Biophysical Journal
Article

Brownian dynamics simulations of mesoscale chromatin fibers

Zilong Li,[1] Stephanie Portillo-Ledesma,[1] and Tamar Schlick[1,2,3,4,*]
[1]Department of Chemistry, New York University, New York, New York; [2]Courant Institute of Mathematical Sciences, New York University, New York, New York; [3]New York University-East China Normal University Center for Computational Chemistry, New York University Shanghai, Shanghai, China; and [4]Simons Center for Computational Physical Chemistry, New York University, New York, New York

ABSTRACT The relationship between chromatin architecture and function defines a central problem in biology and medicine. Many computational chromatin models with atomic, coarse-grained, mesoscale, and polymer resolution have been used to shed light onto the mechanisms that dictate genome folding and regulation of gene expression. The associated simulation techniques range from Monte Carlo to molecular, Brownian, and Langevin dynamics. Here, we present an efficient Compute Unified Device Architecture (CUDA) implementation of Brownian dynamics (BD) to simulate chromatin fibers at the nucleosome resolution with our chromatin mesoscale model. With the CUDA implementation for computer architectures with graphic processing units (GPUs), we significantly accelerate compute-intensive hydrodynamic tensor calculations in the BD simulations by massive parallelization, boosting the performance a hundred-fold compared with central processing unit calculations. We validate our BD simulation approach by reproducing experimental trends on fiber diffusion and structure as a function of salt, linker histone binding, and histone-tail composition, as well as Monte Carlo equilibrium sampling results. Our approach proves to be physically accurate with performance that makes feasible the study of chromatin fibers in the range of kb or hundreds of nucleosomes (small gene). Such simulations are essential to advance the study of biological processes such as gene regulation and aberrant genome-structure related diseases.

SIGNIFICANCE Brownian dynamics simulations of chromatin fibers can provide valuable insight into genome-related processes such as chromatin folding and epigenetic regulation. By using a Compute Unified Device Architecture programming platform on graphics processing units, we accelerate the expensive Brownian dynamics force calculations including hydrodynamic interactions. Such an implementation opens the possibility to study kb- to Mb-range chromatin fibers containing explicit histone tails and linker histones at nucleosome resolution.

INTRODUCTION

Chromatin, a complex of DNA and histone proteins, has an essential role in gene expression (1). Yet, its three-dimensional (3D) structure and folding mechanisms to condense meters of eukaryotic DNA into a micrometer nucleus remain obscure (2). The regulation of these mechanisms by posttranslational modifications, binding of linker histone (LH) or other proteins, or the wrapping/unwrapping of nucleosomes is essential to control chromatin architecture and hence DNA accessibility and genome integrity (3).

With state-of-the-art experimental techniques, such as chromosome conformation capture, superresolution microscopy, and high-throughput sequencing, as well as computational modeling, we have changed our view of the 3D genome organization from a regular ordered folding to a heterogeneous ensemble of folds (4). Experiments have shown that chromosomes are hierarchically organized, and the genome is segregated into transcriptionally active and inactive compartments. Inside these compartments, chromatin establishes self-interactions, creating substructures called topological-associated domains, which often arise from the formation of loops mediated by the CTCF-cohesin complex.

Computational efforts have similarly contributed to our understanding of chromatin architecture and its relationship to function (5–9). These studies have provided valuable information on chromatin's ionic dependence, the role of histone tails and LH, DNA electrostatics and sequence dependence, the effect of LH and nucleosome depletion or posttranslational modifications, gene silencing and folding

Submitted May 31, 2022, and accepted for publication September 13, 2022.
*Correspondence: schlick@nyu.edu
Editor: Alberto Perez.
https://doi.org/10.1016/j.bpj.2022.09.013

mechanisms, loop formation, and regulation of topological-associated domains, among others.

The approaches used in chromatin modeling span a huge range, from all-atom simulations of single nucleosomes to coarse-grained and mesoscale simulations of oligonucleosome fibers and gene systems and polymer models of whole chromosomes, with decreasing resolution as system size increases (see recent reviews in (7,10–17)). While all-atom, coarse-grained, and mesoscale models are constructed based on the atomic structures of chromatin components, such as nucleosomes and LHs, some polymer models are data driven and use experimental information, usually from chromosome conformation capture, to describe chromatin architecture (6,14).

These multiscale modeling approaches for chromatin rely on various simulation techniques for conformational sampling like molecular dynamics (MD) (e.g., (18,19,20)), Monte Carlo (MC) (e.g., (21,22)), Brownian dynamics (BD) (e.g., (23) (24)), and Langevin dynamics (e.g., (25) (26)). In particular, BD is suitable for simulating physical systems containing large particles where diffusive forces are significant compared with internal forces. The bottleneck in BD is the frequent calculation of the hydrodynamic interactions between BD particles, a matrix D, which has an $O(n^2)$ complexity with system size n (n particles). Moreover, to calculate the Brownian random force, whose properties depend on D, a Cholesky factorization of the dense matrix D is required, and this task has an $O(n^3)$ complexity. This limits BD system size (27).

With the emerging Compute Unified Device Architecture (CUDA) for simulations on graphics processing units (GPUs), the calculation of D can be parallelized on as many threads (basic computing elements) as the number of interactions between BD particles, reducing the time complexity to $O(1)$. The corresponding Cholesky factorization can be parallelized to reduce its time complexity to $O(n)$. As has been proven for other molecular modeling applications, such as MD simulations (28) and quantum chemistry (29), the use of GPUs allows significant speed ups and, hence, simulation of larger system sizes over longer simulation times (30,31).

In our early chromatin modeling studies, Beard and Schlick developed and applied BD to investigate the structure and dynamics of small oligonucleosome fibers that did not include LH and flexible histone tails (32). When we incorporated flexible histone tails into the mesoscale model to investigate their role in chromatin folding (33), MC simulations became more efficient for equilibrium sampling of large systems (34–38).

While MC has allowed us to study numerous important structural features of chromatin fibers and gene systems (e.g., (39–44)), BD introduces an exciting dynamic component to study the evolution of chromatin structure, transitions, and rearrangements. Here, we present the implementation of BD simulations of our mesoscale chromatin model with CUDA for calculation on GPUs and examine its performance using different metrics. We then validate the strategy by reproducing experimentally determined properties of different oligonucleosome systems, as well as previous MC results.

