

Supplementary Information for KOBRA: A Fluctuating Elastic Rod Model for Slender Biological Macromolecules

Robert Welch, Sarah A. Harris, Oliver G. Harlen and Daniel J. Read

An accompanying dataset, including molecular dynamics, KOBRA trajectory files and python scripts for plot generation, can be found in the University of Leeds research data repository at: doi.org/10.5518/760. We also (at time of writing) maintain a mirror at: <https://bitbucket.org/Robert-Welch/kobra-raw-data/>.

1 Material Axis Update

Displacing a node affects not only the segments on either side of it, but also the orientation of the material axes associated with those segments. This occurs during a simulation, both when executing node movements during a timestep, or when perturbing node positions to establish the force on a node. We can use parallel transport as described in the main text to transport the material axis from the previous (unperturbed) segment to the new (perturbed) one. For example, if we move the node at position i , then the two elements p_i and p_{i-1} will change, and so we update the material axis m_i and m_{i-1} :

$$m'_i = P(m_i, l_i, l'_i) = R(l_i, l'_i) \cdot m_i \quad (1)$$

$$m'_{i-1} = P(m_{i-1}, l_{i-1}, l'_{i-1}) = R(l_{i-1}, l'_{i-1}) \cdot m_{i-1} \quad (2)$$

where m' is the updated material axis, P is an application of parallel transport, m is the unperturbed material axis, l is the (normalised) unperturbed segment, l' is the (normalised) perturbed segment, and R is the parallel transport rotation matrix.

2 Algorithm and performance

The Ndc80 protein complex is part of a set of molecular machinery responsible for attaching the chromatids to the microtubules of the mitotic spindle during cell division. These protein complexes are millions of atoms in size, and the forces used to move the chromatids are generated by the depolymerisation of spindle microtubules, a process that occurs on timescales of seconds to minutes¹ and length scales of $> 100\text{nm}$.² It is therefore necessary to build extremely performant and highly parallel algorithms capable of reaching these time and length scales.

The rod algorithm is implemented in C++ and is designed to run with a small memory footprint and cache-optimised data structures. A simple API allows for arbitrary additional forces and energies to be applied to the rods, to permit the development of new features and coupling with other simulations. The structure of the algorithm as follows:

- Initialise the rod.
 - Read rod header info,
 - Allocate memory,
 - Read structural data and convert to internal units.
- For each timestep:
 - For each node (OpenMP parallel):

- * Get forces from gradients in energy by perturbing all degrees of freedom (x, y, z, twist).
- For each node and element twist degree of freedom:
 - * Get thermal noise,
 - * Compute and apply dynamics,
 - * Update material axes.
- If algorithm is at a check timestep:
 - write to trajectory file

The most intensive computation task is the calculation of the forces from the gradient of the energy, which is computed using OpenMP (shared memory) parallelisation and takes around 96% of the program's runtime. This gives the program a serial fraction of 4%, although computing the dynamics in parallel is also possible. Performance and scaling are given in Fig. 1 and Fig. 2. We observe that simulation time per timestep is simply proportional to rod length, and inversely proportional to number of processors (an ideal scaling).

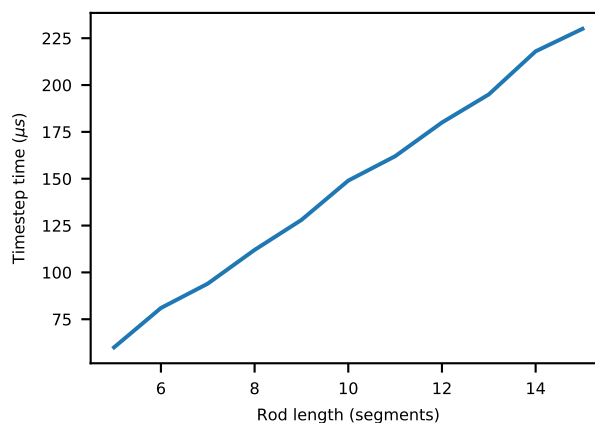


Fig. 1 Single-thread performance scaling with rod length, running on an Intel Xeon E5-2670.

3 Validation of the Equipartition theorem

The equipartition theorem states that the average energy will be $\frac{1}{2}k_B T$ per degree of freedom (D.O.F) where T is the temperature and k_B is Boltzmann's constant.

