Appendix: Numerical Method

Here, we fully elucidate the numerical implementation of the model described in this article. First, we describe the method used for simulating the kinetochore, and then we describe the method used to simulate the rest of the chromosome. All physical parameters not defined here are defined in *Model*.

The chromosome is discretized as explained in *Model*. As mentioned, the central segment of the chromosome $\mathbf{x}_0$ is considered to be connected the the kinetochore via a stiff spring, thus yielding the following relation for the effective 1D diffusion constant of the complex:

$$D_{\text{eff}} = \frac{D_{\text{kinetochore}} D_{\mathbf{x}_0}}{D_{\text{kinetochore}} + D_{\mathbf{x}_0}}, \quad [1]$$

where

$$D_{\mathbf{x}_0} = \frac{k_{\text{B}}T}{(\Delta s)\zeta}. \quad [2]$$

This complex then diffuses in the potential defined by the interactions of the kinetochore with the microtubule and the elastic interactions of the central segment with the rest of the chromosome.

The nature of the Brownian ratchet potential is inherently discontinuous. Thus, one cannot use a standard forward Euler method, because the instantaneous force caused by the potential drop ΔV is nil almost everywhere (and infinite almost nowhere). Instead, we use the method that allows for for discontinuous potentials (1). This method discretizes the motion of the kinetochore in space, yielding jump rates for the motor along a one-dimensional lattice given the effective diffusion coefficient of the kinetochore and the potential acting upon the kinetochore. This simulation technique has been shown to be strongly convergent (1).

Here, the lattice sites are at positions $\left(m + \frac{1}{2}\right)(\Delta x), m = \ldots, -1, 0, 1, \ldots$ along the x-axis, where Δx is the lattice width, chosen such that L (the period of the Brownian ratchet potential) is some integer multiple of Δx. Note that the potential jumps of $-\Delta V$ never occur at the lattice sites themselves, but rather occur halfway between (some) adjacent lattice sites. Recall that $\mathbf{x}_0 = (x_0, 0, 0)$ is the position of the kinetochore. Therefore x_0 must coincide with the position of one of the lattice sites, and so x_0 changes in steps of $\pm\Delta x$. As it does so, the whole configuration of the chromosome changes from $\overline{\mathbf{x}}$ to $\overline{\mathbf{x}} \pm (\Delta x)\mathbf{e}_0^x$, where $\mathbf{e}_0^x$ is a unit vector in the same higher dimensional space as $\overline{\mathbf{x}}$ with components $\left(e_0^x\right)_j^i = \delta_{j0}\delta_{i1}$. Thus, the configuration $\overline{\mathbf{x}} \pm (\Delta x)\mathbf{e}_0^x$ is the same as $\overline{\mathbf{x}}$, except that the kinetochore has been moved forward or backward along the microtubule by an amount Δx (not to be confused with the length L of a tubulin heterodimer: Δx is a numerical parameter that is much smaller than L).

The rate constants (probabilities per unit time) for jumping forward or backward along the compuational lattice are given by

$$F(\overline{\mathbf{x}}) = \frac{D_{\text{eff}}}{(\Delta x)^2} \frac{\alpha_{\text{F}}}{e^{\alpha_{\text{F}}} - 1} \quad [3]$$

$$B(\overline{\mathbf{x}}) = \frac{D_{\text{eff}}}{(\Delta x)^2} \frac{\alpha_{\text{B}}}{e^{\alpha_{\text{B}}} - 1}, \quad [4]$$

where

$$\alpha_{\rm F} = \frac{1}{k_{\rm B}T}\left(E[\overline{\mathbf{x}} + (\Delta x)\mathbf{e}_0^x] - E[\overline{\mathbf{x}}] + U(x_0 + \Delta x) - U(x_0)\right) \quad [5]$$

$$\alpha_{\rm B} = \frac{1}{k_{\rm B}T}\left(E[\overline{\mathbf{x}} - (\Delta x)\mathbf{e}_0^x] - E[\overline{\mathbf{x}}] + U(x_0 - \Delta x) - U(x_0)\right), \quad [6]$$

where $E(\overline{\mathbf{x}})$ is the elastic energy of the chromosome configuration $\overline{\mathbf{x}}$, and $U(x)$ is the discontinuous tilted periodic potential characterizing the imperfect Brownian ratchet motor that pulls on the chromosome. Both E and U are defined in the text.