Overall, our implementation of BD to simulate mesoscale chromatin fibers reproduces experimental trends as MC and demonstrates exceptional computational performance, opening the possibility to study chromatin fibers of several hundred nucleosomes in a reasonable computer time. Such studies, in turn, could help reveal the mechanisms by which chromatin is compacted inside the cell nucleus and the impact of protein regulators on its folding.

MATERIALS AND METHODS

Chromatin mesoscale model

Our mesoscale model of chromatin has evolved for over 20 years, with the current model considering flexible histone tails (45) and their acetylation (46,47), two LH variants (H1E and H1C) with several binding modes (on versus off dyad) (48,49), and nonuniform linker DNA lengths (36), among other features (7).

The chromatin elements are coarse grained at different levels of resolution. The nucleosome cores are treated as charged disks according to the atomistic core particle, whereas linker DNA, histone tails, and LHs are treated as beads (Fig. 1).

The nucleosome core with 300 point charges distributed on its irregular surface calculated by the DiSCO algorithm is designed to reasonably approximate the electric field of the atomistic core particle by a Debye-Hückel (screened electrostatics) approximation (50). An optimization is performed once the charges are distributed to determine the approximation error. DiSCO assigns charges as follows: 1) a distance vector is defined from each atom to the center of the molecule, and its associated surface area is defined as a surface element; 2) multiple rounds of smoothing are performed; 3) surface points are distributed evenly at a

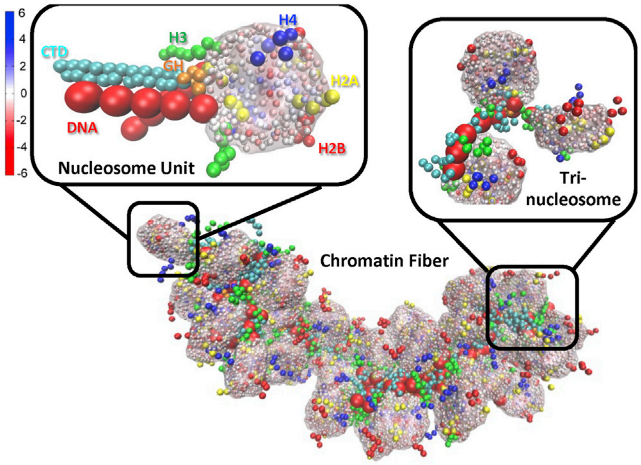

FIGURE 1 Chromatin mesoscale model. 50-nucleosome chromatin fiber with the enlarged basic unit (chromatosome) showing each element and the enlarged tri-nucleosome showing the connection between nucleosomes by linker DNA. Linker DNA is shown as red beads; histone tails as green (H3), yellow (H2A), red (H2B), and blue (H4) beads; LHs are shown as orange (globular head) and cyan (C-terminal domain) beads; and nucleosome cores are shown with their distributed charge beads. To see this figure in color, go online.

density of 4.0 per Å^2; and 4) the overall surface points are divided into 300 groups with representative surface charges assigned. The number 300 was determined to be a good balance between computational cost and electric field approximation. Linker DNAs are treated with a combined worm-like chain and bead model (51) based on the charged colloidal cylinder approach derived by Stigter (52). Histone tails (53) and LHs (48,49) are coarse grained as five residues per bead with the Levitt-Warshel united-atom protein model.

The water solvent surrounding the chromatin fibers is modeled with a continuum of dielectric constant of 80. The screening effect of monovalent salt on electrostatic interactions within the chromatin fiber is treated by the Debye-Hückel electrostatic potential. Thus, electrostatic interactions between all charged beads are screened depending on the salt concentration.

The chromatin model can be considered as a ball/spring system. In particular, linker DNAs connect nucleosome cores, and histone tails and LHs are attached to the nucleosome cores. Each bead can move freely during the simulation, except for the 300 charge beads within each nucleosome core, one fixed tail bead per histone tail, and the six LH globular head beads, which move together with the nucleosome core. Hence, the mesoscale chromatin model has flexible linker DNA, histone tails, LHs, and rigid nucleosome cores.

The total potential energy function of the model includes stretching, bending, and twisting terms for the linker DNA (E_S, E_B, E_T), stretching and bending terms for histone tails (E_{tS}, E_{tB}) and LHs (E_{lhS}, E_{lhB}), and excluded volume (E_V) and electrostatic (E_C) terms for all beads as

$$E(\mathbf{r}) = E_S + E_B + E_T + E_{tS} + E_{tB} + E_{lhS} + E_{lhB} + E_C + E_V, \tag{1}$$

where **r** is the collective position vector.

The forces (**F**) on the system are defined as the negative gradient of the potential energy as

$$\mathbf{F}_i = -\nabla_{r_i} E. \tag{2}$$

The torques (τ) due to the forces (**F**) are calculated by the equation

$$\tau = \mathbf{r} \times \mathbf{F}. \tag{3}$$

The supporting material provides further details on parameters and connections between elements, energy terms, and force calculations.

Simulation

A commonly used BD algorithm was first proposed in 1978 (54) based on the Langevin equation and then was improved in 1989 (55) with a second-order algorithm based on the Runge-Kutta method. The latter overcomes an issue associated with the first-order approximation: when the time step is small, the procedure can be inefficient and, when large, unstable. In 2001, Beard and Schlick (32) further modified the second-order BD algorithm to include rotational motions and to evaluate the dense matrix **D** less frequently. The derived second-order translational and rotational equations are given by

$$\text{Translation}:$$

$$\mathbf{r}^{n+1,*} = \mathbf{r}^n + \frac{\Delta t}{k_B T}\mathbf{D}(\mathbf{r}^n)\mathbf{F}^n + \mathbf{R}^n, \qquad \text{first–order}$$

$$\mathbf{r}^{n+1} = \mathbf{r}^n + \frac{\Delta t}{2k_B T}\mathbf{D}(\mathbf{r}^n)\left(\mathbf{F}^n + \mathbf{F}^{n+1,*}\right) + \mathbf{R}^n, \quad \text{second–order}$$

