Numerical simulation methods for the Rouse model in flow

Michael P. Howard and Scott T. Milner

Phys. Rev. E **84**, 051804 — Published 28 November 2011

DOI: 10.1103/PhysRevE.84.051804

Numerical simulation methods for Rouse model in flow

Michael P. Howard and Scott T. Milner

Department of Chemical Engineering,

Pennsylvania State University, University Park, PA 16802

Abstract

Simulation of the Rouse model in flow underlies a great variety of numerical investigations of polymer dynamics, in both entangled melts and solutions, and in dilute solution. Typically, a simple explicit stochastic Euler method is used to evolve the Rouse model. Here, we compare this approach to an operator splitting method, which splits the evolution operator into stochastic linear and deterministic nonlinear parts, and takes advantage of an analytical solution for the linear Rouse model in terms of the noise history. We show that this splitting method has second-order weak convergence, whereas the Euler method has only first-order weak convergence. Furthermore, the splitting method is unconditionally stable, in contrast to the limited stability range of the Euler method. Similar splitting methods are applicable to a broad class of problems in stochastic dynamics in which noise competes with ordering and flow to determine steady-state order parameter structures.

I. INTRODUCTION

The Rouse model underlies a great many simulations of polymer dynamics. It is used in slip-link models, and to represent dilute polymer solutions (neglecting hydrodynamic interactions), as well as unentangled polymer melts and concentrated solutions. In the Rouse model, a polymer is represented as a sequence of monomer beads bonded by springs. Each monomer feels a spring force from its neighbors, a drag force with respect to the surrounding fluid, and a stochastic force representing the effect of thermal fluctuations.

The Rouse model in shear flow is of interest because the resulting chain motion is complex and unsteady. Individual chains stretch in flow, and tumble over themselves when the leading end of the chain dips below the trailing end.

This unsteady motion leads generally to a varied ensemble of molecular configurations, which for some purposes are not well represented by simple average quantities such as chain average orientation or stretch. This variety of molecular configurations for sheared dilute polymers has been termed "molecular individualism", [1] and is most easily explored by direct simulation. For the case of dilute and semidilute DNA solutions, such simulations have been compared to direct imaging studies carried out with fluorescently labeled DNA. [2, 3]

Numerical simulations of the Rouse model are also a central feature of sliplink simulations, which have been extensively used as a stochastic counterpart to tube-based constitutive theories of monodisperse and polydisperse linear chains. [4–7] In all of these simulations, a Rouse chain is somehow confined to a piecewise linear tube defined by a sequence of entanglement points called sliplinks, which are typically advected affinely with the mean flow.

The Rouse motion of the chain within the tube naturally imparts to the sliplink model the physics of contour fluctuations, constraint release, and stress relaxation via reptation. However, most of the CPU time in such simulations is spent on the Rouse motion; furthermore, the numerical method most often used is the simple but rather inaccurate and unstable explicit Euler method, which necessitates a very short timestep.

In many applications, the Rouse model is augmented with additional terms and interactions, which are typically nonlinear but deterministic. For example, the harmonic springs of the Rouse model can be upgraded to finitely extensible nonlinear elastic (FENE) springs,

to limit the fully extended length of the chain. [8] Or, for simulations of dilute chains in a solution, various approximate hydrodynamic interactions may be added. [9] In either case, the random forces in the model are unaltered, and remain the chief reason for using simple explicit methods like Euler and thus short timesteps, which result in uncomfortable limitations on the lengths of chains that can be simulated. [9]

In short, most polymer dynamics simulations require time-evolution of the Rouse equation in some form. Much of the computational power in such simulations is spent numerically integrating the Rouse equation. This can become a rate-limiting computational step for large or lengthy simulations. However, if there were an exact solution to the Rouse equation, the time-evolution of the Rouse equation for any forward timestep could be done in one quick step, greatly speeding up the simulation.

II. EXACT SOLUTION

The Rouse equation takes the form

$$\zeta\left(\frac{\partial \mathbf{R}(s,t)}{\partial t} - \mathbf{v}(\mathbf{R}(s,t))\right) = K\frac{\partial^2 \mathbf{R}(s,t)}{\partial s^2} + \mathbf{f}(s,t) \tag{1}$$

where $\mathbf{R}(s,t)$ is the chain position at arclength s and time t, $\mathbf{v}(\mathbf{r})$ is the imposed flow field, ζ is the monomeric drag coefficient, K is the stiffness of the spring between neighboring beads, and $\mathbf{f}(s,t)$ is the random force.

The noise $\mathbf{f}(s,t)$ is taken to be delta-correlated in time and space, and Gaussian distributed with zero mean, with variance given by

$$\langle \mathbf{f}_i(s,t)\mathbf{f}_j(s',t')\rangle = \frac{2\zeta}{\beta}\delta_{ij}\delta(s-s')\delta(t-t') \tag{2}$$

where $\beta = 1/(kT)$. The noise amplitude $2\zeta/\beta$ is determined by requiring that the chain diffusion constant D satisfies the Stokes-Einstein relation $D = kT/(N\zeta)$.

A. Rouse modes

The Rouse equation is a system of linear constant-coefficient PDEs with a linear stochastic driving force, which is analytically solvable with Fourier analysis. Each position vector

$\mathbf{R}(s,t)$ can be represented as a linear combination of the Rouse modes $X_p(t)$ as

$$\mathbf{R}(s,t)\cdot\hat{\mathbf{x}} = R_x(s,t) = \sum_{p=1}^{\infty} X_p(t)\cos\left(\frac{p\pi s}{n}\right) \tag{3}$$

where n is the arclength of the chain. Cosine expansion is chosen in order to impose tension-free ends on the chain. The $p = 0$ mode is omitted here because it is only responsible for translational motion. Analogous equations relate y and z coordinates $R_y(s,t)$ and $R_z(s,t)$ to mode amplitudes $Y_p(t)$ and $Z_p(t)$.

The inverse relation between the Rouse modes and the chain conformation is

$$X_p(t) = \frac{2}{n}\int_0^n R_x(s,t)\cos\left(\frac{p\pi s}{n}\right)ds \tag{4}$$

and analogously for $Y_p(t)$ and $R_y(s,t)$.

The mode noise x_p is obtained by Fourier cosine transform of the real space noises $f_x(s,t) = \mathbf{f}(s,t)\cdot\hat{\mathbf{x}}$ with Eqn. 4. The mode noise has variance

$$\langle x_p(t)x_q(t')\rangle = \frac{4\zeta}{\beta n}\delta_{pq}\delta(t-t') \tag{5}$$

A simple shear flow field takes the form

$$\mathbf{v}(\mathbf{r}) = \dot{\gamma}y\hat{\mathbf{x}} \tag{6}$$

with the shear rate $\dot{\gamma}$, the velocity direction taken as $\hat{\mathbf{x}}$, and the gradient direction taken as $\hat{\mathbf{y}}$. Since simple shear flow is a linear function of position, the Fourier transform can usefully be applied to it as well in Eqn. 1.

For a shear flow like Eqn. 6, only the R_x component in Eqn. 1 is coupled to the flow. R_y and R_z are each independent of the other components. The equation of motion for R_x has a source term $\dot{\gamma}R_y$ from the flow, but is completely decoupled from R_z. Thus, solving for R_z is trivial and analogous to solving for R_y, and we neglect it for simplicity.

Since the Rouse modes are only time dependent, Eqn. 1 reduces to a system of linear ordinary differential equations (ODEs). Substituting the modes and rearranging gives

$$\zeta X_p' = \dot{\gamma}Y_p - K_pX_p + x_p \tag{7a}$$

$$\zeta Y_p' = -K_pY_p + y_p \tag{7b}$$

where p is a positive integer, $K_p = \pi^2p^2K/n^2$, and x_p and y_p are the mode noises.

Solution for the Rouse modes Y_p (solving Eqn. 7b) for a finite $\Delta t = t_1 - t_0$ is a simple integrating factor problem. Let $\omega_p = K_p/\zeta$. Then,

$$\begin{aligned} Y_p(t_1) &= e^{-\omega_p \Delta t} Y_p(t_0) + \frac{1}{\zeta} \int_{t_0}^{t_1} e^{\omega_p (t - t_1)} y_p(t) dt \\ &= e^{-\omega_p \Delta t} Y_p(t_0) + \Delta Y_p \end{aligned} \tag{8}$$

Y_p has a fading memory of its current value (the exponential term), plus a change due to stochastic noise (ΔY_p). For sufficiently long times, the new mode value will "forget" its initial value and depend on only its noise history.

The governing equation for the modes X_p (Eqn. 7a) depends not only on X_p but also on Y_p (because of the shear flow force). Substituting Eqn. 8 into Eqn. 7a, and solving with integrating factors over the same time range gives

$$\begin{aligned} X_p(t_1) &= e^{-\omega_p \Delta t} \left(X_p(t_0) + \frac{\dot{\gamma} \Delta t}{\zeta} Y_p(t_0) \right) + \frac{1}{\zeta} \int_{t_0}^{t_1} e^{\omega_p (t - t_1)} x_p(t) dt \\ &\quad + \frac{\dot{\gamma}}{\zeta^2} \int_{t_0}^{t_1} (t_1 - t) e^{\omega_p (t - t_1)} y_p(t) dt \\ &= e^{-\omega_p \Delta t} \left(X_p(t_0) + \frac{\dot{\gamma} \Delta t}{\zeta} Y_p(t_0) \right) + \Delta X_p^{(x)} + \frac{\dot{\gamma}}{\zeta} \Delta X_p^{(y)} \end{aligned} \tag{9}$$

Like Y_p, X_p has a fading memory of its current value, and a stochastic forcing ($\Delta X_p^{(x)}$). It also has a fading memory of the flow force at its current value, and an additional stochastic forcing that depends on y_p. Because of this dependence, X_p is correlated to Y_p.

The integrals ΔY_p, $\Delta X_p^{(x)}$, and $\Delta X_p^{(y)}$ can be thought of as summations of the mode noises (weighted by the exponential) over the given time range. Since the mode noises are Gaussian random variables with zero mean, any linear combination of these variables is also a Gaussian random variable with zero mean. Thus, ΔY_p, $\Delta X_p^{(x)}$, and $\Delta X_p^{(y)}$ are all Gaussian random variables with zero mean, and their variances can be computed as functions of Δt since the variances of the mode noises are known (see Appendix A).

$$\langle \Delta Y_p^2 \rangle = \langle \Delta X_p^{(x)^2} \rangle = \frac{4}{\zeta \beta n} \frac{v_0(2\omega_p \Delta t)}{2\omega_p} \tag{10a}$$

$$\langle \Delta X_p^{(y)^2} \rangle = \frac{4}{\zeta \beta n} \frac{v_1(2\omega_p \Delta t)}{8\omega_p^3} \tag{10b}$$

where $v_0(x) = 1 - \exp(-x)$ and $v_1(x) = 2 - (2 + 2x + x^2)\exp(-x)$.