To correctly compute the expected equipartition energy, we must consider the number of degrees of freedom the system has. For a rod of N nodes and $N - 1$ segments:

- Number of stretch D.O.F. = $N - 1$

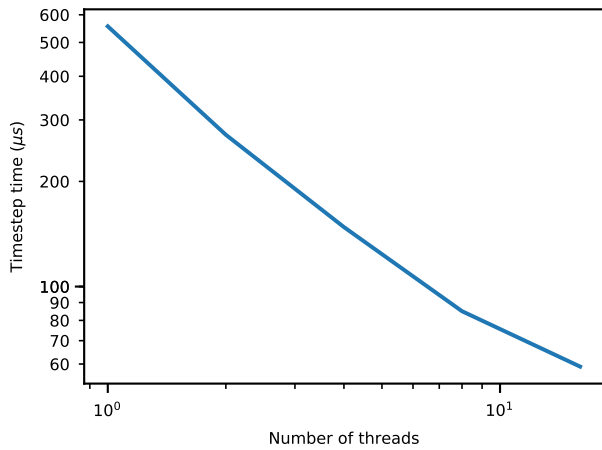


Fig. 2 Shared memory parallel scaling on a dual Intel Xeon E5-2670 system, with a fixed rod length of 32 nodes.

- Number of twist D.O.F. = $N - 2$
- Number of bend D.O.F. = $2(N - 2)$

The latter contains a factor of 2 because bending can occur in two orthogonal directions. Additionally, the entire rod can rotate and translate in six degrees of freedom, although (in the absence of inertia) these do not affect the energy.

A test system, composed of 10 elements, with a total equilibrium length of 100nm and radius $a = 5\text{nm}$, in a medium with dynamic viscosity $\mu = 0.6913\text{MPa}\cdot\text{s}$ (pure water at 310K ³) was simulated for $10\mu\text{s}$. At a temperature of 300K , we would expect the thermal energy to be $\frac{1}{2}k_B T = 2.07 \times 10^{-21}\text{J}$ per degree of freedom. The mean energies per degree of freedom for this trajectory are given in table 3, split according to stretching, bending and twisting.

Stretch	$(2.11 \pm 0.16) \times 10^{-21}\text{J}$
Bend	$(2.16 \pm 0.11) \times 10^{-21}\text{J}$
Twist	$(2.06 \pm 0.16) \times 10^{-21}\text{J}$

Table 1 Average values of the energy in each degree of freedom for a test rod. All parameters are isotropic and inhomogeneous.

4 Extraction of the bending stiffness matrix from fluctuations in a dynamic simulation

4.1 Small bending fluctuations and application of equipartition

If at a given node, i , the average length, L_i and the stiffness matrix, \mathbf{B}_i are such that $k_B T L_i \lambda_{\min}^{-1} \ll 1$ (where λ_{\min} is the smallest eigenvalue of the stiffness matrix \mathbf{B}_i) then the thermal bending fluctuations about that node are expected to be small. In such a case, a simple equipartition argument, leading to Eq. (35) in the main text, can be used to recover the bending stiffness matrix from the trajectory of a dynamical simulation. This is done by extracting the covariance matrix (Eq. (32)) of the bending fluctuations as measured in the simulation.

This method was validated against a trajectory from a KOBRA rod with known values of \mathbf{B} , chosen to be both inhomogeneous (\mathbf{B} varies along the rod) and anisotropic. Here, the question is whether we can extract the (known) values of \mathbf{B} from only the configurations the rod explores during the simulation. The rod had total length 257nm and typical length of rod elements 1.2nm (with 208 rod elements). The smallest eigenvalues of \mathbf{B} are of order $1.5 \times 10^{-28}\text{m}^4\cdot\text{Pa}$ (chosen to be in the typical range observed in NDC80 simulations). In this case the product $k_B T L_i \lambda_{\min}^{-1} \approx 0.034$ so that bending fluctuations are expected to be small. A simulation with 2000 frames recorded at intervals of 5ns was analysed. Fig. 3 compares the known eigenvalues of \mathbf{B} with the values extracted from the fluctuations measured in the trajectory using Eq. (35). Although there remains some statistical sampling error, it can be seen that the \mathbf{B} matrix is recovered to within 10%.

