Toward a Mesoscale Model for the Dynamics of Polymer Solutions

G. H. Miller[1] and D. Trebotich[2, *]

[1]*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*
[2]*Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-560, Livermore, CA 94551, USA*

To model entire microfluidic systems containing solvated polymers we argue that it is necessary to have a numerical stability constraint governed only by the advective CFL condition. Advancements in the treatment of Kramers bead-rod polymer models are presented to enable tightly-coupled fluid-particle algorithms in the context of system-level modeling.

Keywords: Polymer Flow, Microfluidics, Bead-Rod Model, SHAKE, RATTLE.

1. INTRODUCTION

We are interested in modeling the fate of long polymer molecules in macroscopic flows with nontrivial geometry. Applications include predictive modeling of microfluidic bio-sensor and bio-processing devices which are currently being developed for medical, industrial, and defense applications. Our goal is to model such fluid systems in their entirety, which necessitates an approach in which the discretized equations are subject to a stability condition which differs little from the advective Courant-Friedrichs-Lewy one (i.e., CFL ≈ 1 with CFL $\equiv \Delta t |v_{\text{fluid}}|_\infty / \Delta x$). For typical microfluidic applications $\Delta x = \mathcal{O}(10^{-6}$ m$)$ and $|v_{\text{fluid}}|_\infty = \mathcal{O}(10^{-2}$ m/s$)$ so $\Delta t = \mathcal{O}(10^{-4}$ s$)$. This contrasts sharply with approaches such as molecular dynamics which are capable of faithfully modeling molecular-level interactions at the cost of very small time steps $\Delta t = \mathcal{O}(10^{-15}$ s$)$.

When faced with this large disparity in time scales, the usual approach is to eliminate short time scale behavior by freezing high frequency modes of the system. This approach has been employed in the molecular dynamics community for over 40 years, as in approximating the water molecule (with 9 degrees of freedom) by a symmetric rigid rotor (with 6 degrees of freedom).[12] We combine this approach with Kramers' abstraction,[11] which represents long-chain molecules as a collection of point beads connected by rigid rods—each point mass representing dozens or more atoms. This coarse-graining approach has been shown to provide models which capture many

essential mechanical features of molecules like DNA,[9] while giving up resolution of chemical intra-molecule and inter-molecule interactions. These interactions must themselves be coarse-grained if they are to be included without affecting the stable time step. In the case of molecule-solvent interactions, this is generally accomplished with the stochastic Brownian motion model. Approaches to coarse-graining the inter-polymer and intra-polymer interactions were described in[16] and will be further developed here.

Our approach to system-level modeling has been to incorporate simplified polymer models with continuum models (Navier Stokes) for the fluid. Different approaches are possible. For example, Symeonidis and Karniadakis[15] use the dissipative particle dynamics (DPD) coarse-graining approach[8] to model solvated polymers. By itself, DPD offers $\mathcal{O}(100)\times$ speed-up relative to raw molecular dynamics. Symeonidis and Karniadakis realized an additional $100\times$ speed-up by applying different time steps to solvent and polymer. This hybrid DPD approach could be extended to a system-level model through the adaptive mesh and algorithm refinement (AMAR) formalism,[6] an approach which would give molecular-scale attention to the regions of space containing polymers while applying a continuum Navier Stokes approach to the remainder of the problem domain. In complex systems containing large concentrations of polymer, however, this AMAR strategy would require molecular-scale resolution essentially everywhere, which would render the approach impractical. Thus, in our view, for system-level modeling a strategy is required which captures the essential behavior of

---

*Author to whom correspondence should be addressed.

polymer molecules and their interactions with the solvent without requiring molecular-scale treatment of the solvent. This approach could be used even with AMAR to treat some selective polymers or regions with molecular-scale fidelity while using the coarser model in the majority of the domain.

We presented a model with these general features in Ref. [17]. There, we also developed a simple elastic collision model for molecule-wall interaction to prevent polymer molecules from exiting the fluid domain. This elastic collision approach is itself a coarse-graining in the same spirit as the one used to rigidly fix Kramers segment lengths. Chemical interactions between the wall and the polymer have been ignored, which eliminates high-frequency modes to enable long stable time steps.