Frequently, the Rouse model is evolved forward in time to generate "snapshots" of a chain in a given flow. In order to do this numerically, the time period of interest Δt is usually

divided into m fundamental steps, and the chain is evolved over a noise history with m different noise pairs (integrating over the noise). However, since the noise integrals ΔY_p, $\Delta X_p^{(x)}$, and $\Delta X_p^{(y)}$ are themselves Gaussian random variables with known variance, it is sufficient to generate appropriate *representative* values for each over the *total* timestep Δt. Thus, only one triplet of Gaussian random variables needs to be generated for each Δt.

B. Mixed flows

The formal solution to the Rouse model generalizes to flow fields of mixed shear and extensional flows

$$\mathbf{v}(\mathbf{r}) = \dot{\gamma}(y\hat{\mathbf{x}} + \beta x\hat{\mathbf{y}}) \tag{11}$$

where $\dot{\gamma}$ is the shear rate, and $0 < \beta \leq 1$ is the "extensional character". If $\beta = 0$, $\mathbf{v}$ is the shear flow profile, and the solution of Section II A applies. If $\beta = 1$, $\mathbf{v}$ is pure extensional flow along $y = x$. Flow fields with β between zero and unity are of mixed type, with a character between shear and extensional.

The profile for a mixture of shear and extensional flow is still linear, so the solution proceeds through the Rouse modes as before

$$\zeta X_p' = \dot{\gamma} Y_p - K_p X_p + x_p \tag{12a}$$

$$\zeta Y_p' = \dot{\gamma}\beta X_p - K_p Y_p + y_p \tag{12b}$$

In matrix form

$$\begin{pmatrix} X_p' \\ Y_p' \end{pmatrix} = \frac{1}{\zeta}\begin{pmatrix} -K_p & \dot{\gamma} \\ \dot{\gamma}\beta & -K_p \end{pmatrix}\begin{pmatrix} X_p \\ Y_p \end{pmatrix} + \frac{1}{\zeta}\begin{pmatrix} x_p \\ y_p \end{pmatrix} \equiv \mathbf{M}\begin{pmatrix} X_p \\ Y_p \end{pmatrix} + \frac{1}{\zeta}\begin{pmatrix} x_p \\ y_p \end{pmatrix} \tag{13}$$

$\mathbf{M}$ has basis of eigenvectors $(1, \sqrt{\beta})$ and $(1, -\sqrt{\beta})$. If $\beta \neq 0$ (non-shear flow), $\mathbf{M}$ has a complete basis of eigenvectors, and is diagonalizable.

If λ_1 and λ_2 are the corresponding eigenvalues for $\mathbf{M}$, Eqn. 13 takes the form

$$\tilde{X}_p' = \lambda_1 \tilde{X}_p + \frac{1}{\zeta}\tilde{x}_p \tag{14a}$$

$$\tilde{Y}_p' = \lambda_2 \tilde{Y}_p + \frac{1}{\zeta}\tilde{y}_p \tag{14b}$$

where $\tilde{X}_p$, $\tilde{Y}_p$ and the noise terms are in the basis of the eigenspace. Note that the solution to the mixed flow case is actually simpler than the shear flow solution because the modes

are decoupled in the eigenspace. The solution to Eqn. 14 then proceeds as in Section II A (see Appendix C).

C. Finite extensibility

The Rouse model assumes that the bonds of a chain are harmonic springs, which exert a force proportional to the stretch, $F = -K\Delta x$. The change in spring force (or, the "stiffness" of the spring) $\partial F/\partial x$ is the spring constant K. However, in reality, the stiffness of the spring is not constant at all.

There are several models for such "anharmonic" (finitely extensible) springs. Although these models vary in form, they are fundamentally the same – the stiffness of the spring $\partial F/\partial x$ increases with stretch. In most such models, $\partial F/\partial x$ diverges at a certain stretch, which represents the spring becoming too stiff and "breaking". Simulating such springs in a stochastic equation is inconvenient because it can require integration near the singularity.

A quartic spring is a suitable model for a spring that stiffens without the spring force diverging. The stretching energy for a quartic spring is

$$U = \int \frac{1}{2} K \left[\left(\frac{\partial \mathbf{R}}{\partial s} \right)^2 + \frac{\alpha}{2} \left(\frac{\partial \mathbf{R}}{\partial s} \right)^4 \right] ds \tag{15}$$

This amounts to replacing the spring constant in the Rouse equation with a value that depends on local stretch, given by

$$K \to K_0 \left[1 + \alpha \left(\frac{\partial \mathbf{R}}{\partial s} \right)^2 \right] \tag{16}$$

Here α is a parameter that determines the stiffness of the springs. The value of α is chosen so that the spring becomes k times stiffer than K_0 at a given stretch $b^2 = (\partial_s \mathbf{R})^2$, so that

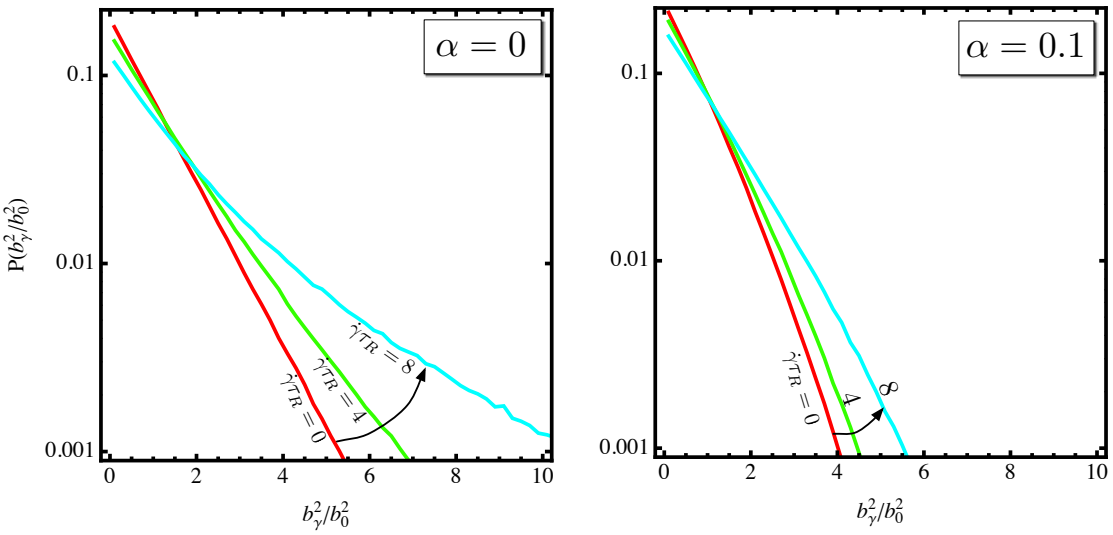
$$\alpha = \frac{k-1}{b^2} \tag{17}$$

In our simulations, we chose $\alpha = 0.1$ so that the spring constant would double at a squared bead-to-bead distance of 10.

D. Simulations

It is computationally simple to collect data on the behavior of chains in shear flow with an explicit update formula for the Rouse modes. A valid starting conformation is defined in

real space, transformed to Fourier space and evolved for a desired Δt, and then transformed back to real space for analysis. This data can be used to generate averages and probability distributions for chain properties, like end-to-end distance or inclination angle.



(a) Harmonic Spring

(b) Quartic Spring

FIG. 1: (Color online) Probability distributions for adjacent bead-to-bead distance squared for shear rates $\dot{\gamma}\tau_R = 0, 4, 8$; $n = 15$ bonds, $\alpha = 0.1$ for nonlinearity. Distance given in units of the average relaxed distance between adjacent beads, so $b_\gamma^2 = (\partial_s \mathbf{R})^2$ for a given shear rate, and $b_0^2 = \langle(\partial_s \mathbf{R})^2\rangle$ for a chain in zero shear with a linear spring.

One measure of interest to nucleation theory is the squared distance between adjacent beads. Fig. 1a shows this property for a chain with harmonic springs in increasing shear. As expected, the distributions broaden as $\dot{\gamma}\tau_R$ increases (more flow stress). However, these distributions also demonstrate the physical simplifications of the Rouse model. Although probability distributions of the distance between adjacent beads are expected to broaden with increasing shear, they should not broaden indefinitely. The bonds of the chain should stretch to a point, but then become stiffer and more resistant to change. Otherwise, a chain could stretch to arbitrary length.

Fig. 1b shows more realistic chain behavior after the quartic spring term is introduced. The distributions become narrower and steeper. This effect is most noticeable at high shear rates (where the harmonic springs would ordinarily become very stretched). For $\dot{\gamma}\tau_R = 8$, the distribution completely changes shape, becoming concave down. The effect of the quartic

spring is most significant at high shear rates because the nonlinear term $\alpha(\partial_s \mathbf{R})^2$ is negligible at small stretches (where the spring behaves mostly harmonically), and dominates at large stretches.

Although the anharmonic spring is a better model for the physical behavior of a chain in flow, it has unfortunate consequences for the formal solution to the Rouse equation. The Rouse equation (Eqn. 1) is solvable because it is linear, and can be diagonalized with a Fourier cosine transform. The nonlinear term "breaks" the solution because the Rouse equation is no longer diagonalizable. Any modification to the Rouse model that introduces a nonlinear term would have this effect. Some type of numerical method is required to evolve the modified Rouse equation.

III. OPERATOR SPLITTING

The typical approach to solving the nonlinear Rouse equation is to integrate over a discrete noise history with very small timesteps, using a stochastic numerical integrator. Although such methods solve deterministic differential equations reasonably well, they are not as effective at solving stochastic equations. Such numerical integrators march across the integration interval based on local values of the derivative at each integration step. If an equation is sufficiently "smooth" (several times differentiable), such methods work well because the value of the derivative changes smoothly and predictably.

Since deterministic equations are usually smooth, reasonably large integration steps can be taken, and these methods are efficient and accurate. However, stochastic equations are not smooth because of the noise, so small integration step sizes are required. A new noise term needs to be generated at each integration step. Since simulations of chain behavior typically require fairly large cumulative time scales, this process becomes computationally intensive.

