Supporting Material:
A multi-scale dynamic model of DNA supercoil relaxation by topoisomerase IB

Todd D. Lillian, Maryna Taranova, Jeff Wereszczynski,
Ioan Andricioaei, and N. C. Perkins

Numerical Algorithm

Here we outline the finite difference algorithm used to simulate DNA in the accompanying paper. Our algorithm was originally developed in (1). Consequently, here we will repeat many of the equations of (1). In addition, our algorithm and notation also closely follows the cable dynamics algorithm of (2); however, our elastic rod formulation (Eqs. 2-5) is distinct. The algorithm is now written in MATLAB® (Natick, Massachusetts, USA) and is formulated as follows.

We begin by rewriting the system of nonlinear partial differential equations (Eqs. 2-5) describing the elastic rod formulation as

$$M\dot{Y} + KY_s + F = 0. \tag{S1}$$

Here Y is a 12×1 vector of field variables $\{v, \omega, \kappa, f\}$ with $\dot{Y} = \frac{\partial Y}{\partial t}$ and $Y_s = \frac{\partial Y}{\partial s}$ as partial derivatives with respect to space and time respectively. The matrices M and K are 12×12 matrices defined as

$$M = \begin{bmatrix} \Theta & \Theta & \Theta & \Theta \\ \Theta & \Theta & I & \Theta \\ \Theta & \mathbf{I} & \Theta & \Theta \\ mI & \Theta & \Theta & \Theta \end{bmatrix}, \tag{S2}$$

and

$$K = -\begin{bmatrix} I & \Theta & \Theta & \Theta \\ \Theta & I & \Theta & \Theta \\ \Theta & \Theta & \mathbf{B} & \Theta \\ \Theta & \Theta & \Theta & I \end{bmatrix}. \tag{S3}$$

Here, Θ is a 3×3 zero matrix, I is a 3×3 identity matrix, m is the mass per unit length, $\mathbf{I}$ is a 3×3 inertia matrix per unit length, and $\mathbf{B}$ is the 3×3 stiffness tensor. The 12×1 vector F is defined as

$$F = \begin{bmatrix} \omega \times \hat{t} - \kappa \times v \\ -\kappa \times \omega \\ -\left(\frac{\partial \mathbf{B}}{\partial s}\kappa - \frac{\partial (\mathbf{B}\kappa_o)}{\partial s}\right) + \omega \times \mathbf{I}\omega + f \times \hat{t} - \kappa \times \mathbf{B}\left(\kappa - \kappa_o\right) - Q_{body} \\ m\left(\omega \times v\right) - \kappa \times f - F_{body} \end{bmatrix}. \tag{S4}$$

Here, $\hat{t}$ is the tangent of the rod, F_{body} is a body force per unit length on the rod, and Q_{body} is a body moment per unit length on the rod. The matrices

M, K, and F in general are functions of the arclength coordinate s and time t. In addition, F may also be a function of the configuration of the rod (the position and orientation of the rod axis as a function of s). For example, a dependence on configuration arises from the inclusion of electrostatic forces.

We write boundary conditions (that in general are functions of time) in the following form

$$\begin{bmatrix} C_0 & C_L \end{bmatrix} \begin{bmatrix} Y_{s=0} \\ Y_{s=L} \end{bmatrix} = c. \tag{S5}$$

Here C_0 and C_L are 12×12 matrices and c is a 12×1 vector. The entries of C_0, C_L and c are chosen to describe the boundary conditions on the rod. To represent Dirichlet boundary conditions on v and ω at both ends, for example, the first 6 rows and columns of C_0 and the second 6 rows and first 6 columns of C_L would both form 6×6 identity matrices while the remaining entries would all be zeros. In this special case, the entries of c would be the prescribed values of v and ω at either end. In the accompanying paper, c is a function of the integral of ω following Eqs. 12-14.