(4a) and (4b)

$$\text{Rotation}: \begin{array}{lll} \Delta\Omega_i^{n,*} = \dfrac{\Delta t}{\xi_i}\left(\tau_i^n + \omega_i^n\right), & & \text{first–order} \\ \Delta\Omega_i^n = \dfrac{\Delta t}{\xi_i}\left(\dfrac{\tau_i^n + \tau_i^{n+1,*}}{2} + \omega_i^n\right), & & \text{second–order} \end{array}$$

(5a) and (5b)

where **r** is the position vector, Δt is the time step, k_B is the Boltzmann's constant, T is the absolute temperature, **D** is the diffusion tensor, **F** is the force, τ is the torque, ξ_i is the rotational friction coefficient, and **R** and ω_i are the stochastic terms (32,54), which are Gaussian-distributed random vectors with means of 0 and variance-covariance of

$$\begin{array}{l} \left\langle \mathbf{R}^n(\Delta t)\left(\mathbf{R}^m(\Delta t)\right)^{\mathrm{T}}\right\rangle = 2\mathbf{D}^n \Delta t \delta_{nm} \\ \left\langle \omega_i^n(\Delta t)\omega_i^m(\Delta t)\right\rangle = 2k_B T \xi_i \delta_{nm} \end{array}. \qquad \text{(6a) and (6b)}$$

The Cholesky approach is used to calculate the vector **R** from the relationship in Eq. 6a. The Cholesky decomposition of the diffusion tensor **D** is determined by $\mathbf{D} = \mathbf{LL}^T$, where **L** is a lower triangular matrix and each element in **L** is given by

$$l_{ij} = \left\{ \begin{array}{ll} \left(D_{ii} - \sum_{k=1}^{i-1} l_{ik}^2\right)^{\frac{1}{2}}, & \text{if } i = j \\ \left(D_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}\right) \Big/ s_{jj}, & \text{if } i > j \\ 0, & \text{if } i < j \end{array} , \right\}. \tag{7}$$

The terms **D**, **F**, and **R** are derived from three types of forces. **D** is derived from the friction force, **F** is a systematic force that includes the interaction between Brownian particles and any external force, and **R** is derived from a random force and determined by the Cholesky decomposition of the diffusion tensor **D**.

Following the first-order estimate of the translation and rotation from Eq. 4a at time $(n+1)\Delta t$, namely $\Delta\mathbf{r}^{n,*} = \frac{\Delta t}{k_B T}\mathbf{D}(\mathbf{r}^n)\mathbf{F}^n + \mathbf{R}^n$ and $\Delta\Omega_i^{n,*} = \frac{\Delta t}{\xi_i}(\tau_i^n + \omega_i^n)$, we calculate the forces $\mathbf{F}^{n+1,*}$ and torques $\tau_i^{n+1,*}$ at the end of the $n+1$ time step and use them to construct an explicit second-order update as shown in Eq. 4b.

The calculation of forces and torques are described in the supporting material. Below, we show the calculation of the diffusion tensor.

Diffusion tensor

There are two types of diffusion tensors introduced in (54), the Oseen and Rotne-Prager tensors. Our approach uses the Rotne-Prager tensor because in the BD simulation, the diffusion tensor needs to be a positive definite matrix to apply the Cholesky decomposition. The Oseen tensor becomes a nonpositive definite when the separation between the particles is small. The Rotne-Prager tensor is given by

$$\left\{ \mathbf{D}_{ij} = \begin{array}{ll} \dfrac{k_B T}{6\pi\eta a_i}\mathbf{I}, & \text{for } i = j \\ \dfrac{k_B T}{8\pi\eta r_{i,j}}\left[\left(\mathbf{I} + \dfrac{\mathbf{r}_{i,j}\mathbf{r}_{i,j}}{r_{i,j}^2}\right) + \dfrac{\left(a_i^2 + a_j^2\right)}{r_{i,j}^2}\left(\dfrac{1}{3}\mathbf{I} - \dfrac{\mathbf{r}_{i,j}\mathbf{r}_{i,j}}{r_{i,j}^2}\right)\right], & \text{for } i \neq j \end{array} , \right\}, \tag{8}$$

where η is the solvent viscosity and a is the sphere radius of the particles.

Eq. 8 is for the two nonoverlapping particles, i,j. For overlapping particles, we use

$$\mathbf{D}_{ij} = \frac{k_B T}{6\pi\eta a_{eff}}\left[\left(1-\frac{9}{32}\frac{r_{i,j}}{a_{eff}}\right)\mathbf{I}+\frac{3}{32}\frac{\mathbf{r}_{i,j}\mathbf{r}_{i,j}}{a_{eff}r_{i,j}}\right]. \qquad (9)$$

Here, $a_{eff} = \sqrt{a_i^2 + a_j^2}$, which has been proven for $a_i = a_j$ (56) and has been proposed for $a_i \neq a_j$ (53,57).

CUDA application

To efficiently run BD simulations, we first incorporated the message-passing interface (MPI) procedure, which can compute the forces on 24 processors in parallel. However, the calculation of **H** requires significant memory as the system size increases. Thus, we calculated it on one processor, and as a result, only small systems could be simulated with proper time consumption.

With the increasing availability of GPU computing, CUDA implementation becomes feasible and preferable in our application. Because the time complexity to calculate forces and **H** is $O(n^2)$, and the Cholesky decomposition $O(n^3)$, with parallelizable algorithms, the use of GPUs has the potential to accelerate computations significantly.

Instead of nested "*for*" loops through all the interactions in the original MPI code, we assign many threads (basic computing elements) in CUDA and use "*if*" statements to accomplish the job. Namely, the n^2 calculations are performed in parallel instead of one by one. Thus, we assign an individual thread with index i,j to calculate the interaction between beads i and j. This means that, for example, for a 100-nucleosome system, 519 million threads are assigned to calculate the interactions between all pairs of beads. Compared with the 24 processors used in computer processing unit (CPU) parallel computation, this is a huge speed up.

In the GPU architecture, the threads are grouped into blocks, which are grouped onto a 3D grid. Each block might contain up to 1,024 threads, and the 3D grid contains up to 2,147,483,647, 65,535, and 65,535 blocks in the x, y, and z dimensions. Thus, the maximum number of threads in a single GPU is ~2 billion, which is more than enough to assign one thread for each interaction in a 100-nucleosome system. While the threads inside the same block can communicate with each other, those in different blocks can only communicate through the global memory, which increases non-calculation latencies due to the launching of extra kernel invocations (parallel function executions) and transferring of data to the global memory. Namely, in the CUDA code, each calculation is performed by a separate kernel (i.e., kernel A calculates forces, and kernel B calculates translation), and each kernel uses all needed threads; only after kernel A's calculations are done are the data saved to the global memory, and kernel B calculations begin.