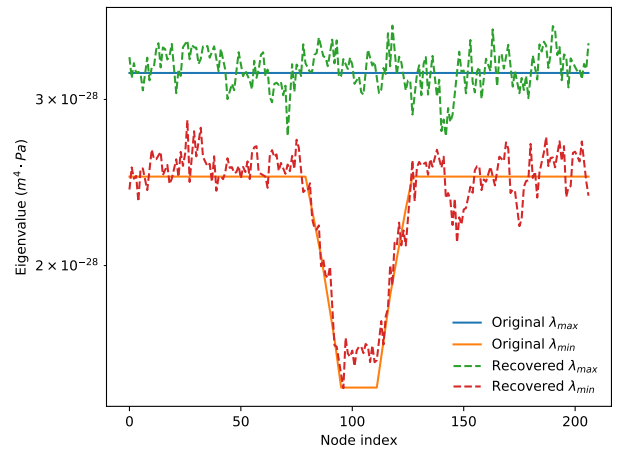


Fig. 3 'Recovered' eigenvalues of the \mathbf{B} matrix from a simulation trajectory, compared to the actual values given to the nodes in that simulation. Here, the two lines represent the maximum and minimum eigenvalues.

A long trajectory is required in order to gather sufficient statistics of the fluctuations. The mean fractional error due to under-sampling in the eigenvalues of \mathbf{B} for a given trajectory length n is computed for each frame in the trajectory:

$$\delta \mathbf{B}^{0,n} = \frac{\sqrt{\frac{\Delta \mathbf{B}_{0,n}}{n}}}{\mathbf{B}} \quad (3)$$

Where $\delta \mathbf{B}^n$ is the error in \mathbf{B} for all the frames in the trajectory from 0 to n , $\Delta \mathbf{B}^n$ is the difference between the currently calculated and final (or known) value of \mathbf{B} and of \mathbf{B} .

The fractional error in the \mathbf{B} recovery for the trajectory used to generate Fig. 3 is shown in Fig. 4. The results for the all-atom trajectory is shown in Fig. 5.

Note that the timescales of rod and all-atom simulation trajectories are not directly comparable. First, the rod simulation is overdamped, meaning it explores conformational space more slowly than the MD simulations which uses an "implicit solvent" is and is therefore less damped because the solvent (and its viscosity) is not explicitly included in the simulation. Also, in an all-atom trajectory, there are modes of vibration which occur on

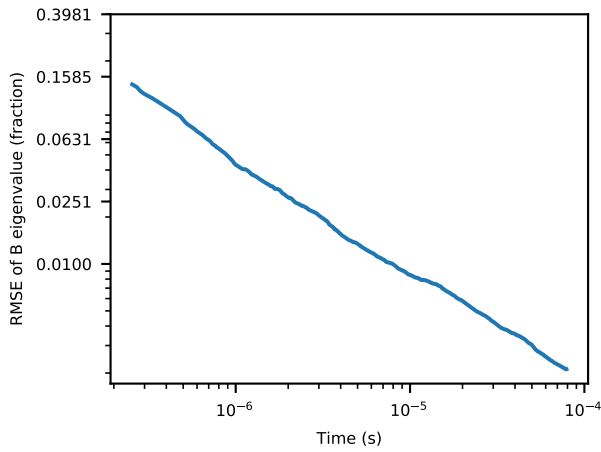


Fig. 4 Fractional RMSE of the \mathbf{B} eigenvalues for different fractions of the sample trajectory used to generate figure 3. The very start of the trajectory has been truncated to better show the scale.

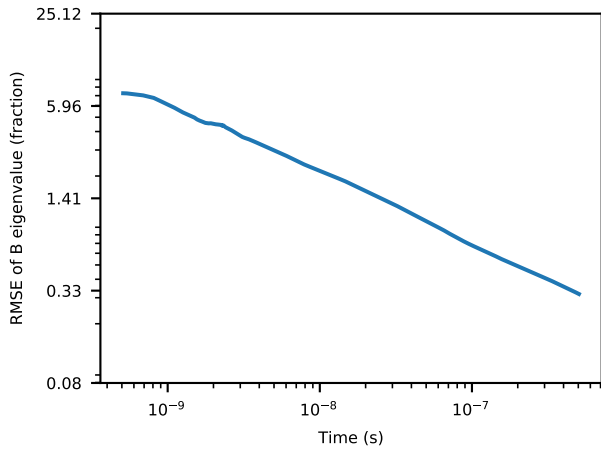


Fig. 5 Fractional RMSE of the \mathbf{B} eigenvalues for different fractions of the all-atom trajectory. The very start of the trajectory has been truncated to better show the scale.