Even though the modified Rouse equation is no longer linear, it is tempting to capitalize on the exact solution to the linear Rouse equation in designing a numerical solution method. The modified Rouse equation can be expressed as the sum of a linear part L and a nonlinear part N,where L represents the linear Rouse equation, and N represents the nonlinearity

introduced by the quartic spring.

$$\frac{\partial \mathbf{R}}{\partial t} = \mathbf{v}(\mathbf{R}) + \frac{1}{\zeta}\left[K_0 + K_0\alpha\left(\frac{\partial \mathbf{R}}{\partial s}\right)^2\right]\frac{\partial^2 \mathbf{R}}{\partial s^2} + \frac{1}{\zeta}\mathbf{f}$$

$$= \left(\mathbf{v}(\mathbf{R}) + \frac{K_0}{\zeta}\frac{\partial^2 \mathbf{R}}{\partial s^2} + \frac{1}{\zeta}\mathbf{f}\right) + \left[\frac{K_0\alpha}{\zeta}\left(\frac{\partial \mathbf{R}}{\partial s}\right)^2\right]\frac{\partial^2 \mathbf{R}}{\partial s^2} \tag{18a}$$

$$= L + N \tag{18b}$$

Although it would be difficult to solve this equation since it is nonlinear and stochastic, it is easy to solve either L or N independently. L can be evolved over the entire Δt in one step because it has an exact solution. N is integrable numerically with reasonably large timesteps since it is deterministic and smooth.

Operator splitting provides a general and effective way to solve equations like Eqn. 18b, in which the time evolution operator is the sum of pieces that can each be solved separately, but not easily at the same time. One first evolves L exactly, and then evolves N numerically. If these evolutions are done quickly enough (by swapping between operators over short time scales Δt_{swap}), then Eqn. 18b is solved exactly in the limit of $\Delta t_{swap} \to 0$. Error propagates as the swap times become "greedy" (too large). This method applies not only to the quartic spring, but also to *any* separable nonlinear operator.

Since L is solved in Fourier space, and numerical integration of N is done in real space, each swapping step requires one forward Fourier transform, and one inverse Fourier transform. Fast Fourier transforms (FFTs) can be applied if the solution is discrete in space. The Rouse model is discretized along arclength s into monomeric beads, with $s = 1, 2, ..., N$. FFTs are $O(N \log N)$ operations, so the conversion between Fourier space and real space can be done quickly.

Operator splitting is commonly employed for ordinary deterministic differential equations, as described in Ref. 10. Splitting was introduced by Ottinger in the context of stochastic dynamics of continuous fields. [11] The convergence properties of Ottinger's algorithm were analyzed by Petersen. [12] More recently, stochastic splitting was applied by Lennon et al. to improve efficiency of complex Langevin simulations. [13]

Ideally, a numerical method has error that quickly decreases as the computational timestep h is taken smaller, and remains stable as large steps are taken. Numerical methods are compared in terms of the rate at which error becomes small, and the range of timesteps over which the methods are stable.

Convergence measures how quickly error decreases as the computational step size decreases. For deterministic equations of motion, convergence is straightforward to describe. A method is "of order α" if the difference between the approximate solution y_a and the true solution y for a single step h is

$$y(h, y_0) - y_a(h, y_0) = O(h^{\alpha+1}) \tag{19}$$

Across a finite time interval Δt, $\Delta t/h$ steps are taken, and the cumulative error for an order α method is

$$y(\Delta t, y_0) - y_a(\Delta t, y_0) = O(h^{\alpha}) \tag{20}$$

If y is a vector, it is convenient to compute the error as the average of the sum of the squares of the differences between the components of y_a and y (the mean-square error). The mean-square error for a finite time interval will be $O(h^{2\alpha})$.

Stochastic differential equations can be thought of as deterministic equations with a particular time-dependent random driving force. As for deterministic equations, "strong" convergence measures the fidelity of the approximate solution to the true solution for a given noise history. For the Rouse model, the entire chain conformation $\mathbf{R}$ is taken as y, and the solution error is the mean-square error in the bead positions. Finally, to eliminate sensitivity to the particular noise history, the solution error is averaged over many noise histories.

Frequently in simulations of polymer dynamics, the statistical properties for a chain are of greater interest than the actual solution for a particular noise history. These properties are represented by distribution functions $P[\mathbf{R}(s,t)]$, the probability for a given trajectory to be generated from a starting condition $\mathbf{R}(s,0)$. The approximate distribution functions should approach the true distribution functions as the step size h becomes small.

Weak convergence describes how quickly the error between distributions decreases as h becomes small. If the distribution functions of possible trajectories match, then all correlation functions will also match. Here we briefly reprise the basic concepts related to strong and weak convergence of numerical methods for stochastic differential equations, which are treated extensively in the monograph of Kloeden and Platen, [14] and can be adapted to the present problem.

A method has weak convergence "of order β" if

$$\langle g[y(t)] \rangle - \langle g[y_a(t)] \rangle = O(h^{\beta}) \tag{21}$$

where g is any functional of the solution $y(t)$ over some time interval Δt, and the average is taken over both noise and initial conditions. Here, $g[y(t)]$ can be any correlation function, such as the autocorrelation function $y(t)y(t')$. Note that any correlation function can be expressed in terms of the distribution $P[y(t)]$, so convergence of the approximate distribution $P[y_a(t)]$ to the true distribution $P[y(t)]$ ensures the validity of Eqn. 21.

Although Eqn. 21 provides a formal definition for weak convergence, it is not useful for verifying the convergence order of a numerical method. It is impossible to confirm explicitly that Eqn. 21 holds for *all* possible correlation functions. It would be more convenient to have Eqn. 21 in a form where the convergence of only a *small set* of quantities needs to be checked.

Consider the evolution of the distribution over a short time interval Δt from a well-defined initial configuration corresponding to a very narrow initial distribution function $P(y_0)$. If the time interval is short enough, the distribution does not have time to change very much. If a numerical scheme is accurate, then the small changes in the approximate distribution $P(y_a)$ should also match the change in the true distribution $P(y)$. Since both the true and approximate distributions remain narrow after the short time Δt, it suffices to compare them by comparing the low-order moments of each distribution.

To verify weak convergence, the error between true and approximate moments of the distribution is compared in the limit of small time intervals. A useful theorem gives this criterion for weak convergence: the difference in the k-th moments for a single step h is [14]

$$\langle \Delta y^k(h) \rangle - \langle \Delta y_a^k(h) \rangle = O(h^{\beta+1}) \tag{22}$$

where $k = 1, 2, ..., 2\beta + 1$. The corresponding cumulative error for the moments evaluated after a finite interval Δt is

$$\langle \Delta y^k(\Delta t) \rangle - \langle \Delta y_a^k(\Delta t) \rangle = O(h^{\beta}) \tag{23}$$

However, note that if y is a vector, it becomes increasingly cumbersome to compute even the successive moments, which are tensors of increasing rank. This is certainly the case for the Rouse model for a chain of N beads, for which there are $2N$ degrees of freedom. Therefore in the present work, instead of explicitly verifying numerically the weak convergence criterion Eqn. 23, we content ourselves with exemplifying weak convergence Eqn. 21 for a particular property of interest, namely the mean squared distance between adjacent beads, $(1/n)\sum(\partial_s \mathbf{R})^2$.

A. Convergence properties

It is easier to understand the significance of convergence properties for some numerical method of interest by comparing it to a baseline numerical method. A suitable point of comparison is the explicit Euler method, because it is a typical solution method for stochastic equations. The explicit Euler method is ordinarily not a good numerical method for solving deterministic equations because it is order $\alpha = 1$ and has a narrow range of stability (see Section III B). However, the explicit Euler method is commonly used to solve stochastic equations because it only relies on the noise at the initial conditions to evolve the equation, and thus is simple to implement.

Operator splitting can be applied symmetrically or asymmetrically, yielding different convergence properties. Asymmetric splitting proceeds as described before, with alternation between L and N for equal times. In symmetric splitting, N is first evolved for half of the swap time, then L is evolved for the full swap time, and finally N is evolved for the remainder of the swap time. In the limit of infinite time evolution, symmetric and asymmetric splitting are indistinguishable. When evolved over finite time scales, symmetric splitting has better convergence properties, with only one additional FFT. For the purposes of this paper, operator splitting is taken to mean symmetric splitting.

The "true" dynamic for each method must be defined in order to compare error as the computational step size is varied. For the explicit Euler method, the true dynamics is defined as the solution for a stepsize $h = dt$. For operator splitting, the true dynamics is defined as the solution for a fundamental stepsize dt' when L and N are evolved on the swap time $\Delta t_{swap} = 2dt'$ (the shortest possible discrete symmetric swap time). We define the true evolution under the nonlinear operator N to be the solution obtained using an accurate numerical integrator. In this way we can focus on the error introduced only by operator splitting.

Approximations are computed for the explicit Euler method by taking $h = 2^k dt$, and for operator splitting by taking $\Delta t_{swap} = 2^k dt'$ for $k = 1, 2, ..., k_{max}$. For each method, the approximations and true dynamic are evolved over the same short time period so that solutions change on equal time scales. The difference between the approximations and the true dynamic will be the cumulative error (of order α or β).

Since all dynamics are compared over the same time period, the overall time explored

Δt must be at a minimum $\Delta t = 2^{k_{max}} dt$. However, the convergence theorems are only true for short stepsizes h. Thus, the fundamental steps dt and dt' must be very short times. Additionally, the explicit Euler method requires *very* small steps so that the solution does not explode over the total Δt.

To exemplify convergence orders with numerical results, the size of dt and dt' must be determined by trial-and-error for each method and convergence type. They must be chosen short enough that the distribution function does not have time to change very much, long enough that the distribution function *does* change, and also long enough that computational underflow is avoided. In general, good choices for dt and dt' to exhibit convergence properties are fractions of the shortest Rouse time τ_R/n^2, and dt for the explicit Euler method is much shorter than dt' for operator splitting.

Analysis shows that both the explicit Euler method and operator splitting exhibit strong convergence of order $\alpha = 1$ (see Appendix E). In order to verify strong convergence, the approximate and true dynamics for each method are evolved over a particular noise history. For the explicit Euler method, this simply requires computing a noise history at the start of each step, and applying an appropriate number of terms based on the size of h. For operator splitting, the exact solution to L must be adapted from continuous time to discrete time in order to apply the integrals over the noise as a summation of a noise history (see Appendix B).