Implementation

To convert the MPI to a CUDA application, we create M_{force} as an $N \times N$ matrix where each element $M_{force_{ij}}$ is a three-component row vector corresponding to the force between beads i and j as

$$M_{force} = \begin{bmatrix} M_{force_{11}} & M_{force_{12}} & \ldots M_{force_{1N}} & \\ M_{force_{21}} & M_{force_{22}} & \ldots M_{force_{2N}} & \\ \vdots & \vdots & & \vdots \\ M_{force_{N1}} & M_{force_{N2}} & \ldots M_{force_{NN}} & \end{bmatrix},$$

$$M_{force_{ij}} = \begin{bmatrix} F_{ij_1} & F_{ij_2} & F_{ij_3} \end{bmatrix}.$$

Thus, we calculate all energies and forces as described in the supporting material and store them in M_{force}. Then, we reduce the matrix by adding the collection of rows in M_{force} into one row ($F_j = \sum_{i=1}^{N} M_{force_{ij}}$) to obtain the array of the forces **F** applied to each bead and feed it into Eqs. 4a and 4b.

Similarly, we create M_D as an $N \times N$ matrix where each element $M_{D_{ij}}$ is a 3×3 matrix that stores the diffusion tensor coming from the hydrodynamic interactions between beads i and j as

$$M_D = \begin{bmatrix} M_{D_{11}} & M_{D_{12}} & \ldots M_{D_{1N}} & \\ M_{D_{21}} & M_{D_{22}} & \ldots M_{D_{2N}} & \\ \vdots & \vdots & & \vdots \\ M_{D_{N1}} & M_{D_{N2}} & \ldots M_{D_{NN}} & \end{bmatrix},$$

$$M_{D_{ij}} = \begin{bmatrix} D_{ij_{11}} & D_{ij_{12}} & D_{ij_{13}} \\ D_{ij_{21}} & D_{ij_{22}} & D_{ij_{23}} \\ D_{ij_{31}} & D_{ij_{32}} & D_{ij_{33}} \end{bmatrix}.$$

As shown in Eqs. 8 and 9, each component D_{ij} can be calculated individually. Thus, it is straightforward to assign $3N \times 3N$ threads, one for each D_{ij} calculation, to obtain the dense matrix M_D with a time complexity of only $O(1)$ and use it directly as **D** in Eqs. 4a and 4b.

For the Cholesky decomposition of the linear system, we implement the library provided by the vendor cuSolver and use the function "cusolverDnDpotrf," which computes the Cholesky factorization of a real double-precision Hermitian positive-definite matrix, suitable for the diffusion tensor matrix. In theory, positive definiteness is guaranteed by the diffusion tensor construction (56), but in practice, round-off error can make the matrix nonpositive definite. When this happens (infrequently), we use the diffusion tensor from the previous BD step.

The parallelization of the Cholesky decomposition is not straightforward, since the calculation of each element ij depends on information from the previous elements 1 to $i-1$ and 1 to $j-1$. In practice, this is calculated column by column rather than element by element. By looping through the columns, the results of each column$_i$ can be achieved by calling three parallelized CUDA kernels: square root kernel (calculates the diagonal element ij), normalization kernel (updates all the elements below the current diagonal element ij), and submatrix update kernel (updates the submatrix ranging from $i+1$, $i+1$ to N,N). In this way, the time complexity of the Cholesky decomposition is reduced from $O(n^3)$ to $O(n)$.

Since we do not need **D** and **F** directly, just their product (Eqs. 4a and 4b), besides the calculation of the large matrices M_{force} and M_D and the Cholesky decomposition, we calculate the translation and rotation inside the CUDA kernel. The efficient way to do so is to assign N threads for each bead i and let each thread i only calculate one translation and rotation for bead i with the corresponding data from **D** and **F** based on Eqs. 4a and 4b and Eqs. 5a and 5b.

Because GPUs cannot read and write data, after calculations are complete, we transfer the results of the locations and orientations of all the beads back from GPU to CPU to generate the desired files (i.e., trajectory, energy, checkpoint files).

RESULTS AND DISCUSSION

Performance

To assess performance, we run on both CPU and GPU simulations of chromatin fibers ranging from 2- to 100-nucleosome cores with nucleosome repeat length (NRL) = 191 bp, histone tails, and one LH per nucleosome. The timestep is 1 ps. The CPU used is an Intel Xeon Platinum 8268 24C 205W 2.9 GHz processor, and the GPU is an Nvidia Tesla v.100 16 GB processor. We recorded times for 1,000 simulation steps using the MPI code run on the CPU and the CUDA code run on the GPU. We display the wall time in Fig. 2, and its breakdown by task in Fig. 3.

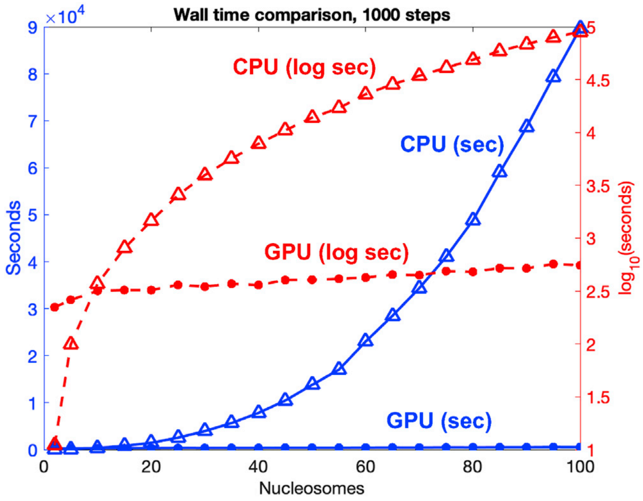

FIGURE 2 CPU versus GPU architecture for chromatin fibers of 2–100 nucleosomes. Wall time needed to complete 1,000 steps with a time step of 1 ps when running on CPU (*triangle*) versus GPU (*circle*) as a function of system size. Blue: plot in seconds. Red: plot in logarithmic scale. To see this figure in color, go online.