To solve this system of equations we discretize it into N spatial gridpoints. We use j to index grid-points from 1 to N. We now write Eq. S1 for a point $(j - \frac{1}{2})$ halfway between grid-points $j-1$ and j

$$(M\dot{Y})_{j-\frac{1}{2}} + (KY_s)_{j-\frac{1}{2}} + (F)_{j-\frac{1}{2}} = 0. \tag{S6}$$

We approximate $(M\dot{Y})_{j-\frac{1}{2}}$ with

$$(M\dot{Y})_{j-\frac{1}{2}} = \frac{1}{2}(M_{j-1}\dot{Y}_{j-1} + M_j\dot{Y}_j), \tag{S7}$$

$(KY_s)_{j-\frac{1}{2}}$ with

$$(KY_s)_{j-\frac{1}{2}} = \frac{1}{2}(K_{j-1} + K_j)\frac{1}{\Delta s}(-Y_{j-1} + Y_j), \tag{S8}$$

and $(F)_{j-\frac{1}{2}}$ with

$$(F)_{j-\frac{1}{2}} = \frac{1}{2}(F_{j-1} + F_j). \tag{S9}$$

In Eq. S8 we introduce the spatial discretization length, Δs, to estimate the spatial derivative Y_s. Substituting Eqs. S7-S9 into Eq. S6, simplifying, and arranging into matrix form we have

$$\begin{bmatrix} M_{j-1} & M_j \end{bmatrix} \begin{bmatrix} \dot{Y}_{j-1} \\ \dot{Y}_j \end{bmatrix} + \begin{bmatrix} -(K_j + K_{j-1})\frac{1}{\Delta s} & (K_j + K_{j-1})\frac{1}{\Delta s} \end{bmatrix} \begin{bmatrix} Y_{j-1} \\ Y_j \end{bmatrix} + F_{j-1} + F_j = 0 \quad \text{(S10)}$$

Considering $1 \leq j \leq N$, we have $(N-1)$ equations like Eq. S10. For more compact notation, we now define the following vectors of dimension $12N \times 1$:

$$\bar{\dot{Y}} = \begin{bmatrix} \dot{Y}_1 \\ \dot{Y}_2 \\ \dot{Y}_3 \\ \vdots \\ \dot{Y}_N \end{bmatrix}, \quad \text{(S11)}$$

$$\bar{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_N \end{bmatrix}, \quad \text{(S12)}$$

and

$$\bar{F} = \begin{bmatrix} F_1 + F_2 \\ F_2 + F_3 \\ F_3 + F_4 \\ \cdots \\ F_{N-1} + F_N \\ c \end{bmatrix}. \quad \text{(S13)}$$

We further define the following matrices of dimension $12N \times 12N$:

$$\bar{M} = \begin{bmatrix} M_1 & M_2 & 0 & \cdots & 0 \\ 0 & M_2 & M_3 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & M_{N-1} & M_N \\ 0 & \cdots & & & 0 \end{bmatrix} \quad \text{(S14)}$$

and

$$\bar{K} = \frac{1}{\Delta s}\begin{bmatrix} -K_2 - K_1 & K_2 + K_1 & 0 & \cdots & 0 \\ 0 & -K_3 - K_2 & K_3 + K_2 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -K_N - K_{N-1} & K_N + K_{N-1} \\ \Delta s C_0 & 0 & \cdots & 0 & \Delta s C_L \end{bmatrix}. \tag{S15}$$

Now assembling Eq. S10 (for $1 \leq j \leq N$) and Eq. S5 we have

$$\bar{M}\dot{\bar{Y}} + \bar{K}\bar{Y} + \bar{F} = 0. \tag{S16}$$

Note that Eq. S16 is continuous in time. We now integrate it in time using the generalized alpha method (3); see also (1, 2).

$$(1-\alpha_m)\bar{M}\dot{\bar{Y}}^i + \alpha_m\bar{M}\dot{\bar{Y}}^{i+1} + (1-\alpha_k)\bar{K}\bar{Y}^i + \alpha_k\bar{K}\bar{Y}^{i-1} + (1-\alpha_k)\bar{F}^i + \alpha_k\bar{F}^{i-1} = 0 \tag{S17}$$