Another important class of interactions concerns the collision of a polymer with itself: an unbroken polymer has the topology of a string, and cannot cross itself. This non-crossing constraint occurs naturally in molecular dynamics simulations by virtue of short-range repulsive forces. The coarse-graining approach analogous to our polymer-wall interaction is to treat polymer–polymer interactions as elastic collisions. A preliminary treatment of this is described in Ref. [16] where dynamics were observed which were essentially identical to those obtained using repulsive potentials.

In this report we present our current model for intra-polymer collisions, which differs from previous ones by incorporation of a Kramers bead velocity constraint, and improvement of the polymer–polymer elastic collision model. Together these improvements have increased the time scale required for stable polymer modeling by a factor of ≈ 100, which now places the stable polymer time step on an equal footing with the fluid CFL timestep, enabling system-level modeling.

2. ALGORITHMIC APPROACH

Our ultimate goal of system-level modeling is to combine our 3D Navier Stokes continuum model with a solver for particle dynamics. Our Navier Stokes method is described briefly in Ref. [17], and will be described more fully in Ref. [2]. Complex geometry is accommodated using the so-called "embedded boundary" method,[10,5] which combines high-order Cartesian grid discretizations with cut-cell representations of the geometry. Our polymer-fluid coupling strategy is described in Ref. [17]. It tightly couples the momentum transfer between fluid and particles in a self-consistent way. High-order accuracy is achieved through a predictor-corrector approach.

Here, our polymer dynamics algorithm will be described independently of the Navier Stokes application. The exposition will begin by considering a single time step in which the polymer bead trajectories, unchecked, allow for at most one rod crossing event. Following this development the

modifications necessary to allow more complex ~~interacts~~ will be described.

For each time step n, beginning with bead coordinates x^n and velocities v^n:
(1) Calculate the unconstrained motion to obtain x^* and v^*—provisional values at time $n+1$. Our time-stepping strategy is described in Ref. [17] including stochastic terms in both coordinate and velocity equations.
(2) Correct the dynamics computed above to include rod length and velocity constraints. The corrected quantities will be labeled $x^\dagger$ and $v^\dagger$. In Ref. [17] we employed the Lagrange multiplier technique of Ref. [4] to enforce the constraint $r_{i,i+1} \cdot r_{i,i+1} = a^2$, where a is the Kramers rod length. The resulting method is similar to the **SHAKE**[13] algorithm of molecular dynamics, appropriate to the coordinate-only Verlet method. Differentiation of this constraint gives $r_{i,i+1} \cdot v_{i,i+1} = 0$, which constrains the velocity. This was introduced as the **RATTLE**[1] algorithm in the molecular dynamics literature, appropriate to the so-called velocity Verlet algorithm, and we adopt it here. Enforcing this velocity constraint resulted in a significant ($\mathcal{O}(100\times)$) improvement in the time stability of our polymer model.
(3) Correct the dynamics computed above to incorporate polymer–wall and polymer–polymer interactions. The corrected quantities will be labeled $x^\ddagger$ and $v^\ddagger$. The details of this step for polymer-polymer interactions are as follows:

Calculate $v^{\Delta t}$, the bead velocities over the current time step

$$v^{\Delta t} = (x^\dagger - x^n)/\Delta t$$

so that the time-linear trajectory of each bead over the current time step is

$$x = x^n + v^{\Delta t}t \qquad t \in (0, \Delta t) \qquad (1)$$

Repeatedly loop through all rod pairs until no more collisions are detected. For each pair of rods i and j:
(3a) Calculate the triple product V_{ij} at times 0 and Δt where

$$V_{ij} = (x_i - x_j) \cdot ((x_{i+1} - x_i) \times (x_{j+1} - x_j)) \qquad (2)$$

The value of V_{ij} will be zero if the infinite lines containing the rods intersect or are parallel. Therefore, if the value of V_{ij} changes sign over the time step, a possible rod–rod crossing has occurred. Otherwise, proceed to step 3h.
(3b) Calculate τ, the time of crossing. One could substitute the time-linear trajectories of (1) into (2) to give a third-order polynomial $V_{ij}(t)$.[3,16] The smallest root of this polynomial in $(0, \Delta t)$ is τ–a potential time of impact. We have implemented an "exact" cubic equation solver and also a Newton's method iterative solver. The exact approach is subject to large numerical errors (Ref. [14], §5.8) and often fails to compute a $\tau \in [0, \Delta t]$ even when $V_{ij}(0)V_{ij}(\Delta t) < 0$. The Newton solver is somewhat more costly, but robust. Even simple bisection is preferred to the analytical calculation.