The wall time measures the actual time taken by the program to complete a task, including calculations, writing/reading, and communication.

The CPU wall time increases with the system size steeply (Fig. 2, *blue triangles*). On the other hand, no increase in wall time is observed on GPUs (Fig. 2, blue circles). For example, for a 100-nucleosome system, a calculation of 1,000 steps on CPUs takes 25 h, versus less than 10 min on GPUs. As observed in the plot with log scale (Fig. 2, *red curves*), the CPU calculation (*red triangles*) is advantageous compared with the GPU (red circles) for very small systems (1–9 nucleosomes) due to the “noncalculation latencies” associated with data transfer, kernel launch, and synchronization. As mentioned above, we must transfer data, such as locations and orientations of all beads, as well as random numbers between GPUs and CPUs. The kernel launch overhead reflects the time required for the GPU to execute a launched kernel. Because we cannot limit usage to a single GPU block, we must launch additional kernels, and this increases the time needed in each step of the simulation. During the simulation, each thread either belonging to the same block or different blocks needs to be synchronized to ensure that data are ready for the calculations in the next step. While synchronization of threads belonging to the same block is fast due to the use of shared memory, synchronization of threads belonging to different blocks requires the use of the global memory, which is more time consuming.

Fig. 3 shows the profiling breakdown for wall times by the MPI code for the CPU and the CUDA code for the GPU. The wall time includes the calculation time and the noncalculation latencies. When the system size increases, the time consumption of noncalculation latency increases as $O(n)$. For the MPI code, all calculations show exponential trends, and for CUDA, the calculations show linear and exponential trends (which are constant and linear trends when ignoring the noncalculation latency).

Fig. 3 *A* shows that in the MPI code, the force calculations range from 0 to ~45,000 s for systems containing 2–100 nucleosomes, which is due to the $O(n^2)$ complexity with system size. Meanwhile, the time is ~400 s for all the systems on CUDA because this calculation is performed in parallel on many threads. In addition (see implementation), each thread handles one interaction between one pair of Brownian particles, and since the nucleosome core is treated as one particle, the calculation for 300 charged beads is assigned to one thread. For calculations between two nucleosome cores, $300 \times 300 = 90{,}000$ calculations are done on one thread, explaining the force calculation time of 400 s. Although this approach might increase the force calculation time compared with treating each nucleosome charge individually, it reduces the force matrix size and thus the memory usage, allowing us to simulate larger systems.

The calculations for the diffusion tensor and Cholesky decomposition are performed every 20 steps. Fig. 3 *B* shows that the time used for the diffusion tensor calculation is smaller than the time used for the force calculation due to our treatment of the nucleosome core as one hydrodynamic particle instead of 300 charged beads as in the force calculation. The calculation on the MPI code takes as much as 1,400 s for the 100-nucleosome system, while on the CUDA code, it takes less than 1 s for all the systems.

Fig. 3 *C* shows that the most time-consuming computation is the Cholesky decomposition. Even though the Cholesky decomposition is calculated every 20 steps, it requires 36,000 s in the MPI code due to its $O(n^3)$ complexity. In comparison, in the CUDA code, the Cholesky decomposition only increases from 0 to ~50 s ($O(n)$).

Fig. 3 *D* shows the time for other calculations (such as reading and writing from and to files; translation and rotation of the beads; Euler angles; math functions such as random, exp., log, sin, cos, tan, cross product, norm, etc.) is also reduced from the CPU to the GPU due to the parallel implementations. The GPU time shows a linear trend due to remaining serial computations that increase with system size, such as random number generation at each step. Note that the “zigzag” pattern results from system sizes not being divisible by the kernel’s block size, in which case larger matrices filled with zeros need to be allocated in the GPU memory before transferring the data and launching the kernel, increasing the allocation time. Still, the time consumed for these calculations remains very small.

Validation

For our chromatin BD simulations, we set the temperature to 293.15 K and the time step to $\Delta t = 1$ ps. We saved coordinates every 100 steps and simulated at least five independent replicas started from different random seeds for each