Here the superscript i is an index that denotes the discrete temporal grid-point. The variables α_m and α_k are numerical parameters for the generalized alpha method. Because $\bar{M}$ and $\bar{K}$ may themselves be functions of time we perform a weighted average following (2)

$$\bar{M} = (1-\alpha_m)\bar{M}^i + \alpha_m\bar{M}^{i-1}, \tag{S18}$$

$$\bar{K} = (1-\alpha_k)\bar{K}^i + \alpha_k\bar{K}^{i-1}. \tag{S19}$$

Now substituting Eq. S18 and Eq. S19 into Eq. S17 we have

$$\begin{aligned}(1-\alpha_m)\left[(1-\alpha_m)\bar{M}^i + \alpha_m\bar{M}^{i-1}\right]\dot{\bar{Y}}^i + \alpha_m\left[(1-\alpha_m)\bar{M}^i + \alpha_m\bar{M}^{i-1}\right]\dot{\bar{Y}}^{i-1} \\ + (1-\alpha_k)\left[(1-\alpha_k)\bar{K}^i + \alpha_k\bar{K}^{i-1}\right]\bar{Y}^i + \alpha_k\left[(1-\alpha_k)\bar{K}^i + \alpha_k\bar{K}^{i-1}\right]\bar{Y}^{i-1} \\ + (1-\alpha_k)\bar{F}^i + \alpha_k\bar{F}^{i-1} = 0.\end{aligned} \tag{S20}$$

Here

$$\bar{Y}^i = \bar{Y}^{i-1} + \Delta t\left[(1-\gamma)\dot{\bar{Y}}^{i-1} + \gamma\dot{\bar{Y}}^i\right] \tag{S21}$$

and therefore

$$\dot{\bar{Y}}^i = \frac{1}{\gamma}\left[\frac{1}{\Delta t}(\bar{Y}^i - \bar{Y}^{i-1}) - (1-\gamma)\dot{\bar{Y}}^{i-1}\right]. \tag{S22}$$

Here, γ is an additional numerical parameter of the generalized alpha method. The parameters α_m, α_k, and γ may all be chosen to adjust numerical dissipation and accuracy of the generalized alpha method. In the accompanying paper, we choose $\alpha_m = -0.5$, $\alpha_k = 0.0$, and $\gamma = 1.0$, which yields a second order accurate method with numerical damping of spurious high frequencies (2, 3). (In addition, in our simulations, we choose $\Delta s = 0.34$ nm and $\Delta t = 0.1$ ns.) Now substituting Eq. S22 into Eq. S20 we have

$$(1-\alpha_m)\left[(1-\alpha_m)\bar{M}^i + \alpha_m \bar{M}^{i-1}\right]\frac{1}{\gamma}\left[\frac{1}{\Delta t}(\bar{Y}^i - \bar{Y}^{i-1}) - (1-\gamma)\dot{\bar{Y}}^{i-1}\right]$$
$$+\alpha_m\left[(1-\alpha_m)\bar{M}^i + \alpha_m \bar{M}^{i-1}\right]\dot{\bar{Y}}^{i-1} + (1-\alpha_k)\left[(1-\alpha_k)\bar{K}^i + \alpha_k \bar{K}^{i-1}\right]\bar{Y}^i$$
$$+\alpha_k\left[(1-\alpha_k)\bar{K}^i + \alpha_k \bar{K}^{i-1}\right]\bar{Y}^{i-1} + (1-\alpha_k)\bar{F}^i + \alpha_k \bar{F}^{i-1} = 0. \quad \text{(S23)}$$

For convenience, we group terms for $\bar{Y}^i$, $\bar{Y}^{i-1}$, and $\dot{\bar{Y}}^{i-1}$ and define their coefficients as A_1, A_2, and A_3 respectively. Specifically, we define

$$A_1 = (1-\alpha_m)\left[(1-\alpha_m)\bar{M}^i + \alpha_m \bar{M}^{i-1}\right]\frac{1}{\gamma\Delta t} + (1-\alpha_k)\left[(1-\alpha_k)\bar{K}^i + \alpha_k \bar{K}^{i-1}\right], \quad \text{(S24)}$$

$$A_2 = -(1-\alpha_m)\left[(1-\alpha_m)\bar{M}^i + \alpha_m \bar{M}^{i-1}\right]\frac{1}{\gamma\Delta t} + \alpha_k\left[(1-\alpha_k)\bar{K}^i + \alpha_k \bar{K}^{i-1}\right], \quad \text{(S25)}$$

and

$$A_3 = (1-\alpha_m)\left[(1-\alpha_m)\bar{M}^i + \alpha_m \bar{M}^{i-1}\right]\frac{1}{\gamma}[-(1-\gamma)] + \alpha_m\left[(1-\alpha_m)\bar{M}^i + \alpha_m \bar{M}^{i-1}\right] \quad \text{(S26)}$$

such that Eq. S23 becomes

$$A_1\bar{Y}^i + A_2\bar{Y}^{i-1} + A_3\dot{\bar{Y}}^{i-1} + (1-\alpha_k)\bar{F}^i + \alpha_k \bar{F}^{i-1} = 0. \quad \text{(S27)}$$

Note that $\bar{F}^i$ may in general be a function of $\bar{Y}^i$.