The procedure for choosing the waiting time $\Delta\tau$ for the jump to occur, and the direction of the jump when it does occur, is as follows (2). First, a random time increment $\Delta\tau$ is chosen from an exponential distribution with mean value $(F(\overline{\mathbf{x}}) + B(\overline{\mathbf{x}}))^{-1}$. This represents the waiting time until the next jump event occurs, be it either forward or backward along the lattice. Next, another random number r is generated, uniformly distributed in the interval $[0, 1]$. Then, if $r < F(\overline{\mathbf{x}})/(F(\overline{\mathbf{x}}) + B(\overline{\mathbf{x}}))$, q is set to $x_0 + \Delta x$; otherwise, q is set to $x_0 - \Delta x$. The outputs of this procedure are $\Delta\tau$, a randomly chosen timestep duration, and $(q, 0, 0)$, the position of the kinetochore at the end of the time step. The procedure used to update the configuration of the rest of the chromosome over the time interval $\Delta\tau$ will be described next.

To update the configuration of the rest of the chromosome, we use the Euler-Maruyama method (3) for the solution of systems of stochastic ordinary differential equations. There are, of course, stiffness constraints on the time step, which are especially severe because of the fourth-order differences in the computation of the force density of an elastic rod. This poses a difficulty because the time increment chosen in the manner described above is exponentially distributed, allowing for the possibility of arbitrarily large time increments. This difficulty is circumvented by first empirically determining a maximal time step $\Delta\tau^*$ at which the simulation is stable, and then dividing the chosen time increment $\Delta\tau$ into p equal parts, where p is set equal to $\lceil \frac{\Delta\tau}{\Delta\tau^*} \rceil$. This yields a timestep $\Delta t = \Delta\tau/p$. The following Euler update scheme is then executed p times:

$$x_j^i(t + \Delta t) = x_j^i(t) + \frac{\Delta t}{\zeta}(F^{\rm elastic})_j^i(t) + \sqrt{\frac{2k_{\rm B}T(\Delta t)}{\zeta(\Delta s)}}\, w_j^i(t),\ |j| < N \quad [7]$$

$$x_j^i(t + \Delta t) = x_j^i(t) + \frac{\Delta t}{\zeta}(F^{\rm elastic})_j^i(t) + \sqrt{\frac{2k_{\rm B}T(\Delta t)}{\zeta(\Delta s/2)}}\, w_j^i(t),\ |j| = N, \quad [8]$$

where i is the index of the spatial component, j is the segment (node) index, $(F^{\rm elastic})_j^i(t)$ refers to the elastic force density at time t, and $w_j^i(t)$ is chosen (independently for different values of (j, i, t)) from a Gaussian distribution with mean 0 and unit variance. In practice, the lattice parameter Δx is chosen small enough that the probability of more than one Euler step being needed to traverse any one of the randomly chosen time increments $\Delta\tau$ is very low. After p Euler steps have been taken, $\mathbf{x}_0$ is set to $(q, 0, 0)$, where q is the final position of the kinetochore that was chosen in the first part of the algorithm, in which a random choice was made whether to move one lattice step forward or one lattice step backward.

Note that in the Euler-Maruyama step(s) as described above, only the elastic force density is used, the motor potential $U(x)$ is ignored. Moreover, the kinetochore is treated like any

other node of the chromosome during this part of the algorithm. In particular, it moves freely in 3D space even though it will be put back on the computational lattice at the predetermined (randomly chosen) position $(q, 0, 0)$ as soon as the p Euler-Maruyama steps have been completed. An alternative that may occur to the reader is to freeze the position of the kinetochore during the Euler-Maruyama step(s), holding it at the lattice point where it was at the beginning of the timestep while the p Euler-Maruyama steps are being carried out for the rest of the chromosome, and then finally moving the kinetochore to $(q, 0, 0)$ at the end of the time step. We remark that these two possible schemes are identical for $p = 1$, and only differ for $p > 1$. In practice, we choose parameters so that $p > 1$ is a rare event, and thus there is little practical difference between these two alternatives.