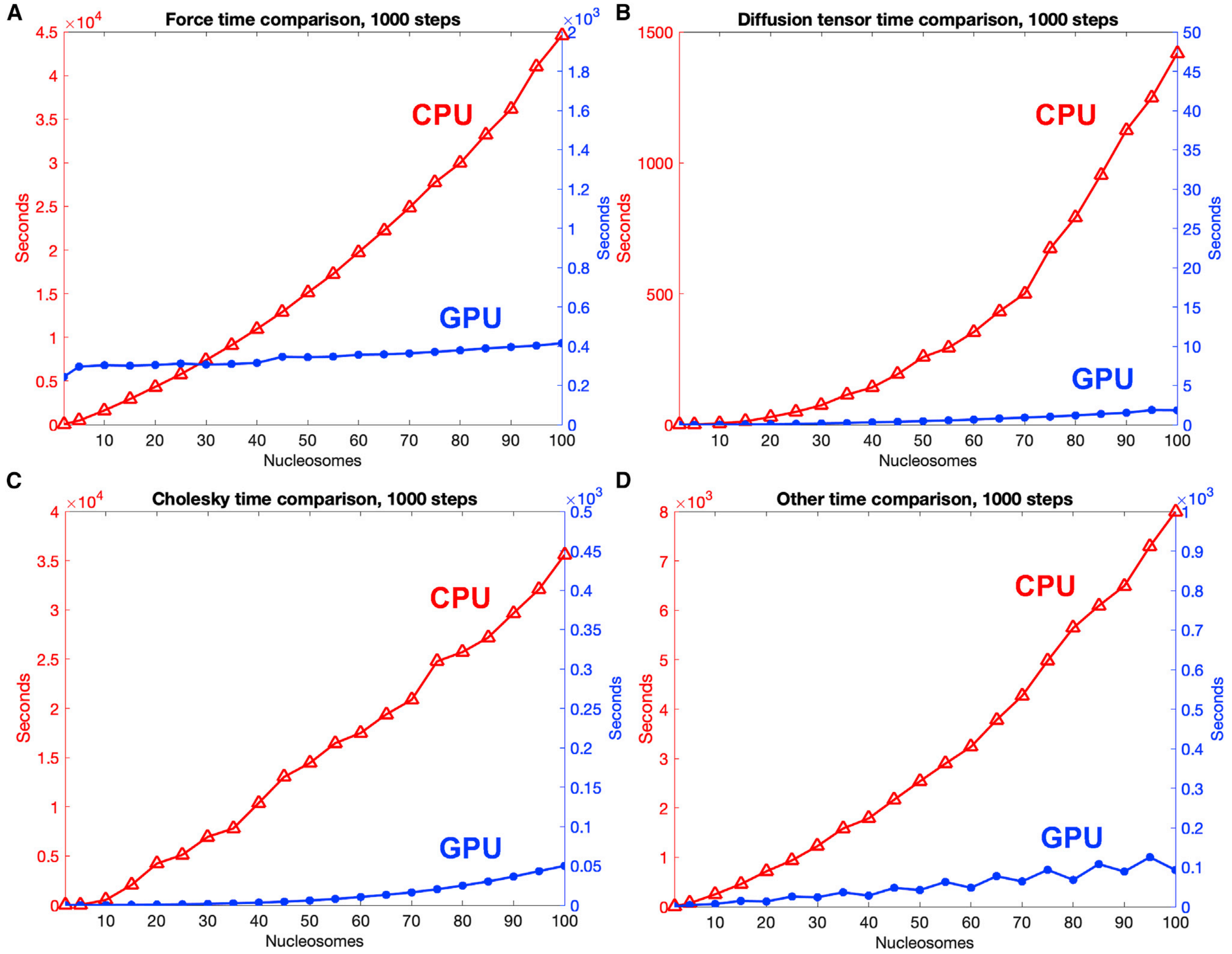

FIGURE 3 Computation time breakdown on CPU and GPU for chromatin systems of 2–100 nucleosomes. (*A–D*) Force (*A*), diffusion tensor (*B*), Cholesky decomposition (*C*), and other calculations (*D*). Note different timescales for each plot. The force includes all forces and energy calculations mentioned in the supporting material, such as stretching, bending, twisting, nonbonded electrostatics, and excluded volume. Other calculations include reading and writing from/to files; translation and rotation of the beads; Euler angle calculations; and math functions such as random, exp., log, sin, cos, tan, cross product, norm, etc. To see this figure in color, go online.

trajectory. Full details of parameters are provided in the supporting material.

Translational diffusion coefficient of nucleosomes

The velocity of the Brownian motion is defined by the translational diffusion coefficient (D), which depends on the size of the particle, the surface structure, and the ionic concentration and viscosity of the medium.

Here, we calculate D from the Einstein relation for 3D space as

$$D = \lim_{t\to\infty}\frac{\left\langle[\mathbf{r}(t_0+t)-\mathbf{r}(t_0)]^2\right\rangle}{6\cdot t}, \qquad (10)$$

where t is the simulation time, t_0 is the initial time, and $\langle\cdot\rangle$ is an average over the simulation time.

We calculate D for nucleosome arrays containing 1 to 4 nucleosomes with NRL = 209 bp at salt of 150 mM NaCl. We perform 10 independent trajectories started from different random seeds for 700 ns and compare the average (D) values with experimental data (Fig. 4) by (8,58–60) of 1- to 4-nucleosome systems obtained by fluorescence correlation spectroscopy experiments that measure fluorescence intensity fluctuations due to particle diffusion. These in vitro experiments are comparable with our BD simulations because major relevant parameters can be matched. Nonetheless, in vivo nucleosome diffusion experiments in the context of interphase chromatin report values on the order of 10^{-9} to 10^{-10} cm^2/s (61,62), close to predicted values here.

As shown in Fig. 4, the trend for diffusivity as a function of system size is reproduced by our BD simulations, and the values are close to experimental data.

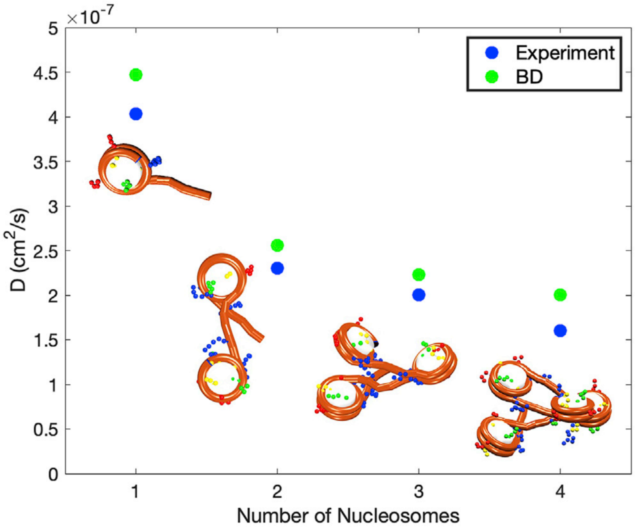

FIGURE 4 Diffusion coefficients for 1- to 4-nucleosome systems obtained from BD simulations (*green*) compared with experimental results (*blue*) from (8,58–60). To see this figure in color, go online.

TABLE 1 Chromatin compaction assessment by sedimentation coefficients of 12-nucleosome fibers with NRLs = 173, 191, and 209 bp and with 20, 50, 100, 150, and 200 mM NaCl

NRL (bp)	Monovalent salt (mM)	Computed $S_{20,w}$ (S)
173	20	35.4 ± 0.4
	50	41.5 ± 0.8
	100	47.5 ± 0.9
	150	49.8 ± 1.7
	200	51.9 ± 2.0
191	20	32.7 ± 0.5
	50	37.0 ± 1.1
	100	40.0 ± 1.4
	150	41.0 ± 1.1
	200	42.0 ± 1.3
209	20	32.1 ± 0.4
	50	34.9 ± 0.4
	100	36.0 ± 0.5
	150	36.5 ± 0.7
	200	36.5 ± 0.6

See Fig. 5 for plots with NRL = 209 bp.

Salt-dependent chromatin folding

The effect of salt concentration on chromatin folding is well known. Here, we study the folding of 12-nucleosome chromatin arrays at different monovalent salt concentrations and compare our results with two sets of experimental values obtained by Hansen et al. (63) and Howe et al. (64). In particular, we study 1) arrays with NRLs = 173, 191, and 209 bp, without LHs, and with 20, 50, 100, 150, and 200 mM NaCl, and 2) arrays with an NRL of 209 bp, with and without LH, and with 25, 50, 75, and 100 mM NaCl.