(3c) Calculate $\boldsymbol{x}^\tau$, the bead positions at time τ:

$$\boldsymbol{x}^\tau = \boldsymbol{x}^n + \boldsymbol{v}^{\Delta t}\tau$$

(3d) Calculate $\boldsymbol{\chi}$, the point of intersection at time τ, by solving the following set of equations:

$$\boldsymbol{\chi} = \boldsymbol{x}_i^\tau + \lambda_i(\boldsymbol{x}_{i+1}^\tau - \boldsymbol{x}_i^\tau) = \boldsymbol{x}_j^\tau + \lambda_j(\boldsymbol{x}_{j+1}^\tau - \boldsymbol{x}_j^\tau)$$

where λ_i and λ_j define the point of intersection between the lines containing rods i and j respectively. If $(0 \le \lambda_i \le 1)$ and $(0 \le \lambda_j \le 1)$ then the point of intersection lies on both rods and a rod–rod crossing has occurred. Otherwise, proceed to step 3h

(3e) Calculate $\mathbf{n}$, the unit vector normal to the plane formed by the two rods at time τ

$$\mathbf{n} = \frac{(\boldsymbol{x}_{i+1}^\tau - \boldsymbol{x}_i^\tau) \times (\boldsymbol{x}_{j+1}^\tau - \boldsymbol{x}_j^\tau)}{|(\boldsymbol{x}_{i+1}^\tau - \boldsymbol{x}_i^\tau) \times (\boldsymbol{x}_{j+1}^\tau - \boldsymbol{x}_j^\tau)|}$$

(3f) Compute the change in velocity at time τ that beads i, $i+1$, j, and $j+1$ would experience in the perfectly elastic collision of two "dumbbells." This correction applies only to the velocity components in direction $\mathbf{n}$. Let $v_i = \boldsymbol{v}^{\Delta t} \cdot \mathbf{n}$ prior to collision, and let $\hat{v}_i$ be the corresponding quantity after the collision. If J is the specific impulse (impulse per bead mass) applied to rod i, $i+1$ in direction $\mathbf{n}$, assuming all beads have equal mass. By conservation of linear momentum, $-J$ is the specific impulse applied to rod j, $j+1$:

$$J + v_i + v_{i+1} = \hat{v}_i + \hat{v}_{i+1}$$
$$-J + v_j + v_{j+1} = \hat{v}_j + \hat{v}_{j+1}$$

Conservation of angular momentum is expressed by

$$Ja\left(\frac{1}{2} - \lambda_i\right) + \frac{a^2}{2}\frac{(v_i - v_{i+1})}{a} = \frac{a^2}{2}\frac{(\hat{v}_i - \hat{v}_{i+1})}{a}$$
$$-Ja\left(\frac{1}{2} - \lambda_j\right) + \frac{a^2}{2}\frac{(v_j - v_{j+1})}{a} = \frac{a^2}{2}\frac{(\hat{v}_j - \hat{v}_{j+1})}{a}$$

where $\pm Ja(1/2 - \lambda)$ is the specific angular impulse. Finally, kinetic energy is conserved in a perfectly elastic collision,

$$v_i^2 + v_{i+1}^2 + v_j^2 + v_{j+1}^2 = \hat{v}_i^2 + \hat{v}_{i+1}^2 + \hat{v}_j^2 + \hat{v}_{j+1}^2$$

The solution to these equations is

$$J = \frac{-v_i(1-\lambda_i) - v_{i+1}\lambda_i + v_j(1-\lambda_j) + v_{j+1}\lambda_j}{1 - \lambda_i(1-\lambda_i) - \lambda_j(1-\lambda_j)}$$
$$\boldsymbol{v}_i^{\mathrm{col}} = \boldsymbol{v}_i^{\Delta t} + J(1-\lambda_i)\mathbf{n}$$
$$\boldsymbol{v}_{i+1}^{\mathrm{col}} = \boldsymbol{v}_{i+1}^{\Delta t} + J\lambda_i\mathbf{n}$$
$$\boldsymbol{v}_j^{\mathrm{col}} = \boldsymbol{v}_j^{\Delta t} - J(1-\lambda_j)\mathbf{n}$$
$$\boldsymbol{v}_{j+1}^{\mathrm{col}} = \boldsymbol{v}_{j+1}^{\Delta t} - J\lambda_j\mathbf{n}$$

where $\boldsymbol{v}^{\mathrm{col}}$ is the post-collision velocity.

