the Riemann problem solution is simple upwinding:

$$\hat{\mathbf{u}}^{n+(1/2)}_{i+(1/2),j} = \begin{cases} \mathbf{u}^{x,+}_{i,j} & \text{if } u^n_{i+(1/2),j} > 0, \\ \mathbf{u}^{x,-}_{i+1,j} & \text{if } u^n_{i+(1/2),j} < 0, \\ \frac{1}{2}\left(\mathbf{u}^{x,+}_{i,j} + \mathbf{u}^{x,-}_{i+1,j}\right) & \text{if } u^n_{i+(1/2),j} = 0, \end{cases} \tag{28}$$

where

$$u^n_{i+(1/2),j} = \frac{1}{2}\left(u^n_{i,j} + u^n_{i+1,j}\right). \tag{29}$$

The solution to the Riemann problem is used as the flux for the transverse correction of the plus and minus states:

$$\mathbf{u}^{x\pm}_{i,j} := \mathbf{u}^{x\pm}_{i,j} - \frac{\Delta t}{2\Delta y}\left(\hat{\mathbf{u}}^{n+(1/2)}_{i,j+(1/2)} - \hat{\mathbf{u}}^{n+(1/2)}_{i,j-(1/2)}\right). \tag{30}$$

The Riemann problem (28) is then solved again based on plus and minus states that include the transverse flux difference.

An intermediate projection is applied to make up for the omitted pressure gradient in the extrapolation (25) in order to be consistent with the constraint $\nabla\cdot\mathbf{u} = 0$:

$$u^{n+(1/2)}_{i+(1/2),j} = \hat{u}^{n+(1/2)}_{i+(1/2),j} - \left[\mathbf{Q}^{\mathrm{MAC}}(\hat{\mathbf{u}})_x\right]^{n+(1/2)}_{i+(1/2),j}. \tag{31}$$

Here, MAC denotes the marker-and-cell centring. The subscript x indicates the direction of the gradient to correspond to the x-component of velocity, for example.

The final step in the predictor is a repeat of the Godunov extrapolation with two exceptions: the velocity used to solve the Riemann problems is the projected normal velocity at a cell edge, e.g. $u^{n+(1/2)}_{i+(1/2),j}$, not $u^{n}_{i+(1/2),j}$; and the intermediate projection is not performed, but, rather, the previous solution to the Poisson's equation is used, since the normal velocities comprising the right-hand side have not changed. Thus, this step only affects the transverse velocities at cell edges, which are corrected with transverse gradients obtained from nearest neighbour normal gradients from the previous projection according to

$$v^{n+(1/2)}_{i+(1/2),j} = \hat{v}^{n+(1/2)}_{i+(1/2),j} - \left[\mathbf{Q}^{\mathrm{MAC}}(\hat{\mathbf{u}})_y\right]^{n+(1/2)}_{i+(1/2),j}, \tag{32}$$

resulting in a second-order estimate of $\mathbf{u}^{n+(1/2)}_{i+(1/2),j}$, which can be used to construct the convective derivative.

3.3 Coupling strategy

To couple the second-order strong polymer solver with a solver for the Navier–Stokes equations, while retaining second-order accuracy in the strong sense, we use a predictor–corrector strategy. The algorithm is split as follows.

3.3.1 Step 1: Fluid predictor

To advance the fluid solution with second-order accuracy, the force must be centred at $t^{n+(1/2)}$ and accurate to second order. With particle positions initially known only at t^n, this accuracy cannot be achieved. Instead, we use $\mathbf{f}^n$ (7) – the lower-order t^n centring. With this source, discrete divergence-free edge-centred – and time-centred – estimates of the fluid velocity, $\mathbf{u}^{n+(1/2)}_{i+(1/2),j}$, are computed with the predictor component of the Navier–Stokes method (see Section 3.2.2).

3.3.2 Step 2: Polymer evaluation at time $t + (\Delta t/2)$

We average $\mathbf{u}^{n+1/2}_{i+1/2,j}$ to cell centres, e.g. in 2D,

$$\mathbf{u}^{n+(1/2)}_{ij} = \frac{1}{4}\left(\mathbf{u}^{n+(1/2)}_{i-(1/2),j} + \mathbf{u}^{n+(1/2)}_{i+(1/2),j} + \mathbf{u}^{n+(1/2)}_{i,j-(1/2)} + \mathbf{u}^{n+(1/2)}_{i,j+(1/2)}\right), \tag{33}$$

then use a second-order particle-in-cell method to approximate $\mathbf{u}^{n+(1/2)}(\mathbf{x}^n_i)$, the fluid velocity at the $t^{n+(1/2)}$ predicted location of each particle. Using this, the material time derivative $\dot{\mathbf{u}}^n(\mathbf{x}^n_i)$ is estimated by

$$\dot{\mathbf{u}}^n(\mathbf{x}^n_i) = 2\cdot\frac{\mathbf{u}^{n+(1/2)}(\mathbf{x}^n_i) - \mathbf{u}^n(\mathbf{x}^n_i)}{\Delta t} + (\mathbf{v}^n_i\cdot\nabla)\mathbf{u}(\mathbf{x}^n_i). \tag{34}$$