Fibers are simulated for 700 ns with a time step of $\Delta t = 1$ ps. For the systems without LHs, we start the simulations from an ideal zigzag extended geometry, whereas for the systems with LHs, which have 336 more particles, we start them from the equilibrated structures obtained with MC simulations to reduce the simulation time needed

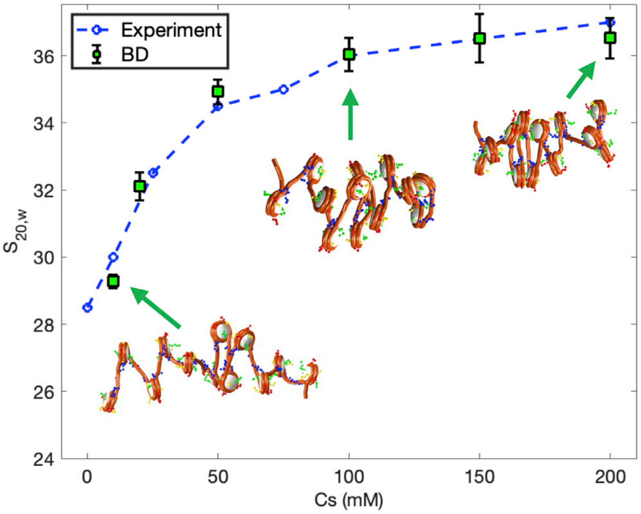

FIGURE 5 Salt-dependent folding of 12-nucleosome chromatin fibers. Sedimentation coefficients for 12-nucleosome fibers with NRL = 209 bp, without LHs, and with 10, 20, 50, 100, 150, and 200 mM NaCl. Blue dashed line represents experimental data taken from (63). Green squares with error bars show the average and standard deviation obtained from our BD simulations. Chromatin fiber configurations at increasing salt concentrations, from 10 to 200 mM. Linker DNA is shown in orange, and histone tails beads are shown in green (H3), yellow (H2A), red (H2B), and blue (H4). To see this figure in color, go online.

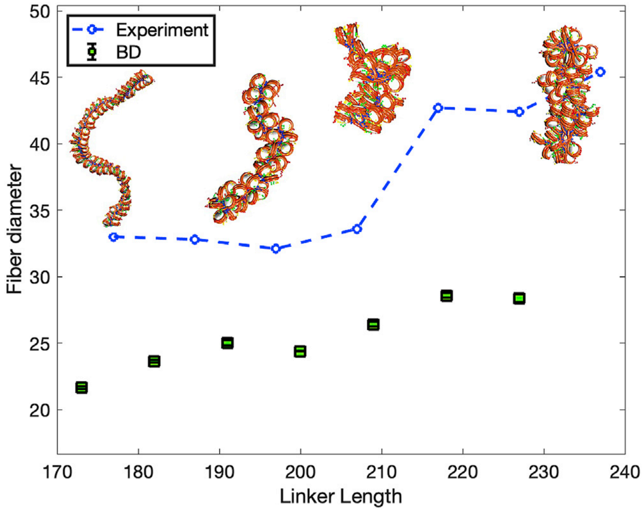

FIGURE 6 Fiber diameters for NRL-dependent folding of chromatin fibers. Fiber systems of 72, 52, 52, 61, 47, 55, 66, and 56 nucleosomes, with NRLs = 173, 182, 191, 200, 209, 218, 227, and 237 bp, respectively, without LHs, and with 120 mM NaCl. Blue dashed line represents experimental data taken from (67). Green squares with error bars show the average and standard deviation obtained from our BD simulations. Representative chromatin fiber configurations at NRLs = 173, 191, 209, and 237 bp are shown with linker DNA in orange and histone tails beads in green (H3), yellow (H2A), red (H2B), and blue (H4). To see this figure in color, go online.

to reach equilibrium conformations. We run 10 independent trajectories starting with different random seeds. Results are obtained from the last 10 ns of each simulation and averaged over the 10 trajectories.

We calculate the sedimentation coefficient at each salt concentration as follows:

$$S_{20,w} = ((S_1 - S_0) * \rho + S_0) * \left(1 + \frac{R_1}{N_C} \sum_i \sum_j \frac{1}{R_{ij}}\right), \tag{11}$$

where S_0 and S_1 are the sedimentation coefficients of a mononucleosome without LHs ($S_0 = 11.1\ S$) (65) and with LHs ($S_1 = 12\ S$) (66), respectively, ρ is the LH density on the fiber, R_1 is the radius of a nucleosome ($R_1 = 5.5$ nm), N_C is the number of nucleosomes in the chromatin fiber, and R_{ij} is the distance between the nucleosomes i and j.

BD simulations accurately reproduce the fiber compaction as a function of salt concentration observed experimentally. The sedimentation coefficient increases with the salt concentration, with a plateau reached at around 100 mM NaCl (Fig. 5). This difference in compaction is also clearly seen in the fiber configurations. Fibers at low salt appear unfolded, whereas fibers in physiological conditions are compact. Similar results are seen for the other two fibers with NRLs = 173 and 191 bp (see Table 1). Moreover, in agreement with the observations of Hansen et al. in (63), the increase of fiber compaction with salt concentration does not depend on the NRL.

At a single salt concentration, an increase in NRL produces an increase in fiber diameter, and two distinct chromatin structural classes emerge, as previously shown by Rhodes and coworkers by cryo-electron microscopy (67). As we show in Fig. 6, although absolute values are smaller, BD simulations reproduce the experimental trend for fiber diameters, showing a first structural class, composed of fibers with NRLs = 177 to 207 bp, and a second class, composed of fibers with NRLs = 217 to 237 bp. Additionally, the fiber structures resemble those seen by Rhodes et al.