(3g) Update positions for beads i, $i+1$, j, and $j+1$ at time Δt, and adjust the final velocity to account for elastic rebound:

$$\boldsymbol{x}^\ddagger = \boldsymbol{x}^\tau + (\Delta t - \tau)\boldsymbol{v}^{\mathrm{col}}$$
$$\boldsymbol{v}^\ddagger = \boldsymbol{v}^\dagger + (\boldsymbol{v}^{\mathrm{col}} - \boldsymbol{v}^{\Delta t})$$

(3h) If no collision occurs, then for beads i, $i+1$, j, and $j+1$

$$\boldsymbol{v}^\ddagger = \boldsymbol{v}^\dagger$$
$$\boldsymbol{x}^\ddagger = \boldsymbol{x}^\dagger$$

This 3D algorithm does not account for the vanishingly-unlikely possibility of two co-planar rods colliding in their initial plane.

Step 3f assumes the 4 participating beads exist as two independent, rigid, dumbbells. Of course this is not the case, in general, since the participating beads may each be linked to another rod. The dumbbell approximation used to calculate the the elastic collision is therefore somewhat arbitrary. However, it at least has the merit of correctly computing the result for two colliding Kramers dimers. The bead-wall constraint described in Ref. [17] is similarly approximated.

We note that the constraints are enforced hierarchically: if the rod–rod crossing constraint calls for the correction of a bead's motion, then the rods connected to that bead will no longer have the correct length, but the error is only order Δt^2 and persists only for the affected time step. The velocity constraint is similarly affected.

This hierarchical approach is employed because the simultaneous enforcement of the nonlinear length constraint is not easily combined with the elastic impact treatment of the crossing constraint. With respect to the rod-length constrain, the same is true of the polymer–wall constraint which says all beads must remain in the fluid domain. In Ref. [17] that constraint was enforced by elastically rebounding balls at their point of collision with a wall. In complex geometries we represent the wall with the zero value of a distance function level set. The discrete level set is interpolated on the linearized trajectory of each bead over the time step Δt using (1).

The bead–wall and rod–rod collisions may be handled simultaneously, however, with no hierarchical precedence and in a manner which permits multiple such collisions in a single time step. Briefly, the algorithm given above is extended in the following way: for each time step,

(1) Initialize the reference time $t_0 = 0$. Compute provisional velocities $\boldsymbol{v}^\dagger$ and provisional coordinates $\boldsymbol{x}^\dagger$ at Δt, and compute the average velocity $\boldsymbol{v}^{\Delta t}$. Initialize $\boldsymbol{x}^* = \boldsymbol{x}^n$ and $\boldsymbol{v}^* = \boldsymbol{v}^{\Delta t}$.

(2) While $t_0 < \Delta t$:

(2a) Find the first time $\tau \in [t_0, \Delta t]$ at which a rod–rod crossing or bead–wall crossing is predicted to occur. If

no such collision is found, let $x^* := x^* + (\Delta t - t_0)v^{\Delta t}$, and break to 3

(2b) For the predicted impact, compute the crossing coordinates x^τ and post-impact velocities v^{col}.

(2c) For all beads involved in the collision, update coordinates with

$$x^* = x^\tau$$
$$v^* = v^{\text{col}}$$
$$v^\dagger := v^\dagger + (v^{\text{col}} - \mathbf{v}^{\Delta t})$$
$$v^{\Delta t} := v^{\text{col}}$$

For all beads not involved in the collision let

$$x^* := x^* + (\tau - t_0)v^{\Delta t}$$

and let $t_0 = \tau$.