Since the mean-square error is $O(h^{2\alpha})$, log-log plots of the error versus step (swap) size should appear linear with slope 2α. Both operator splitting and explicit Euler have approximately a slope of two, verifying that operator splitting and explicit Euler are strong order $\alpha = 1$.

In contrast to strong convergence, the explicit Euler method only has weak convergence order $\beta = 1$ while operator splitting has order $\beta = 2$ (see Appendix E). The change in the sum of squared distances between adjacent beads is computed using the explicit Euler method and operator splitting by averaging over initial configurations and noise. The error is log-log plotted versus the step (swap) size (Fig. 3).

Operator splitting has the same strong order as the explicit Euler method, but a better weak convergence order, which is often of greater interest. Note that in Fig. 2 and Fig. 3 the timesteps explored with operator splitting are considerably larger than those explored with explicit Euler. The timesteps for the explicit Euler method have to be taken shorter

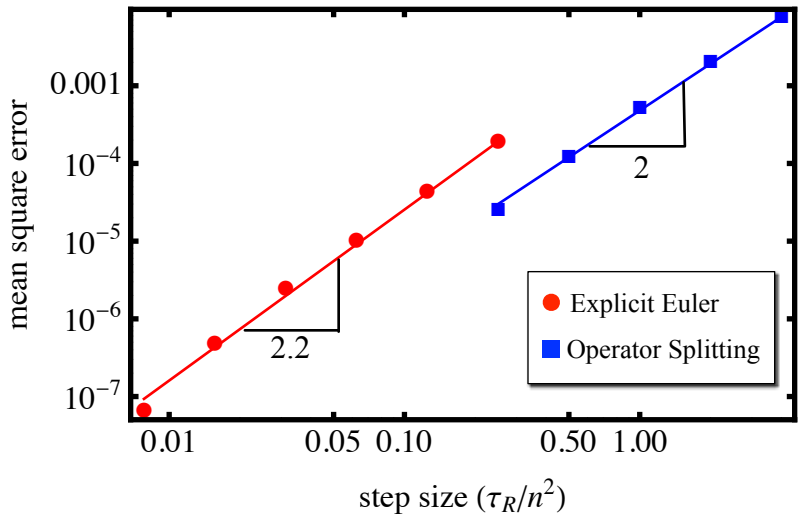


FIG. 2: (Color online) Log-log plots of mean-square error for bead positions. $\dot{\gamma}\tau_R = 4$, $n = 15$ bonds, $\alpha = 0.1$. For explicit Euler, $dt = (1/256)(\tau_R/n^2)$ and maximum step size = 64 dt. For operator splitting, $dt' = (1/16)(\tau_R/n^2)$ and maximum swap time = 64 dt'.

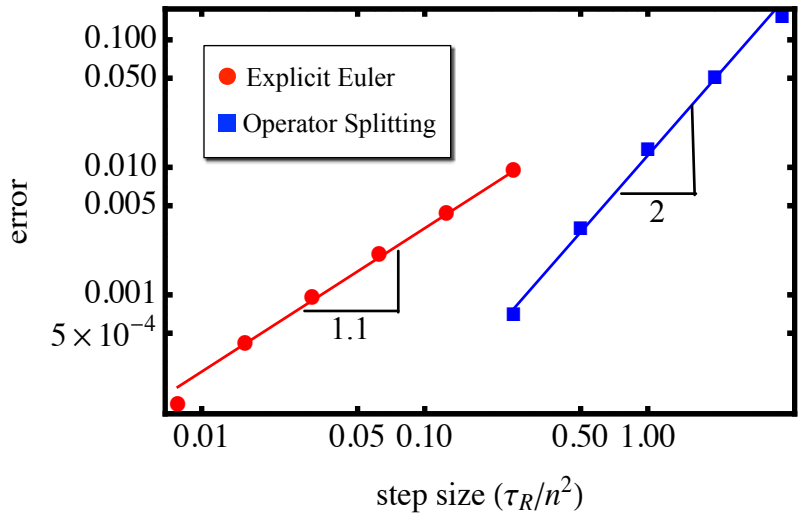


FIG. 3: (Color online) Log-log plots of the absolute averaged error in the mean-square distance between adjacent beads. $\dot{\gamma}\tau_R = 4$, $n = 15$ bonds, $\alpha = 0.1$. For explicit Euler method, $dt = (1/256)(\tau_R/n^2)$ and maximum step size = 64 dt. For operator splitting, $dt' = (1/16)(\tau_R/n^2)$ and maximum swap time = 64 dt'.

than operator splitting, or else the error in the solution diverges. This is a result of the most important advantage of operator splitting over the explicit Euler method: stability.

B. Stability

While convergence describes the behavior of the method as the time step becomes small, stability describes behavior as the time step becomes large. Stability is generally of more interest to running a simulation because one wants to take large timesteps to minimize the computational work.

Stability analysis studies how small error perturbations propagate as a numerical method marches forward. These errors may either die out, diverge, or maintain an approximately constant amplitude. The radius of stability describes how large of a step h can be taken without the method becoming unstable (causing the solution to diverge). Implicit numerical methods are typically more stable than explicit methods. Additionally, methods that are faithful at higher order derivatives tend to be stable.

The explicit Euler method is notoriously unstable. In the case of the nonlinear Rouse model, we observe that fundamental step sizes larger than $dt = 2\tau_R/n^2$ cause the method to become unstable for $n = 15$ bonds. This is why such small step sizes were required to verify the convergence orders (see Figs. 2 and 3).

Although the radius of stability can be investigated analytically, we have not carried out this analysis for operator splitting because of the complexities introduced by the nonlinear term. Instead, we have analyzed stability numerically by examining the steady-state error in the Rouse modes (Fig. 4). Let the error in the p-th mode ϵ_p be defined as

$$\epsilon_p = \frac{\langle (X_p - X_p^{(a)})^2 \rangle}{\langle X_p^2 \rangle} \tag{24}$$

where X_p is the true mode and $X_p^{(a)}$ is the approximate mode. With this definition, $\epsilon_p = 0$ if the modes match exactly, and $\epsilon_p = 1$ if they are totally uncorrelated.

Fig. 4 shows that as the swap time increases, the error in the Rouse modes increases. The higher modes are error-sensitive, so $\epsilon_p \to 1$ faster. The higher modes are more sensitive because changes occur on a shorter time scale than in the lower modes, so large swapping steps "skip" these dynamics. In contrast, the error in the lowest Rouse mode ($p = 1$) is less than 1% if the swap time is approximately half the Rouse time. Further, the error for each mode at a given swap time saturates to a steady-state value. Cumulatively, this implies unconditional stability for operator splitting.

Error in the high Rouse modes is not as important as error in the low Rouse modes for

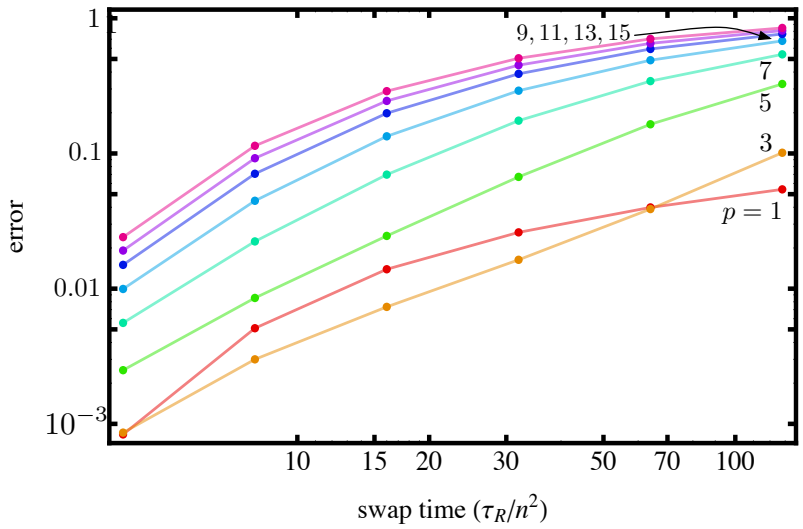


FIG. 4: (Color online) Steady-state errors for the odd-number Rouse modes. $\dot{\gamma}\tau_R = 4$, $n = 15$ bonds, $\alpha = 0.1$, $dt' = \tau_R/n^2$. Chains were allowed to evolve for $4\tau_R$ to reach steady state.

representing the polymer conformation. The low Rouse modes give the general orientation and shape of the conformation, while the high Rouse modes provide the "definition" to the conformation – the contours, sharp twists, and turns. As error is introduced to the high Rouse modes, chains become smoother (Fig. 5). However, the low Rouse modes are mostly faithful, so the chains remain synchronized to the true chain.

One can impose an error tolerance on the modes (say, 10% in the highest mode), and select the corresponding swap time. Error will enter the solution almost immediately after the simulation begins, but it will not increase. The swap time can then be chosen depending on the required fidelity at a given mode.

C. Computational benchmarking

The computational efficiency of operator splitting was compared to the explicit Euler method. Operator splitting was implemented using FFTW v.3.3b for FFTs (cite), and the fifth order adaptive step size Cash-Karp Runge-Kutta method and driver from *Numerical Recipes in C* for numerical integration.[15, 16] The time required for a single step was averaged over ten million steps, each with a different initial condition and noise.