Using the estimate of the fluid velocities and its derivatives at time t, with the initial condition $\{\mathbf{x}^n_i, \mathbf{v}^n_i\}$, we can solve the stochastic ODEs for the constrained particle motion over the interval $[t^n, t^{n+(1/2)}]$. This provides $\{\mathbf{x}^{n+(1/2)}_i, \mathbf{v}^{n+(1/2)}_i\}$, which permits the second-order evaluation of $\mathbf{f}^{n+(1/2)}$ from (7).

3.3.3 Step 3: Fluid corrector

Using the second-order estimate to the flux $\mathbf{f}^{n+(1/2)}$ known to second order, the corrector steps of the fluid solver are carried out. This consists of the heat solver (20) and final projection (21). The fluid velocity $\mathbf{u}^{n+1}$ is now updated to the new time and is second-order accurate.

3.3.4 Step 4: Polymer evaluation at time $t + \Delta t$

In this final step, we advance the polymer equations from $t^{n+(1/2)}$ to t^{n+1}, now using velocity $\mathbf{u}^{n+1}(\mathbf{x}^{n+(1/2)}_i)$ to compute derivative $\dot{\mathbf{u}}^{n+(1/2)}(\mathbf{x}^{n+(1/2)}_i)$ in a manner analogous to Step 2. The updated particle parameters $\{\mathbf{x}^{n+1}, \mathbf{v}^{n+1}\}$ may then be used to evaluate $\mathbf{f}^{n+1}$ to second order.

4. Results

We apply this hybrid fluid-particle algorithm to the simulation of polymers that are represented by a chain of beads and rods coupled to an incompressible viscous solvent. We perform 2D simulations of two polymers, 150 and 80 nodes, respectively, flowing in an abrupt contraction microchannel at three different Reynolds numbers. This geometry is chosen so as to simulate the transport of a polymer like a DNA molecule into a smaller, confined channel where a sensor might be located. The channel section is 10 μm long and the inlet is 3.75 μm wide. The contraction ratio is 4:1. We choose the interparticle spacing, or constraint length, to be the Kuhn length for DNA, $a = 100$ nm, which is a measure of the flexibility in the polymer chain; other parameters are the particle mass, $m =$ 1e-19 g, relaxation time, $\gamma =$ 1e + 12 s and $\sigma =$ 5e + 08 cm/s$^{3/2}$ (see [2,11] for definitions).

In Figure 1, the Reynolds number at the inlet of the channel is 0.000375. The polymers are in an initially coiled state. The convective forces are not strong enough to stretch them out as they proceed through the abrupt contraction where the flow accelerates, which is a typical behaviour in experimental DNA flows [8]. The low resolution of these simulations is such that the viscous forces, which dominate inertial forces at this very low

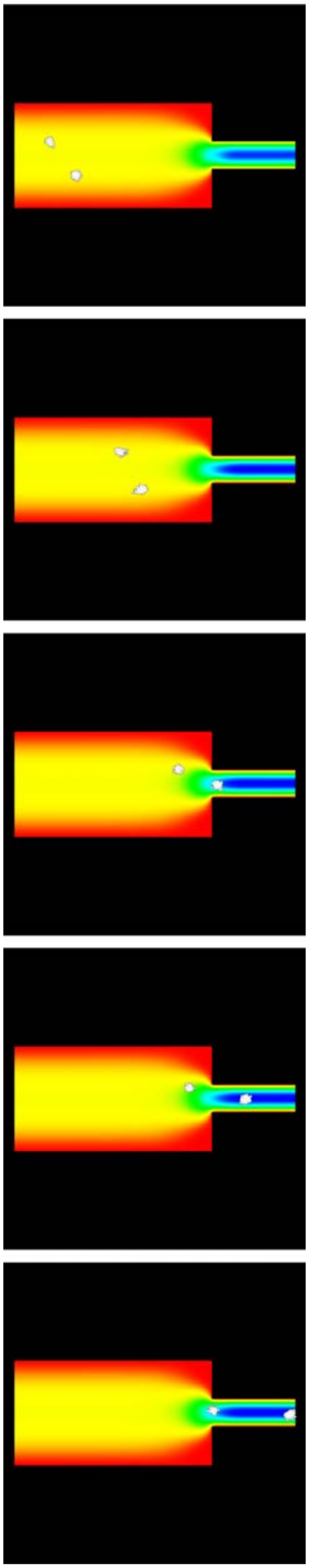

Figure 1. Polymer locations at times 0.00199, 0.01999, 0.03499, 0.03699 and 0.03989 s for $Re = 0.000375$. Background colour is velocity: blue (high) and red (low).