To find the solution $(\bar{Y}^i)$ of this system of nonlinear equations we use Newton-Raphson iterations. We index our iterations with the left superscript l. To initiate the iterations we define

$${}^0\bar{Y}^i = \bar{Y}^{i-1} + \Delta t\dot{\bar{Y}}^{i-1}. \quad \text{(S28)}$$

Newton-Raphson iterations are based upon linearizing Eq. S27 about ${}^l\bar{Y}^i$, the l^{th} estimate of $\bar{Y}^i$. To do so we define the Jacobian

$${}^l\bar{J}^i = (1-\alpha_k)\left.\frac{\partial \bar{F}^i}{\partial \bar{Y}^i}\right|_{{}^l\bar{Y}^i} + A_1. \quad \text{(S29)}$$

Now we iteratively solve the the following equation for ${}^{l}\bar{Y}^{i}$:

$$ {}^{l}\bar{Y}^{i} = {}^{l-1}\bar{Y}^{i} - \left({}^{l-1}\bar{J}^{i}\right)^{-1} \left(A_1 {}^{l-1}\bar{Y}^{i} + A_2 \bar{Y}^{i-1} + A_3 \bar{\dot{Y}}^{i-1} + (1-\alpha_k)^{l-1}\bar{F}^{i} + \alpha_k \bar{F}^{i-1} \right) \tag{S30} $$

until some error tolerance is satisfied; for example, when $||{}^{l}\bar{Y}^{i} - {}^{l-1}\bar{Y}^{i}||$ is sufficiently small.

To aid in the calculation of the Jacobian, we define G^i_j as

$$ G^i = \frac{\partial F^i_j}{\partial Y^i_j}. \tag{S31} $$

Therefore, in the special case when F_{body} and Q_{body} are zero vectors and $\mathbf{B}$ is constant we have

$$ G^i_j = \begin{bmatrix} -\tilde{\kappa}^i_j & -\tilde{\hat{t}} & \tilde{v}^i_j & \Theta \\ \Theta & -\tilde{\kappa}^i_j & \tilde{\omega}^i_j & \Theta \\ \Theta & -\widetilde{\left(\mathbf{I}\omega^i_j\right)} + \tilde{\omega}^i_j \mathbf{I} & \widetilde{\left(\mathbf{B}(\kappa^i_j - \kappa_o)\right)} - \tilde{\kappa}^i_j \mathbf{B} & -\tilde{\hat{t}} \\ m\tilde{\omega}^i_j & -m\tilde{v}^i_j & \tilde{f}^i_j & -\tilde{\kappa}^i_j \end{bmatrix}. \tag{S32} $$

Here, the $\tilde{()}$ operator generates a skew symmetric matrix, such that for any pair of 3-vectors a and b, $a \times b = \tilde{a}b$. Now we can define $\bar{G}^i$ (similar to Eq. S18 and Eq. S19) as

$$ \bar{G}^i = \frac{\partial \bar{F}^i}{\partial \bar{Y}^i} = \begin{bmatrix} G^i_1 & G^i_2 & 0 & \cdots & 0 \\ 0 & G^i_2 & G^i_3 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & G^i_{N-1} & G^i_N \\ \frac{\partial c^i}{\partial Y^i_1} & 0 & \cdots & 0 & \frac{\partial c^i}{\partial Y^i_N} \end{bmatrix}. \tag{S33} $$

Finally, this expression is evaluated at ${}^{l}\bar{Y}^{i}$ and substituted into Eq. S29.