It was observed that, as expected, explicit Euler had a constant computational time

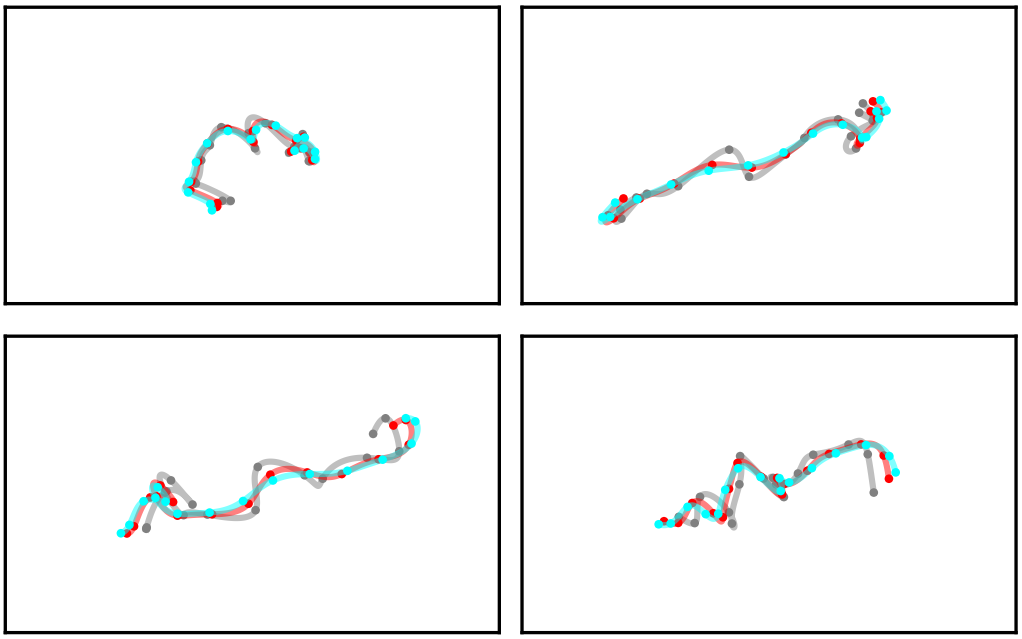


FIG. 5: (Color online) Sample conformations computed over a particular noise history with different swap rates. The conformation with the largest wiggles is the "true" dynamics. The progressively smoother conformations were run with swap times $\Delta t_{swap} = 4dt$ and $\Delta t_{swap} = 8dt$.

independent of step size. However, operator splitting had a variable time on a single step, due to the amount of work done by the numerical integrator, which depended on the starting chain configuration and step size.

It is worth noting that the amount of work required for operator splitting is dependent on the strength of the flow. Under strong shear flow, the chain becomes more stretched during the linear step than under weak or no flow. The stretched configuration makes the system stiffer for the numerical integrator in the nonlinear step, which necessitates shorter steps and thus more work. For our benchmarking tests, we have applied a fairly strong shear rate of $\dot{\gamma}\tau_R = 4$. If we were to simulate diffusion (no flow), it is likely that the added computational cost for operator splitting would be somewhat reduced.

A plot of the relative amount of computational time required on a single step (t_{OS}/t_E) for operator splitting and explicit Euler versus step size (Fig. 6) shows that splitting initially requires about five times more computational work for single short steps. However, when large steps are taken, the increase in work is significantly less than the increase in step size.

This means that operator splitting can significantly reduce the amount of work required in standard practice. We previously noted that explicit Euler certainly became unstable near $2\tau_R/n^2$ when $n = 15$. A safer step size to guarantee stability is $(1/4)\tau_R/n^2$, as we have used as a maximum in our convergence tests. If we enforce a modest error of no more than $\approx 30\%$ (square error 0.1) in the highest three Rouse modes when $n = 15$, we find from Fig. 4 that we can take a step of approximately $10\tau_R/n^2$ with operator splitting. From an interpolation of Fig. 6, this corresponds to approximately a ten fold increase in work. However, we were able to increase the step size forty fold, so this corresponds to a factor of four speed up using operator splitting.

At very large timesteps, splitting is not faithful to the dynamics of the highest modes, even if it is faithful to the lowest modes. However, the highest modes still take on random values with the correct equilibrium statistics because they are regenerated at each step from a random noise with fading memory of the previous value. This is ultimately what leads to the stability of splitting, and prevents the error in the modes from blowing up. It is thus reasonable to take *very* large timesteps, with considerably higher error in the highest Rouse modes than 30%. It is ultimately this stability property that gives operator splitting the computational edge over explicit Euler.

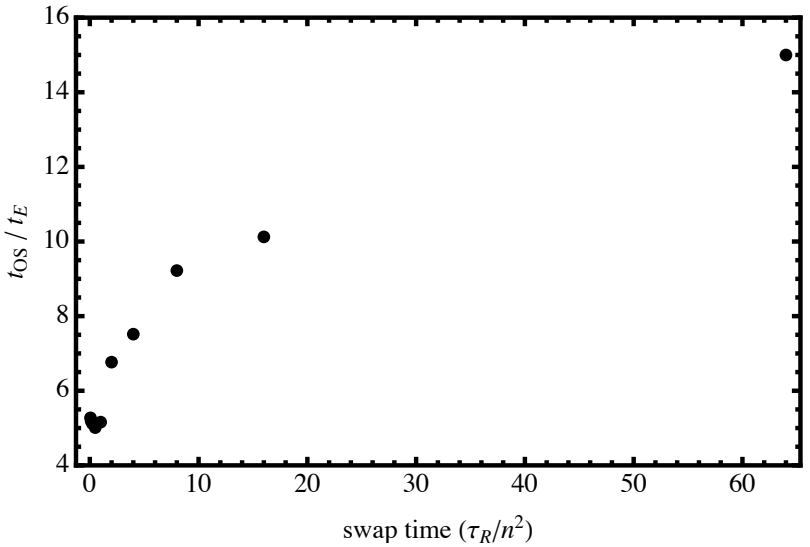


FIG. 6: Relative computational time required for operator splitting compared to explicit Euler method, $\dot{\gamma}\tau_R = 4$, $n = 15$ bonds, $\alpha = 0.1$. The computational time for explicit Euler was evaluated at $dt = (1/16)\tau_R/n^2$.

IV. CONCLUSIONS

Although ideally a numerical method has high order convergence and good stability properties, it is difficult to devise a method with weak convergence higher than order $\beta = 2$ because non-Gaussian random variables are required. Thus, operator splitting is a very useful numerical method because it has weak convergence $\beta = 2$ and seemingly unconditional stability.

Operator splitting was successfully implemented to evolve the nonlinear Rouse equation in shear flow in order to generate probability distributions for the distance between adjacent beads. Strong convergence order $\alpha = 1$ and weak convergence order $\beta = 2$ were verified numerically. The stability of operator splitting was explored numerically, and shown to be almost unconditionally stable. The exact solution to the linear Rouse equation was also extended to mixed flow, demonstrating the generalizability of the method.

An implementation of operator splitting requires

1. *"Splittable" equation.* The differential equation must be separable into an exactly solved part, and a numerically solvable part.

2. *Faithful numerical integrator.* A high order numerical integrator is required to minimize error in the numerical solution, so that error is introduced mostly by operator splitting.

3. *Fast Fourier transforms.* Frequently, the exact solution is treated in Fourier space and the numerical solution in real space. Fast Fourier transform allows for cheap conversion between the two at each swapping step.

Although implementing operator splitting may seem inconvenient compared to a simple implementation of the explicit Euler method, the higher order weak convergence and stability properties of operator splitting are worth the effort. In particular, large stepsizes may be taken so that evolutions which would be computationally intensive with explicit Euler can be done considerably more quickly.

Operator splitting can be applied to a variety of problems in the stochastic dynamics of continuous-field order parameters, for which the evolution equations are of the same essential form as the nonlinear Rouse model, i.e., conforming to the three conditions listed above. Examples include mixed binary fluids in flow, block copolymer order-disorder transitions in

flow, and flow alignment of ordered fluids such as smectic or nematic liquid crystals. Flow effects in these systems are often treated as initial value problems with noise effects neglected, but clearly for each case there is a regime where noise effects compete with flow and ordering in determining the steady-state structure, for which the present splitting approach should prove useful.

Acknowledgements: We thank NSF DMR-0907370 and CBET-1067554, and ACS-PRF 49964-ND7 for support.

Appendices

Appendix A: Shear flow solution

From Eqn. 7, the Rouse equations are

$$\zeta Y_p' = -K_p Y_p + y_p \tag{A1a}$$

$$\zeta X_p' = -K_p X_p + x_p + \dot{\gamma} Y_p \tag{A1b}$$

Rearranging and letting $\omega_p = K_p/\zeta$ gives a coupled system of ODEs

$$Y_p' = -\omega_p Y_p + \frac{1}{\zeta} y_p \tag{A2a}$$

$$X_p' = -\omega_p X_p + \frac{1}{\zeta} x_p + \frac{\dot{\gamma}}{\zeta} Y_p \tag{A2b}$$

Note that X_p is coupled to Y_p, but Y_p is independent.

The solution for Y_p over the finite time interval $\Delta t = t_1 - t_0$ is a simple integrating factor problem, with $\mu = \exp(\omega_p t)$

$$Y_p(t_1) = Y_p(t_0) e^{-\omega_p \Delta t} + \frac{1}{\zeta} \int_{t_0}^{t_1} e^{\omega_p (t - t_1)} y_p(t) dt \tag{A3}$$

$$= e^{-\omega_p \Delta t} Y_p(t_0) + \Delta Y_p \tag{A4}$$

Since ΔY_p is the integral over Gaussian random variables with zero mean, it is itself a Gaussian random variable with zero mean.

Solution for X_p begins with the same integrating factor to give

$$X_p(t_1) = X_p(t_0) e^{-\omega_p \Delta t} + \frac{1}{\zeta} \int_{t_0}^{t_1} e^{\omega_p (t - t_1)} x_p(t) dt + \frac{\dot{\gamma}}{\zeta} \int_{t_0}^{t_1} e^{\omega_p (t - t_1)} Y_p(t) dt \tag{A5}$$

This solution is similar to Eqn. A3, except for the coupled integral over Y_p. Considering only this term, and substituting Eqn. A3 gives

$$\frac{\dot{\gamma}}{\zeta} \int_{t_0}^{t_1} e^{\omega_p (t - t_1)} \left[Y_p(t_0) e^{\omega_p (t_0 - t)} + \frac{1}{\zeta} \int_{t_0}^{t} e^{\omega_p (t' - t)} y_p(t') dt' \right] dt \tag{A6}$$

Simplifying the integrals and exponentials

$$\frac{\dot{\gamma}}{\zeta} e^{\omega_p (t_0 - t_1)} Y_p(t_0) \int_{t_0}^{t_1} dt + \frac{\dot{\gamma}}{\zeta^2} \int_{t_0}^{t_1} e^{\omega_p (t - t_1)} \int_{t_0}^{t} e^{\omega_p (t' - t)} y_p(t') dt' dt \tag{A7}$$

$$= \frac{\dot{\gamma} \Delta t}{\zeta} e^{-\omega_p \Delta t} Y_p(t_0) + \frac{\dot{\gamma}}{\zeta^2} \int_{t_0}^{t_1} \int_{t_0}^{t} e^{\omega_p (t' - t_1)} y_p(t') dt' dt \tag{A8}$$

The twice-repeated integral can be simplified with the Cauchy formula with $n = 2$

$$\frac{\dot{\gamma}\Delta t}{\zeta} e^{-\omega_p \Delta t} Y_p(t_0) + \frac{\dot{\gamma}}{\zeta^2} \frac{1}{(2-1)!} \int_{t_0}^{t_1} (t_1 - t)^{2-1} e^{\omega_p (t - t_1)} y_p(t) dt \tag{A9}$$