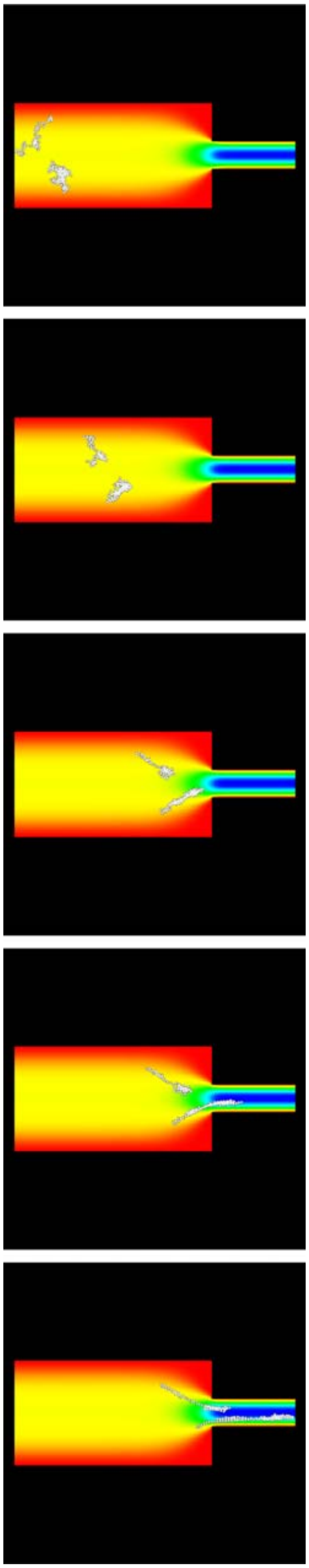

Figure 2. Polymer locations at times 3.3e-06, 0.00017, 0.00034, 0.00038 and 0.00042 s for $Re = 0.0375$. Background colour is velocity: blue (high) and red (low).

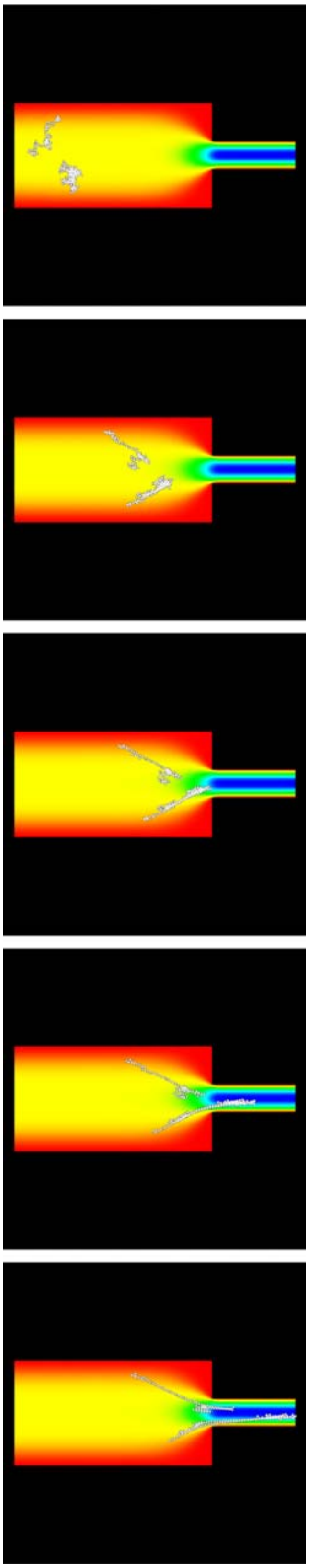

Figure 3. Polymer locations at times 3.4e-07, 2.7e-06, 3.4e-06, 3.8e-06, 4.0e-06 s for $Re = 3.75$. Background colour is velocity: blue (high) and red (low).

Reynolds number, are not resolved, and, therefore, do not manipulate the polymer, either. This flow scenario is likely not ideal for single-molecule detection to occur in the smaller contracted channel.