During the algorithm we also compute the position R and orientation of the rod as a function of the arclength coordinate s. (Our algorithm here is based on integrating curvature, however we could instead base it on integrating velocity and angular velocity.) To do so we first define

$$ \theta^i_j = \frac{\Delta s}{2} \left\{ \kappa^i_{j-1} + \kappa^i_j \right\}. \tag{S34} $$

Next we calculate

$$\mathbf{L}_j^i = \exp\left\{-\tilde{\theta}_j^i\right\}\mathbf{L}_{j-1}^i, \tag{S35}$$

where exp{} is the matrix exponential operator and $\mathbf{L}_j^i$ is the direction cosine matrix

$$\mathbf{L}_j^i = \begin{bmatrix} \mathbf{a}_{j1}^i \cdot \mathbf{e}_1 & \mathbf{a}_{j1}^i \cdot \mathbf{e}_2 & \mathbf{a}_{j1}^i \cdot \mathbf{e}_3 \\ \mathbf{a}_{j2}^i \cdot \mathbf{e}_1 & \mathbf{a}_{p2}^i \cdot \mathbf{e}_2 & \mathbf{a}_{j2}^i \cdot \mathbf{e}_3 \\ \mathbf{a}_{j3}^i \cdot \mathbf{e}_1 & \mathbf{a}_{j3}^i \cdot \mathbf{e}_2 & \mathbf{a}_{j3}^i \cdot \mathbf{e}_3 \end{bmatrix}. \tag{S36}$$

Here, $\mathbf{a}_j^i$ is a body fixed reference frame at spatial gridpoint j along the arclength. Its basis vectors are $\mathbf{a}_{j1}^i$, $\mathbf{a}_{j2}^i$, and $\mathbf{a}_{j3}^i$, with $\mathbf{a}_{j3}^i = \hat{t}_j^i$. In addition, the basis vectors for the inertial reference frame are denoted $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$. Finally, we compute R_j^i

$$R_j^i = R_{j-1}^i + \frac{\Delta s}{2}\left\{\hat{t}_{j-1}^i + \hat{t}_j^i\right\}. \tag{S37}$$

In general, R_1^i and $\mathbf{L}_1^i$ at the first end of the rod ($s = 0$) could vary with time; in this case we integrate velocity and angular velocity following the development to determine R_1^i and $\mathbf{L}_1^i$.

To summarize our algorithm, we present an outline below.

- Set $i = 0$
- Define initial conditions $\bar{Y}^i$ and $\dot{\bar{Y}}^i$
- Loop in time with index i
  - Update i using $i = i + 1$
  - Reset Newton-Raphson index, $l = 0$
  - Calculate ${}^0\bar{Y}^i$, the first estimate of $\bar{Y}^i$, by using Eq. S28
  - Loop for Newton-Raphson iterations
    - Update l using $l = l + 1$
    - Calculate ${}^{l-1}R_j^i$ and ${}^{l-1}\mathbf{L}_j^i$ for all j using Eqs. S34-S37
    - Update A_1, A_2, and A_3 when necessary (for example if the boundary conditions change)
    - Calculate ${}^{l-1}\bar{F}^i$
    - Calculate ${}^{l-1}\bar{J}^i$ using Eq. S29

    * Solve Eq. S30 for ${}^{l}\bar{Y}^{i}$
    * Exit loop if ${}^{l}\bar{Y}^{i}$ is sufficiently converged, otherwise continue Newton-Raphson loop

  - Set $\bar{Y}^{i} = {}^{l}\bar{Y}^{i}$
  - Calculate $\dot{\bar{Y}}^{i}$ from Eq. S22
  - Calculate R_j^i and $\mathbf{L}_j^i$ for all j using Eq. S34-S37
  - Exit loop when integrated through time sufficiently long

- Algorithm complete, post-process output

References

1. Goyal, S., N. C. Perkins, and C. L. Lee. 2005. Nonlinear dynamics and loop formation in Kirchhoff rods with implications to the mechanics of DNA and cables. *J. Comput. Phys.* 209:371–389.

2. Gobat, J. I., and M. A. Grosenbaugh. 2001. Application of the generalized-alpha method to the time integration of the cable dynamics equations. *Comput. Method. Appl. M.* 190:4817–4829.

3. Chung, J., and G. M. Hulbert. 1993. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-alpha method. *J. Appl. Mech.* 60:371–375.