The final update formula for X_p is given as

$$\begin{aligned} X_p(t_1) &= X_p(t_0) e^{-\omega_p \Delta t} + \frac{1}{\zeta} \int_{t_0}^{t_1} e^{\omega_p (t - t_1)} x_p(t) dt \\ &\quad + \frac{\dot{\gamma}\Delta t}{\zeta} e^{-\omega_p \Delta t} Y_p(t_0) + \frac{\dot{\gamma}}{\zeta^2} \int_{t_0}^{t_1} (t_1 - t) e^{\omega_p (t - t_1)} y_p(t) dt \qquad \text{(A10)} \\ &= e^{-\omega_p \Delta t} \left(X_p(t_0) + \frac{\dot{\gamma}\Delta t}{\zeta} Y_p(t_0) \right) + \Delta X_p^{(x)} + \frac{\dot{\gamma}}{\zeta} \Delta X_p^{(y)} \qquad \text{(A11)} \end{aligned}$$

It is apparent that $\Delta X_p^{(x)}$ is an independent Gaussian variable with variance analogous to ΔY_p. However, ΔY_p and $\Delta X_p^{(y)}$ are correlated Gaussian variables since they both depend on $y_p(t)$. Thus, $\langle \Delta Y_p^2 \rangle = \langle {\Delta X_p^{(x)}}^2 \rangle$, $\langle {\Delta X_p^{(y)}}^2 \rangle$, and $\langle \Delta Y_p \Delta X_p^{(y)} \rangle$ must be computed to generate appropriate representative Gaussian variables for the integrals.

Starting with $\langle \Delta Y_p^2 \rangle$,

$$\langle \Delta Y_p^2 \rangle = \frac{1}{\zeta^2} \int_{t_0}^{t_1} \int_{t_0}^{t_1} e^{-2\omega_p (t - t_1)} \langle y_p(t) y_p(t) \rangle dt dt \tag{A12}$$

We can substitute Eqn. 5 for $\langle y_p(t) y_p(t) \rangle$ to obtain

$$\begin{aligned} \langle \Delta Y_p^2 \rangle &= \frac{1}{\zeta^2} \int_{t_0}^{t_1} e^{2\omega_p (t - t_1)} \int_{t_0}^{t_1} \left[\frac{4\zeta}{\beta n} \delta_{pq} \delta(t - t') \right] dt dt \qquad \text{(A13)} \\ &= \frac{1}{\zeta^2} \int_{t_0}^{t_1} e^{2\omega_p (t - t_1)} \left(\frac{4\zeta}{\beta n} \right) dt \qquad \text{(A14)} \\ &= \frac{4}{\zeta \beta n} \frac{v_0(2\omega_p \Delta t)}{2\omega_p} \qquad \text{(A15)} \end{aligned}$$

where $v_0(x) = 1 - \exp(-x)$. This result applies analogously to $\langle {\Delta X_p^{(x)}}^2 \rangle$.

Proceeding as in Eqn. A12 for $\langle {\Delta X_p^{(y)}}^2 \rangle$

$$\langle {\Delta X_p^{(y)}}^2 \rangle = \frac{1}{\zeta^2} \int_{t_0}^{t_1} \int_{t_0}^{t_1} (t_1 - t)^2 e^{2\omega_p (t - t_1)} \langle y_p(t) y_p(t) \rangle dt dt \tag{A16}$$

which may be simplified as before with Eqn. 5 to give

$$\begin{aligned} \langle {\Delta X_p^{(y)}}^2 \rangle &= \frac{4}{\zeta \beta n} \int_{t_0}^{t_1} (t_1 - t)^2 e^{2\omega_p (t - t_1)} dt \qquad \text{(A17)} \\ &= \frac{4}{\zeta \beta n} \frac{v_1(2\omega_p \Delta t)}{8\omega_p^3} \qquad \text{(A18)} \end{aligned}$$

where $v_1(x) = 2 - (2 + 2x + x^2)\exp(-x)$.

Finally, the correlation term is computed

$$\langle \Delta Y_p \Delta X_p^{(y)} \rangle = \frac{1}{\zeta^2} \int_{t_0}^{t_1} \int_{t_0}^{t_1} (t_1 - t) e^{2\omega_p (t - t_1)} \langle y_p(t) y_p(t) \rangle dt dt \tag{A19}$$

which simplifies as before

$$\langle \Delta Y_p \Delta X_p^{(y)} \rangle = \frac{4}{\zeta \beta n} \int_{t_0}^{t_1} (t_1 - t) e^{2\omega_p (t - t_1)} dt \tag{A20}$$

$$= \frac{4}{\zeta \beta n} \frac{v_2(2\omega_p \Delta t)}{4\omega_p^2} \tag{A21}$$

where $v_2(x) = 1 - (1 + x)\exp(-x)$.

Appendix B: Discretization of shear flow solution

Discretize the operators forward in time, and with a central-difference with respect to space. Then, $\mathbf{R}(s,t)$ is discretized into $\mathbf{R}_{s,t}$ for integer values of s and t. We take the discrete Fourier transform as

$$\mathbf{X}_s = \frac{2}{N} \sum_{r=1}^{N} \mathbf{R}_r \cos\left(\frac{\pi}{N}(r - 1/2)(s - 1)\right) \tag{B1}$$

where $\mathbf{X}_s = \langle X_s, Y_s \rangle$, and r and s range from 1 to N. The inverse of this transform is given as

$$\mathbf{R}_s = \frac{\mathbf{X}_1}{2} + \sum_{r=2}^{N} \mathbf{X}_r \cos\left(\frac{\pi}{N}(r - 1)(s - 1/2)\right) \tag{B2}$$

Given this transform, it can be shown that the eigenvalues become

$$\lambda_s = 4 \sin^2\left(\frac{\pi}{2N}(s - 1)\right) \tag{B3}$$

The center of mass of the chain can be fixed at the origin by forcing $\mathbf{X}_1 = \mathbf{0}$.

We choose this form of the FFT because Eqn. B1 and Eqn. B2 are the discrete analogs of Eqn. 4 and Eqn. 3, respectively. Given this convention for the FFT, the discrete noise variance is

$$\langle x_p(t_i) x_q(t_j) \rangle = \frac{4\zeta}{\beta} \frac{\delta_{ij}}{\Delta} \frac{\delta_{pq}}{n} \tag{B4}$$

with Δ the fundamental step. This is the discrete analog of Eqn. 5.

Letting $\omega_s = (K/\zeta)\lambda_s$, the discrete update formula for Y_s becomes

$$Y_s(t + k\Delta) = (1 - \omega_s \Delta)^k \, Y_s(t) + \frac{\Delta}{\zeta} \sum_{i=1}^{k} (1 - \omega_s \Delta)^{i-1} \, y_p(t + (k - i)\Delta) \tag{B5}$$

where Δ is the fundamental time step, and $y_p(t+(k-i)\Delta)$ is an entry in the noise history. Likewise, it can be shown that the update formal for X_s is

$$\begin{aligned} X_s(t+k\Delta) = (1-\omega_s\Delta)^k X_s(t) + \frac{\dot\gamma\Delta}{\zeta^2}(1-\omega_s\Delta)^{k-1} Y_p(t) \\ + \frac{\Delta}{\zeta}\sum_{i=1}^{k}(1-\omega_s\Delta)^{i-1} x_p(t+(k-i)\Delta) \\ + \frac{\dot\gamma\Delta^2}{\zeta^2}\sum_{i=2}^{k}(i-1)(1-\omega_s\Delta)^{i-1} y_p(t+(k-i)\Delta) \end{aligned} \tag{B6}$$

Appendix C: Mixed flow solution

From Eqn. 13 in Section II B and letting $\omega_p = K_p/\zeta$,

$$\begin{aligned} \begin{pmatrix} \dot X_p \\ \dot Y_p \end{pmatrix} &= \begin{pmatrix} -\omega_p & \dot\gamma/\zeta \\ \dot\gamma\beta/\zeta & -\omega_p \end{pmatrix} \begin{pmatrix} X_p \\ Y_p \end{pmatrix} + \frac{1}{\zeta}\begin{pmatrix} x_p \\ y_p \end{pmatrix} \\ &\equiv \mathbf{M}\begin{pmatrix} X_p \\ Y_p \end{pmatrix} + \frac{1}{\zeta}\begin{pmatrix} x_p \\ y_p \end{pmatrix} \end{aligned} \tag{C1}$$

The eigenvalues of the matrix $\mathbf{M}$ are

$$\lambda_{1,2} = -\omega_p \pm \frac{\dot\gamma\sqrt{\beta}}{\zeta} \tag{C2}$$

It is apparent that λ_2 is always negative, and that λ_1 may be negative, positive, or zero depending on the relative magnitudes of $\omega_p, \dot\gamma, \beta$, and ζ. The corresponding eigenvectors are $(1, \sqrt{\beta})$ and $(1, -\sqrt{\beta})$. For non-shear flow ($\beta \neq 0$), the eigenvectors are linearly independent, and $\mathbf{M}$ is diagonalizable.