(2d) Loop to 2

(3) Continue to next time step with

$$x^{n+1} = x^*$$
$$v^{n+1} = v^\dagger$$

This multi-collision algorithm has the property that each sub-step starts with coordinates in impact position, but with corrected velocities. Some care is required to implement impact detection and crossing detections for subsequent sub-steps because, for example, $V_{ij}(t_0) = 0$ if the previous sub-step found a rod–rod crossing. Therefore, the criterion $V_{ij}(\Delta t)V_{ij}(t_0) < 0$ will not be useful. A heuristic fix to this problem is to "nudge" coordinates away from the calculated impact point by some amount $|\epsilon|v^{\text{col}}$ where ϵ is small but sufficiently large that $V_{ij} \neq 0$ and with the correct sign. We have had good success with this approach, though it can fail as will be described in the next section.

In developing these algorithms we discovered some interesting subtleties to the rod-length constraint methods of Refs. [4, 13]. One way to express that method is to say that one computes a set of multipliers λ such that

$$\tilde{x}_i := x_i + \lambda_{i-1}(r_i^c - r_{i-1}^c) - \lambda_i(r_{i+1}^c - r_i^c)$$

with $\tilde{x}$ a coordinate satisfying $|\tilde{x}_{i+1} - \tilde{x}_i| = a$. The coordinates r^c define the directions in which the constraining adjustments may be made. In Refs. [4, 13] $r^c = x^n$, the vectors defined by the bead positions at the start of the time step. However, in Ref. [17] we found improved stability using coordinates at the end of the time step. Whatever the choice, the resulting system of equations approximates a tridiagonal linear system $A\lambda = b$ (b has constants and terms quadratic in λ which are assumed negligible for the purposes of analysis). One can analyze the iterative properties of the $(N-1) \times (N-1)$ matrix A, with N the number of Kramers beads, whose elements may be written

$$A_{i,i-1} = -2(r_{i-1}^c - r_i^c) \cdot (x_i - x_{i+1}), \quad i > 1$$
$$A_{i,i} = 4(r_i^c - r_{i+1}^c) \cdot (x_i - x_{i+1})$$
$$A_{i,i+1} = -2(r_{i+1}^c - r_{i+2}^c) \cdot (x_i - x_{i+1}), \quad i < N-1$$

It is clear that whatever the choice of r^c, this matrix is not automatically diagonally dominant. In the limit $\Delta t \to 0$, one has $r^c \to x$ for any centering coordinates r^c interpolated from bead positions in $[0, \Delta t]$. So, in this limit the Jacobi matrix J associated with A has a spectral radius ρ bounded by $\rho \leq 1$. The inequality is applicable if one further assumes that no three consecutive rods are co-linear. Qualitatively, therefore, one can see that reducing the time step Δt promotes stability of this constraint calculation by the iterative approach recommended in Ref. [4]. In the general case, however, the iterative convergence of this linear system is not guaranteed (c.f. Ref. [13], p. 332). By including the velocity constraint in our calculations, $|r_i - r_{i+1}| \approx |x_j - x_{j+1}| \approx a$ to higher order. This, with the low probability of any three consecutive rods being co-linear, promotes stability of the method and accounts for the $\mathcal{O}(100)\times$ improvement in stable time step that we observe in Ref. [17].

3. RESULTS AND DISCUSSION

In Figure 1, a 400-bead polymer begins in the shape of a cut trefoil knot. This shape was chosen to encourage rod–rod crossing: the initial configuration is smooth for convenience. As time progresses (clockwise in the figure) the polymer contracts due to contraction of the rod-bead-rod angles in response to Brownian bombardment. In this figure the bead diameter is meant as a visual aid and has no significance in the model. The polymer rapidly becomes entangled, resulting in frequent rod–rod interactions which are handled in a stable manner by the algorithms described here.

We have also experimented with a configuration in which the trefoil knot is not cut. For this closed-loop polymer, the rod length constraint matrix, now $N \times N$, is a so-called Laub matrix: one that differs from the tridiagonal form by addition of $1, N$ and $N, 1$ entries which, in this application, link the polymer's ends. The velocity constraint matrix is modified similarly. For this special case our numerical experiments fail because the polymer becomes so twisted that successive sub-steps in the algorithm approach $\tau \to 0$. In this limit, any nudging factor $\epsilon > 0$ can potentially cause rods to cross. We note that this mode of failure, $\tau \to 0$, is physically correct though numerically inconvenient. Improvements to address this behavior are in progress.