In Figure 2, the Reynolds number is 0.0375. The stronger inertial flow along with velocity gradients due to viscous forces uncoils the polymers and stretches them out in the accelerated region as if threading the polymers through the contracted channel. These dynamics seem to be ideal for the design of a sensor.

In Figure 3, the Reynolds number is 3.75. The polymer transport is faster and the polymers are stretched out. These could be acceptable parameters for sensor design in this channel, but the flow might be too fast for capture and detection of the molecule to occur. This finding is consistent with detection systems that rely on DNA capture techniques before amplification to ensure a signal [9]. In such systems, the typical operating Reynolds number is on the order of 1, so that individual DNA molecules are not transported too rapidly to avoid capture.

A careful numerical convergence study of this coupled system is underway to verify the accuracy claims that are made in this paper on the basis of theoretical analysis.

Acknowledgements

B. Kallemov was supported by the Bolashak scholarship of the President of the Republic of Kazakhstan. G.H. Miller was supported by DOE contract number DE-SC0001981 and NSF grant number DMS-0810939. D. Trebotich was supported by the US Department of Energys Office of Advanced Scientific Computing Research under contract number DE-AC02-05-CH11231.

References

[1] D. Trebotich, G.H. Miller, and P. Colella, *A stable and convergent scheme for viscoelastic flow in contraction channels*, J. Comput. Phys. 205 (2005), pp. 315–342.

[2] D. Trebotich and G.H. Miller, *A tightly coupled particle-fluid model for DNA-laden flows in complex microscale geometries*, in *Computational Fluid and Solid Mechanics*, K.J. Bathe, ed., Elsevier, Amsterdam, 2005, pp. 1018–1022.

[3] G.H. Miller and D. Trebotich, *Toward a mesoscale model for the dynamics of polymer solutions*, J. Comput. Theoret. Nanosci. 4 (2007), pp. 797–801.

[4] D. Trebotich, G.H. Miller, and M.D. Bybee, *A penalty method to model particle interactions in DNA-laden flows*, J. Nanosci. Nanotechnol. 8 (2008), pp. 3749–3756.

[5] B. Kallemov, G.H. Miller, and D. Trebotich, *A Duhamel approach for the Langevin equations with holonomic constraints*, Mol. Simul. 35(6) (2009), pp. 440–447.

[6] B. Kallemov and G.H. Miller, *A second-order strong method for the Langevin equations with holonomic constraints*, SIAM J. Sci. Comput. (2010), submitted.

[7] D. Trebotich, B. Van Straalen, D.T. Graves, and P. Colella, *Performance of embedded boundary methods for CFD with complex geometry*, J. Phys.: Conf. Ser. 125 (2008), 012083.

[8] P. Shrewsbury, S.J. Muller, and D. Liepmann, *Concentration effects of a biopolymer in a microfluidic device*, Biomed. Microdevices 3 (2001), pp. 225–238.

[9] F.P. Milanovich, J. Dzenitis, B.J. Hindson, A.J. Makarewicz, M.T. McBride, and B.W. Colston, *APDS, A network-ready, broad-spectrum, environmental pathogen detection system*, in *Defense against Bioterror: Detection Technologies, Implementation Strategies and Commercial Opportunities*, D. Morrison, ed., Springer, Berlin, Vol. 1, 2006, pp. 67–75.

[10] P. Langevin, *Sur la thèorie du mouvement Brownien*, C.R. Hebdomadaires des Séances de l'Académie des Sciences 146 (1908), pp. 530–533.

[11] B. Kallemov, G.H. Miller, and D. Trebotich, *Numerical simulation of polymer flow in microfluidic devices*, 2009 Proceedings of the 4th SIAM Conference on Mathematics for Industry (MI09) (2010), pp. 93–98.

[12] W. Wagner and E. Platen, *Approximation of Itô integral equations*, Preprint ZIMM, Akad. Wissenschaften, DDR, Berlin (1978).

[13] E.H. Twizell, A.B. Gumel, and M.A. Arigu, *Second-order, L_0-stable methods for the heat equation with time-dependent boundary conditions*, Adv. Comput. Math. 6 (1996), pp. 333–352.

[14] J.B. Bell, P. Colella, and H.M. Glaz, *A second-order projection method for the incompressible Navier-Stokes equations*, J. Comput. Phys. 85 (1989), pp. 257–283.

[15] M.F. Lai, *A projection method for reacting flow in the zero Mach number limit*, Ph.D. thesis, University of California, Berkeley, 1994.

[16] P. Colella, *Multidimensional upwind methods for hyperbolic conservation laws*, J. Comput. Phys. 87 (1990), pp. 171–200.

[17] B. van Leer, *Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method*, J. Comput. Phys. 32 (1979), pp. 101–136.