Let $\mathbf{\Lambda}$ be the diagonal matrix of eigenvalues, and $\mathbf{T} = [\mathbf{e}_1, \mathbf{e}_2]$ the matrix with corresponding eigenvectors as its columns. Diagonalizing Eqn. C1 gives

$$\tilde{\mathbf{S}}' = \mathbf{\Lambda}\tilde{\mathbf{S}} + \frac{1}{\zeta}\tilde{\mathbf{B}} \tag{C3}$$

where

$$\tilde{\mathbf{S}} = \begin{pmatrix} \tilde X_p \\ \tilde Y_p \end{pmatrix} = \mathbf{T^{-1}}\begin{pmatrix} X_p \\ Y_p \end{pmatrix} \tag{C4}$$

and

$$\tilde{\mathbf{B}} = \begin{pmatrix} \tilde x_p \\ \tilde y_p \end{pmatrix} = \mathbf{T^{-1}}\begin{pmatrix} x_p \\ y_p \end{pmatrix} \tag{C5}$$

Then, we readily arrive at Eqn. 14 from Section II B

$$\tilde{X}_p' = \lambda_1 \tilde{X}_p + \frac{1}{\zeta}\tilde{x}_p \tag{C6a}$$

$$\tilde{Y}_p' = \lambda_2 \tilde{Y}_p + \frac{1}{\zeta}\tilde{y}_p \tag{C6b}$$

which is easily solved over the finite interval $\Delta t = t_1 - t_0$ by an integrating factor $\mu = \exp(-\lambda t)$ to give

$$\tilde{X}_p(t_1) = e^{\lambda_1 \Delta t}\tilde{X}_p(t_0) + \frac{1}{\zeta}\int_{t_0}^{t_1} e^{-\lambda_1(t-t_1)}\tilde{x}_p(t)dt \tag{C7}$$

$$= e^{\lambda_1 \Delta t}\tilde{X}_p(t_0) + \Delta\tilde{X}_p \tag{C8}$$

and likewise for $\tilde{Y}_p$

$$\tilde{Y}_p(t_1) = e^{\lambda_2 \Delta t}\tilde{Y}_p(t_0) + \frac{1}{\zeta}\int_{t_0}^{t_1} e^{-\lambda_2(t-t_1)}\tilde{y}_p(t)dt \tag{C9}$$

$$= e^{\lambda_2 \Delta t}\tilde{Y}_p(t_0) + \Delta\tilde{Y}_p \tag{C10}$$

Observe that since $\tilde{x}_p$ and $\tilde{y}_p$ are linear combinations of Gaussian random variables x_p and y_p, $\Delta\tilde{X}_p$ and $\Delta\tilde{Y}_p$ are correlated Gaussian random variables. We must then compute the variances $\langle\Delta\tilde{Y}_p^2\rangle$, $\langle\Delta\tilde{X}_p^2\rangle$, and $\langle\Delta\tilde{X}_p\Delta\tilde{Y}_p\rangle$.

Proceeding for $\langle\Delta\tilde{Y}_p^2\rangle$ as in Appendix A, we obtain

$$\langle\Delta\tilde{Y}_p^2\rangle = \frac{1}{\zeta^2}\int_{t_0}^{t_1}\int_{t_0}^{t_1} e^{-2\lambda_2(t-t_1)}\langle\tilde{y}_p(t)\tilde{y}_p(t)\rangle dtdt \tag{C11}$$

From Eqn. C5 we know that $\tilde{y}_p(t) = (1/2)(x_p(t) - y_p(t)/\sqrt{\beta})$, so Eqn. C11 becomes

$$\langle\Delta\tilde{Y}_p^2\rangle = \frac{1}{\zeta^2}\int_{t_0}^{t_1} e^{-2\lambda_2(t-t_1)}\int_{t_0}^{t_1}\left\langle\frac{1}{4}\left(x_p(t) - \frac{y_p(t)}{\sqrt{\beta}}\right)^2\right\rangle dtdt \tag{C12}$$

$$= \frac{1}{\zeta^2}\int_{t_0}^{t_1} e^{-2\lambda_2(t-t_1)}\int_{t_0}^{t_1}\left\langle\frac{1}{4}\left(x_p(t)^2 - \frac{2}{\sqrt{\beta}}x_p(t)y_p(t) + \frac{y_p(t)^2}{\beta}\right)\right\rangle dtdt \tag{C13}$$

Since $x_p(t)$ and $y_p(t)$ are uncorrelated, the cross term can be eliminated from the average. Substituting from Eqn. 5, and integrating twice gives the variance expression

$$\langle\Delta\tilde{Y}_p^2\rangle = \frac{\beta+1}{2n\beta_0\zeta\beta\lambda_2}(e^{2\lambda_2\Delta t} - 1) \tag{C14}$$

where β_0 is the Boltzmann constant.

Solution for $\langle \Delta \tilde{X}_p^2 \rangle$ proceeds similarly, with $\tilde{x}_p(t) = (1/2)(x_p(t) + y_p(t)/\sqrt{\beta})$, to give

$$\langle \Delta \tilde{X}_p^2 \rangle = \frac{\beta + 1}{2n\beta_0 \zeta \beta \lambda_1} (e^{2\lambda_1 \Delta t} - 1) \tag{C15}$$

when $\lambda_1 \neq 0$. If $\lambda_1 = 0$, the solution simplifies to

$$\langle \Delta \tilde{X}_p^2 \rangle = \frac{\beta + 1}{n\beta_0 \zeta \beta} \Delta t \tag{C16}$$

Finally, computing $\langle \Delta \tilde{X}_p \Delta \tilde{Y}_p \rangle$ gives

$$\langle \Delta \tilde{X}_p \Delta \tilde{Y}_p \rangle = \frac{1}{\zeta^2} \int_{t_0}^{t_1} \int_{t_0}^{t_1} e^{-(\lambda_1 + \lambda_2)(t - t_1)} \langle \tilde{x}_p(t) \tilde{y}_p(t) \rangle dt dt \tag{C17}$$

The product $\tilde{x}_p(t)\tilde{y}_p(t)$ is a difference of squares, simplifying to

$$\langle \Delta \tilde{X}_p \Delta \tilde{Y}_p \rangle = \frac{1}{\zeta^2} \int_{t_0}^{t_1} e^{-(\lambda_1 + \lambda_2)(t - t_1)} \int_{t_0}^{t_1} \left\langle \frac{1}{4} \left(x_p(t)^2 - \frac{y_p(t)^2}{\beta} \right) \right\rangle dt dt \tag{C18}$$

which may be reduced with Eqn. 5 and twice integrated as before to give

$$\langle \Delta \tilde{X}_p \Delta \tilde{Y}_p \rangle = \frac{\beta - 1}{n\beta_0 \zeta \beta (\lambda_1 + \lambda_2)} (e^{(\lambda_1 + \lambda_2)\Delta t} - 1) \tag{C19}$$

This may be further simplified by observing that λ_1 and λ_2 are conjugates

$$\langle \Delta \tilde{X}_p \Delta \tilde{Y}_p \rangle = \frac{\beta - 1}{2n\beta_0 \zeta \beta \omega_p} (e^{-2\omega_p \Delta t} - 1) \tag{C20}$$

It should be noted that $\Delta \tilde{X}_p$ and $\Delta \tilde{Y}_p$ become totally uncorrelated if $\beta = 1$, the case of pure extensional flow.

Appendix D: Correlated random values

We need to generate two correlated Gaussian random variables x and y with zero mean. We want these variables to have a covariance matrix $\mathbf{M}$ such that

$$\mathbf{M} = \begin{pmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 \end{pmatrix} \tag{D1}$$

where $\sigma_{xx}^2 = \langle x^2 \rangle$, $\sigma_{yy}^2 = \langle y^2 \rangle$, and $\sigma_{xy}^2 = \sigma_{yx}^2 = \langle xy \rangle$.

The covariance matrix has a generating function

$$Z \propto \int \exp\left(-\frac{1}{2} \mathbf{v}^T \cdot \mathbf{M}^{-1} \cdot \mathbf{v} + \mathbf{h}^T \cdot \mathbf{v} \right) d\mathbf{v} \propto \exp\left(\frac{1}{2} \mathbf{h}^T \cdot \mathbf{M} \cdot \mathbf{h} \right) \tag{D2}$$

where $\mathbf{v}$ is the vector of correlated variables, and $\mathbf{h}$ is a conjugate variable.

Since it is our objective to obtain $\mathbf{v}$, and $\mathbf{M}^{-1}$ is not diagonal, we need to make a change of variables to $\mathbf{v}$ that allows us to generate independent Gaussian random variables with appropriate variances, and correlate them as a linear combination.

Since the matrix $\mathbf{M}$ is symmetric positive definite, $\mathbf{M}^{-1}$ has a complete basis of eigenvectors. Thus, $\mathbf{M}^{-1}$ is diagonalizable, and

$$\mathbf{M}^{-1} \cdot \mathbf{T} = \mathbf{\Lambda} \cdot \mathbf{T} \tag{D3}$$

where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues, and $\mathbf{T}$ is the matrix of eigenvectors, $[\mathbf{e}_1, \mathbf{e}_2]$. We can then expand $\mathbf{v}$ in the eigenvector basis as

$$\mathbf{v} = \mathbf{T} \cdot \mathbf{a} \tag{D4}$$

Since $\mathbf{M}^{-1}$ is a symmetric positive definite matrix, its eigenvectors are orthonormal such that $\mathbf{T}^T \cdot \mathbf{T} = \mathbf{I}$. Combining this result with Eqn. D3 and Eqn. D4, we obtain

$$\mathbf{v}^T \cdot \mathbf{M}^{-1} \cdot \mathbf{v} = \mathbf{a}^T \cdot \mathbf{\Lambda} \cdot \mathbf{a} \tag{D5}$$

where $\mathbf{a}$ is a vector of independent Gaussian random variables with zero mean, and with variances such that $\langle a_i^2 \rangle = 1/\Lambda_i$.

The vector of correlated Gaussian random variables may be recovered by generating a representative vector $\mathbf{a}$ of independent Gaussian random variables, and then taking $\mathbf{v} = \mathbf{T} \cdot \mathbf{a}$.

Appendix E: Convergence order analysis

Although a treatment of weak convergence for stochastic differential equations as applied to operator splitting can be found in Ref. 12 (with essential background materials in Ref. 14), as a convenience to the reader we briefly summarize in this appendix the basic elements of this analysis.

The starting point for developing methods for stochastic DEs is the Ito equation, which in differential form (for additive noise) is

$$dX = a(X)dt + b\, dW \tag{E1}$$

and in integral form is

$$X(t) = X_0 + \int_0^t dt'\, a(X(t')) + b \int_0^t dW(t') \tag{E2}$$

In the above, $W(t)$ is a Wiener process, i.e., $dW(t)$ is delta-correlated white noise. The noise integral $W(t) = \int_0^t dW('t)$ executes a Gaussian random walk, with $\langle W(t)\rangle = 0$ and $\langle W^2(t)\rangle = t$. Eq. (E1) is quite general: X can denote a single variable or a vector of values, $a(X)$ can be any function (linear or nonlinear) of X.

To resolve ambiguities involving averages over the noise $dW(t)$, it is helpful to regard the time t as discrete, with timestep Δt:

$$X_{i+1} = X_i + a(X_i)\Delta t + b\,\Delta W_i \tag{E3}$$

with noise ΔW_i a sequence of uncorrelated random steps, of magnitude $\pm\sqrt{\Delta t}$. The discretized equation is causal, since X at the "next" time X_{i+1} depends only on the "present" X_i and noise ΔW_i. Also, it is straightforward to average over such discretized noise.