One must, of course, demonstrate convergence of this numerical scheme. One difficulty in checking for convergence in a stochastic numerical scheme is that result of a simulation is different every time. To circumvent this difficulty, we chose to look for convergence of steady chromosome velocity as measured over the duration of a long simulation run, thus averaging out the variability inherent in the stochastic method.

In this manner, we checked for convergence of the steady chromosome velocity with respect to Δx and Δs. We ran simulations by using parameters for the long chromosomes ($s_0 = 1.5\mu$m) for 1.92×10^{10} iterations with $\Delta\phi = 0$. This choice of $\Delta\phi$ is, of course, different from the value of 8 $k_{\mathrm{B}}T$ used in the text. However, at 8 $k_{\mathrm{B}}T$, the chromosomes moved too slowly to allow one to obtain statistically significant values for steady chromosome velocity within a reasonable amount of computational time. We expect the convergence properties to be similar regardless of the choice of $\Delta\phi$, though, as $\Delta\phi$ does not introduce any significant discontinuities into the motor potential.

For Δx, velocities were computed for $N = 25$ over a range of Δx values. For Δs, velocities were computed for $\Delta x = L/512$ over a range of Δs values (the ranges are shown in Figs. 3 and 4, respectively). In general, determining the convergence rate from these data presents a challenge because there is no theoretical value with which to compare our computed values to find an error. Usually, this is circumvented by considering the "error" at each data point be the difference between successive runs; for instance, the error in the computed velocity for $N = 6$ is the difference between the velocity at $N = 6$ and $N = 12$, the error for $N = 12$ is the difference between $N = 12$ and $N = 24$, etc. However, this approach is limited in that it reduces the number of data points by one and also may not accurately reflect the error. To circumvent these difficulties, we took the approach of considering the actual theoretical value of the velocity, to which our data is then compared, to be a variable. Then the problem becomes a nonlinear optimization problem: one must find the theoretical value such that it makes a plot of the error versus Δx (or Δs) as linear as possible, on a log-log scale. To be more precise, for a prospective theoretical velocity v^*, we define the log of the errors $l_{err}(\Delta x)$ by:

$$l_{err}(\Delta x) \quad = \quad \log(|v(\Delta x) - v^*|), \qquad \textbf{[9]}$$

where $v(\Delta x)$ is the computed velocity for the given value of Δx. The optimization problem is then to find v^* that maximizes the correlation coefficient between $\log(\Delta x)$ and $l_{err}(\Delta x)$ over the range of Δx for which velocities were computed. This procedure was done to find convergence rates for both Δx and Δs; the results are shown in Figs. 3 and 4, respectively.

The rates of convergence are 1.76 for Δx and 2.61 for Δs, with corresponding theoretical velocities v^* of 8.742×10^{-7}m/s and 8.777×10^{-7} m/s.

These rates are reasonable given the nature of the numerical scheme. For Δx, the nature of the scheme is such that $\langle \Delta\tau \rangle \propto (\Delta x)^2$, yielding a convergence rate of 0.88 with respect to $\langle \Delta\tau \rangle$. This is higher than the 0.5 rate of convergence one generally finds for an Euler-Maruyama scheme. This may be caused by the distributed rather than deterministic nature of $\Delta\tau$.

There is a slight subtlety when trying to understand the rate of convergence of Δs. This arises from the manner in which the jump transition rates are chosen in Eqs. **3** and **4**. Because D_{eff} depends on Δs, as indicated in Eqs. **1** and **2**, varying Δs will also effectively cause a decrease in $\langle \Delta\tau \rangle$. Thus, the convergence rate should actually slighly higher than what one might expect solely based on the nature of the spatial discretization of the chromosome. This is indeed the case, as we have obtained a convergence rate of 2.61, which is more than one would expect given our use of a second-order discretization scheme. The fact that the additional convergence gained is slightly lower than the rate of convergence with respect to $\Delta\tau$ may be because D_{eff} does not depend on Δs in a linear fashion.

References

1. Wang, H. Y., Peskin, C. S. & Elston, T. C. (2003) *J. Theor. Biol.* **221**, 491–511.

2. Gillespie, D. T. (1977) *J. Phys. Chem.* **81**, 2340–2361.

3. Kloeden, P. & Platen, E. (1993) *Numerical Solution of Stochastic Differential Equations* (Springer, New York).