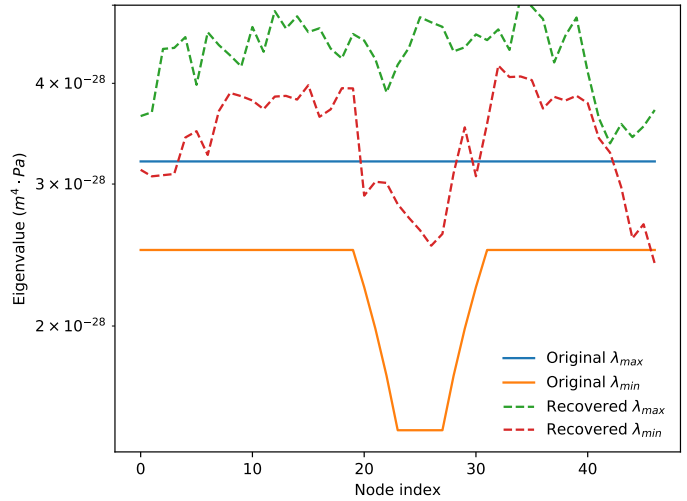
smaller timescales, and are not present in the rod simulation.

4.2 An iterative scheme for recovering the values of the \mathbf{B} matrix

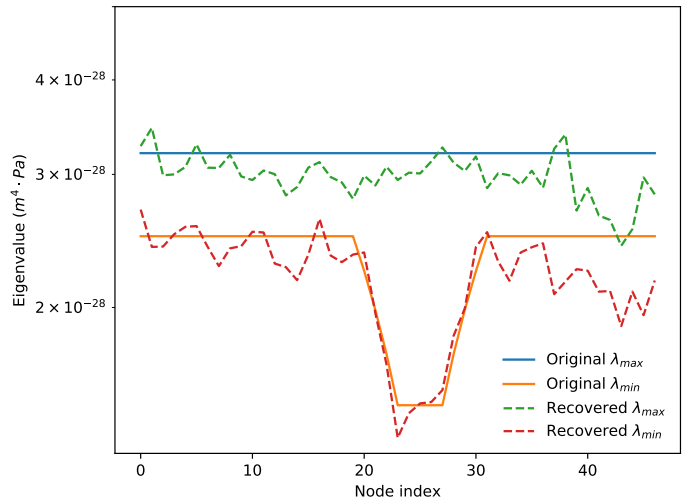
As noted in the Section 3.3 of the main text, if the bending fluctuations at a given node are larger, then non-linearities in the mapping from the Cartesian co-ordinates of node fluctuations onto the generalised co-ordinates corresponding to the bending degrees of freedom means that the probability distribution for bending co-ordinates are typically perturbed from their initially expected normal distributions and equipartition does not exactly apply. Under such circumstances, an iterative method and formula (Eq. 36) was proposed to recover the \mathbf{B} matrix from the observed fluctuations.

Increased bending at nodes occurs either if the elements of the stiffness matrix \mathbf{B} are smaller, or if the rod is more coarsely discretised using longer elements. Fig. 6 shows the results of a test in which we retain the same \mathbf{B} matrix and overall rod length as

in the simulations for Fig. 3, but decrease the number of rod elements used to discretise the rod (so that each individual element is longer). The typical length of rod elements is now 5nm (with 53 elements). In this case the product $k_B T L_i \lambda_{min}^{-1} \approx 0.14$ and the bending at each node in the trajectory is larger. As seen in Fig. 6a, application of equipartition via Eq. (35) does not recover the correct \mathbf{B} (as the eigenvalues are overestimated). However, a single iteration of Eq. (36), as shown in Fig. 6b is sufficient to converge to the correct \mathbf{B} within the statistical sampling error.



(a) Original recovery



(b) Iteration 1

Fig. 6 Maximum and minimum eigenvalues of the \mathbf{B} matrix using the iterative parameterisation scheme, compared to known reference values. After one iteration, the values of \mathbf{B} have converged.

Notes and references

- 1 R. Milo, P. Jorgensen, U. Moran, G. Weber and M. Springer, *Nucleic Acids Research*, 2010, **38**, D750–D753.
- 2 S. Westermann, H.-W. Wang, A. Avila-Sakar, D. G. Drubin, E. Nogales and G. Barnes, *Nature*, 2006, **440**, 565–569.
- 3 J. Cooper and R. Dooley, *The International Association for the Properties of Water and Steam*, Berlin, Germany, 2008.