Suppose we have some function $g(X(t), t)$ of our stochastic process $X(t)$, and want to know how g itself varies over some short time interval dt. The Ito formula, essentially the chain rule for stochastic processes, expands g about its value at time t as

$$dg = \frac{\partial g}{\partial t}dt + \frac{\partial g}{\partial X}\left(a(X)dt + b\,dW\right) + \frac{1}{2}\frac{\partial^2 g}{\partial X^2}b^2 dt \tag{E4}$$

In integral form, the relation reads

$$g(X(t)) = g(X(0)) + \int_0^t dt'\,\left(a(X(t'))g'(X(t')) + (1/2)b^2 g''(X(t'))\right) + b\int_0^t dW(t') \tag{E5}$$

in which g' denotes $\partial g/\partial X$, and for simplicity we have dropped explicit time-dependence (i.e., $\partial g/\partial t = 0$).

The formal solution Eq. (E2) can be expanded by repeatedly replacing $a(X(t'))$ and its derivatives using Eq. (E5), to obtain the Ito-Taylor expansion,

$$X(t) = X(0) + aI_{(0)} + bI_{(1)} + \left(a'a + \frac{1}{2}a''b^2\right)I_{(0,0)} + a'bI_{(1,0)} + a''b^2 I_{(1,1,0)} + \ldots \tag{E6}$$

in which a and its derivatives are all evaluated at the initial condition $X = X_0$. Also, $I_{(i,j,k,\ldots)}$ with $i, j, k, \ldots = 0$ or 1 denote various integrals over the noise,

$$I_{(i_1,i_2,\ldots,i_n)} = \int_0^t dW^{i_n}(s_n)\int_0^{s_n} dW^{i_{n-1}}(s_{n-1})\int_0^{s_{n-1}}\ldots\int_0^{s_2} dW^{i_1}(s_1) \tag{E7}$$

Dropping the noise ($b = 0$) gives the Taylor expansion of the deterministic equation $dX = a(X)dt$.

It is useful to assign to each of the various noise integrals $I_{(i,j,k,\ldots)}$ a scaling power of the time interval Δ. Each integral with a dW implies a factor $\Delta^{1/2}$ (since the variance of $\int dW$ is of order Δ) and each integral without a dW a factor Δ. Terms $I_{(0)}$ and $I_{(0,0)}$ are deterministic, given by $I_{(0)} = \Delta$ and $I_{(0,0)} = \Delta^2/2$ respectively. In what follows, we scale the noise integrals by appropriate factors of Δ to make power-counting explicit.

We are interested in the weak and strong convergence properties of two approximate methods, explicit Euler and operator splitting. The integral form of the Euler method over the time interval from $t = 0$ to $t = \Delta$ is

$$X_1 = X_0 + \Delta a(X_0) + b \int_0^\Delta dW(t') \tag{E8}$$

in which X_0 is the initial condition $X(0)$ and X_1 the final value $X(\Delta)$.

Symmetric stochastic splitting is defined (see main text) by alternating between exact solutions with one of two partial time evolution operators $a_1(X)$ and $a_2(X)$:

$$\begin{aligned} X_1 &= X_0 + \int_0^{\Delta/2} a_2(X_1(t'))dt' \\ X_2 &= X_1 + \int_0^{\Delta} a_1(X_2(t'))dt' + b\int_0^{\Delta} dW(t') \\ X_3 &= X_2 + \int_0^{\Delta/2} a_2(X_3(t'))dt' \end{aligned} \tag{E9}$$

in which X_3 is the final value $X(\Delta)$.

To determine strong convergence order, we simply compare the Ito-Taylor expansions of the approximate solutions to the true expansion, Eq. (E6). The Ito-Taylor expansion for the Euler method is obvious from Eq. (E8),

$$X(t+\Delta) = X(t) + \Delta a(X(t)) + b\Delta^{1/2} I_{(1)} \tag{E10}$$

and agrees with Eq. (E6) through terms of order $O(\Delta)$, hence we say the method is of strong order 1.0.

The Ito-Taylor expansion for the operator splitting method is obtained by using the full Taylor expansion (with $a = a_2$) to describe the "outer" deterministic steps, and the full Ito-Taylor expansion (with $a = a_1$) to describe "inner" stochastic step. The result of each step serves as the initial condition for the next, so the expansions are nested, with the final result

$$\begin{aligned} X(t+\Delta) = X(t) &+ \Delta(a_1 + a_2) + b\Delta^{3/2} I_{(1)} + b\Delta^{3/2}\left(a_1' I_{(1,0)} + (1/2)a_2' I_{(1)}\right) \\ &+ (1/4)\Delta^2\left(2(a_1+a_2)(a_1'+a_2') + b^2 a_1'' + 4b^2 a_1'' I_{(1,1,0)} + b^2 a_2' I_{(1)}\right) + \ldots \end{aligned} \tag{E11}$$

The above agrees with Eq. (E6) only through order $O(\Delta)$, so is of strong order 1.0 as well.

We now sketch a proof of the criterion for weak convergence, defined as follows. (See 14 p. 474, Theorem 14.5.2, Eqn. (5.12) for details.) Suppose we have some smooth function $g(X)$ that we want to average over trajectories, given a definite starting point X_0 at time t_0. Weak convergence means the average computed with approximate dynamics approaches the true average, as the time difference Δ becomes small:

$$\langle g(X(t))|X_0\rangle_{true} - \langle g(X(t))|X_0\rangle_{approx} = O(\Delta^{\beta+1}) \tag{E12}$$

where β is the weak order of convergence.

Note that if we have a finite time difference $t_{max} - t_0$, then the number of small timesteps Δ needed is $n = (t_{max} - t_0)/\Delta$. Since error accumulates on each timestep, the total error in the average is of order $O(\Delta^{\beta})$.

The "weak order β" notation is consistent with nomenclature for strong convergence, for which a "strong order α" method means agreement of the Ito-Taylor expansions of the true and approximate dynamics *through* order $O(\Delta^{\alpha})$ for a *single* step, and therefore an error for a finite timestep $t_{max} - t_0$ of order $O(\Delta^{\alpha})$.

Since the timestep Δ is small, $X(t)$ does not stray very far from X_0. This motivates us to expand $g(X(t))$ as

$$g(X(t)) = g(X_0) + g'(X_0)(X(t) - X_0) + (1/2)g''(X_0)(X(t) - X_0)^2 + \dots \tag{E13}$$

Evaluating the averages in Eqn. (E12) then reduces to evaluating the averages of moments of the true and approximate evolution of $X(t)$, defined as

$$M_k(\Delta t) = \langle (X(\Delta t) - X_0)^k \rangle \tag{E14}$$

for the exact and approximate evolutions.

How high a moment must we examine, to verify Eqn. (E12) for some given β? Note that the Ito-Taylor expansion for $X(t)$ begins

$$X(t) - X_0 = a(X_0)\Delta + b\int_0^t dW(t') + \dots \tag{E15}$$

To find the largest contribution to $\langle (X(t) - X_0)^k \rangle$, we take as many powers of noise as possible, since the noise scales as $\Delta^{1/2}$. (We need an even number of factors of the noise, so that the average does not vanish.)

As we consider higher moments, eventually even the lowest order contribution will be higher order than $\Delta^{\beta+1}$. Evidently, $k = 2\beta$ can generate terms of order Δ^{β}. Therefore, we need to compare the real and approximate dynamics up through the 2βth moment, to verify weak convergence of order β. (The average of a term with $k = 2\beta + 1$ powers of noise will vanish, since this k is odd, hence the error in the noise moments will be $O(\Delta^{\beta+1})$.)

Since the moments to be compared are to be computed for small Δ, we use the Ito-Taylor expansions for the true and approximate solutions, together with a set of results for averaging various products of noise integrals. The results we need to sufficient order for present purposes are

$$\begin{aligned} \langle I_{(1)}^2 \rangle &= 1 \\ \langle I_{(1)}^4 \rangle &= 3 \\ \langle I_{(1)} I_{(1,0)} \rangle &= 1/2 \\ \langle I_{(1,0)}^2 \rangle &= 1/3 \end{aligned} \tag{E16}$$

Any averages of noise integrals in which a given noise increment $dW(t)$ appears with an odd power, will vanish, so that $\langle I_{(1)} \rangle$, $\langle I_{(1,0)} \rangle$, $\langle I_{(1)}^3 \rangle$, $\langle I_{(1)} I_{(1,0)}^2 \rangle$, $\langle I_{(1)}^2 I_{(1,0)} \rangle$, and $\langle I_{(1)} I_{(1,1,0)} \rangle$ are all zero.

Carrying out moment averages using these methods, we can verify that the explicit Euler method is weak order 1.0, while operator splitting is weak order 2.0.

---

[1] P. deGennes, Science, **276**, 1999 (1997).

[2] J. Hur, E. Shaqfeh, H. Babcock, D. Smith, and S. Chu, Journal of Rheology, **45**, 421 (2001).

[3] J. Hur, E. Shaqfeh, H. Babcock, and S. Chu, Physical Review E, **66**, 011915 (2002).

[4] J. Schieber, J. Neergaard, and S. Gupta, Journal of Rheology, **47**, 213 (2003).

[5] S. Shanbhag and R. Larson, Macromolecules, **37**, 8160 (2004).

[6] A. Likhtman, Macromolecules, **38**, 6128 (2005).

[7] Y. Masubuchi, G. Ianniruberto, F. Greco, and G. Marrucci, Rheol Acta, **46**, 297 (2006).

[8] J. Hur, E. Shaqfeh, and R. Larson, Journal of Rheology, **44**, 713 (2000).

[9] C. Hsieh and R. Larson, Journal of Rheology, **48**, 995 (2004).

[10] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical recipes: The art of scientific computing," (Cambridge University Press, 2007) Chap. 20.3.3.

[11] H. C. Ottinger, "Stochastic processes in polymeric fluids," (Springer, Berlin, 1996) Chap. 3.

[12] W. Petersen, Siam J Numer Anal, **35**, 1439 (1998).

[13] E. M. Lennon, G. O. Mohler, H. D. Ceniceros, C. J. Garcia-Cervera, and G. H. Fredrickson, Multiscale Model Sim, **6**, 1347 (2007).

[14] P. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations (Stochastic Modelling and Applied Probability)* (Springer, Berlin, 2011).

[15] M. Frigo and S. G. Johnson, Proceedings of the IEEE, **93**, 216 (2005), special issue on "Program Generation, Optimization, and Platform Adaptation".

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical recipes in c (2nd ed.): the art of scientific computing," (Cambridge University Press, 1992) Chap. 16.2.