Also in progress is the merger of these contact algorithms with our 3D embedded boundary Navier Stokes method. A model is being constructed to study so-called entropic trapping[7] by which size-dependent DNA separation occurs in a simple microfluidic device.
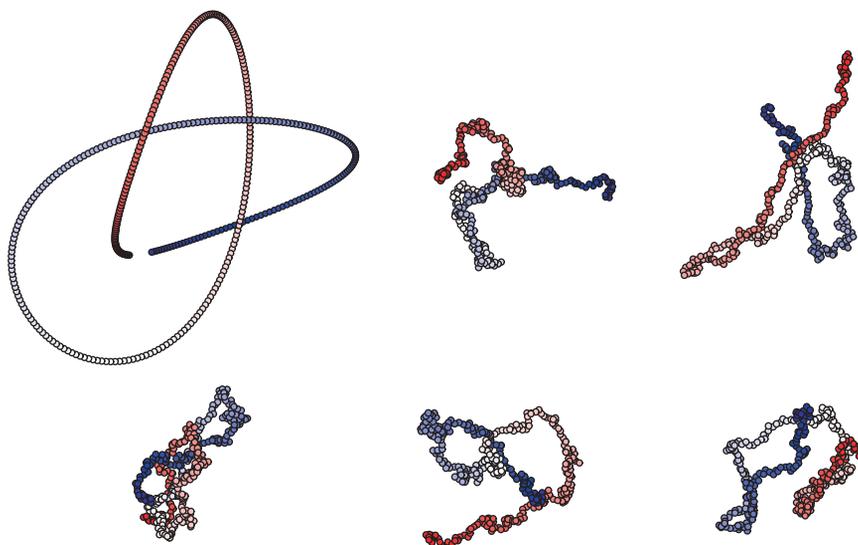
Fig. 1. A 400-bead polymer subject to Brownian motion only.

References

1. H. C. Anderson, *J. Comp. Phys.* 52, 24 (**1983**).
2. M. Barad, P. Colella, D. T. Graves, B. van Straalen, and D. Trebotich, *J. Comp. Phys.* (**2006**), in preparation.
3. M. D. Bybee, G. H. Miller, and D. Trebotich, Particle interactions in DNA flows, Technical Report UCRL-TR-217843, LLNL (**2005**).
4. G. Ciccotti, M. Ferrario, and J.-P. Ryckaert, *Mol. Phys.* 47, 1253 (**1982**).
5. P. Colella, Volume-of-fluid methods for partial differential equations. Godunov Methods: Theory and Applications, edited by E. F. Toro, Kluwer, New York (**2001**), pp. 161–176.
6. A. L. Garcia, J. B. Bell, W. Y. Crutchfield, and B. J. Alder, *J. Comp. Phys.* 154, 134 (**1999**).
7. J. Han and H. G. Craighead, *Science* 288, 1026 (**2000**).
8. P. J. Hoogerbrugge and J. M. Koelman, *Europhys. Lett.* 18, 155 (**1992**).
9. J. S. Hur, E. S. G. Shaqfeh, and R. Larson, *Journal of Rheology* 44, 713 (**2000**).
10. H. Johansen and P. Colella, *J. Comp. Phys.* 147, 60 (**1998**).
11. H. A. Kramers, *J. Comp. Phys.* 14,415 (**1946**).
12. A. Rahman and F. H. Stillinger, *J. Chem. Phys.* 55, 3336 (**1971**).
13. J.-P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, *J. Comp. Phys.* 23, 327 (**1977**).
14. J. Stoer and R. Bulirsch, Introduction to Numerical Analysis, Springer (**1999**), 2nd edn.
15. V. Symeonidis and G. E. Karniadakis, *J. Comp. Phys.* 218, 82 (**2006**).
16. D. Trebotich, G. H. Miller, and M. D. Bybee, *Nanoscale & Microscale Thermophys. Engr.* (**2006**), submitted. Also available as LLNL Technical Report UCRL-JRNL-223318.
17. D. Trebotich, G. H. Miller, P. Colella, D. T. Graves, D. F. Martin, and P. O. Schwartz, *Computational Fluid and Solid Mechanics* 1018 (**2005**).

RESEARCH ARTICLE