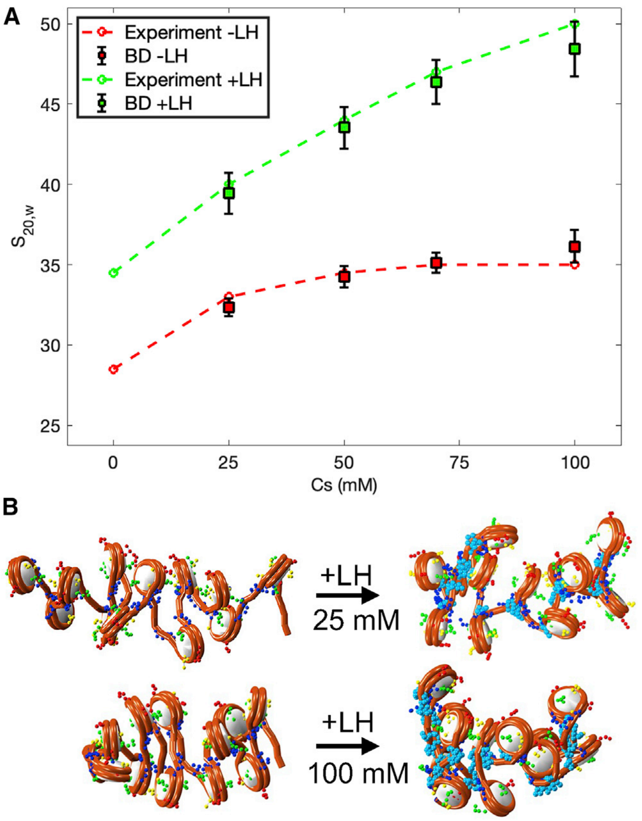

FIGURE 7 Salt- and LH-dependent folding of 12-nucleosome chromatin fibers. (*A*) Sedimentation coefficients of chromatin fibers with NRL = 209 bp, with and without LHs, and with 25, 50, 75, and 100 mM NaCl. Green (with LHs) and red (without LHs). Dashed lines represent experimental data taken from (64), and squares with error bars show average and standard deviation obtained from BD. (*B*) Fiber configurations with and without LHs at 25 and 100 mM NaCl. To see this figure in color, go online.

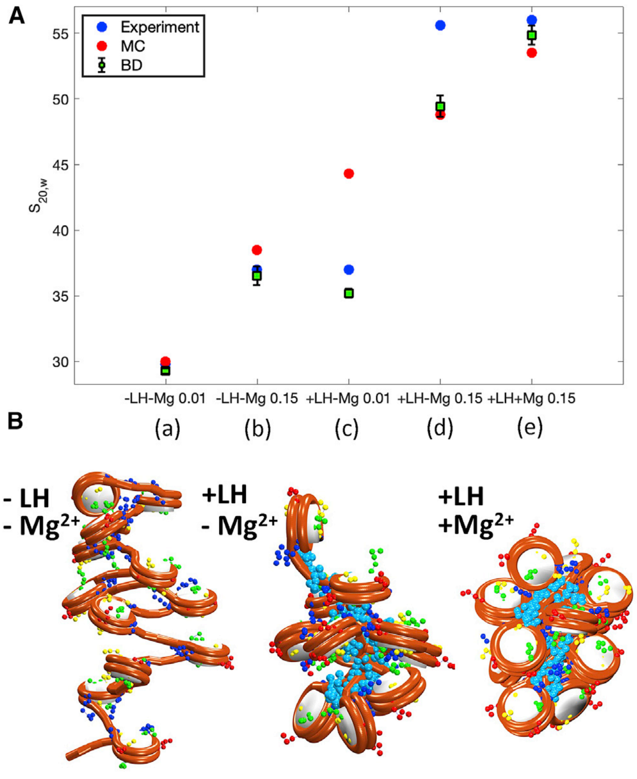

FIGURE 8 Effect of salt, Mg^{2+}, and LHs on chromatin compaction of 12-nucleosome chromatin fibers. (*A*) Sedimentation coefficient as a function of simulation condition for five fiber conditions (*left* to *right*): (*a*) 0.01 mM NaCl, no LH; (*b*) 0.15 mM NaCl, no LH; (*c*) 0.01 mM NaCl, with LH, no Mg^{2+}; (*d*) 0.15 mM NaCl, with LH, no Mg^{2+}; and (*e*) 0.15 mM NaCl, with LH, with Mg^{2+}. Blue circles represent experimental data and red circles MC results, both taken from (68). Green squares with error bars show averages and standard deviations obtained from our BD simulations. (*B*) Fiber configurations from (*b*), (*d*), and (*e*), respectively. To see this figure in color, go online.

(67). These simulations of 20 ns were started from equilibrated structures obtained from MC simulations. Five independent trajectories were run for each system starting from different random seeds, and the last 10 ns of each replica's trajectory were used for analysis.

As shown in Fig. 7 *A*, BD simulations can also accurately reproduce the experimental fiber compaction for arrays with NRL = 209 bp with and without LHs. As before, the sedimentation coefficient increases with salt concentration, and LH further increases compaction compared with fibers without LHs. Fiber configurations (Fig. 7 *B*) clearly show the impact of salt and LHs on compaction.

Salt, Mg^{2+}, and LH roles in chromatin compaction

In a previous study, we combined our MC simulations of chromatin fibers with electron microscopy nucleosome interaction capture experiments to study the internal organization of chromatin fibers (68).

There, we assessed chromatin compaction at different monovalent salt concentrations and with/without Mg^{2+} and LHs. We showed that physiological salt concentrations and LHs produce a zigzag topology with dominant *k* ± 2 internucleosome contacts. Addition of Mg^{2+} further compacts the fibers and increases the bending of the linker DNA, promoting *k* ± 1 contacts.

Here, we reproduce these results by simulating 12-nucleosome chromatin arrays with NRL = 209 bp in the following conditions: 1) 10 and 150 mM NaCl, 2) 10 and 150 mM NaCl plus 1 LH per nucleosome, and 3) 150 mM NaCl plus 1 LH per nucleosome and 1 mM Mg^{2+}. BD simulations are run for 100 ns starting from MC-equilibrated configurations in the case of fibers with LHs and from ideal zigzag structures in the case of fibers without LHs. Note that although similar results are expected from MC and BD, MC samples the equilibrium conformational ensemble, while BD samples thermal fluctuations around representative conformations.

The Mg^{2+} effect is modeled using our phenomenological approach developed in (68). That is, we reproduce the effect of 1 mM Mg^{2+} on fiber compaction by reducing the DNA persistence length from 50 to 30 nm, according to published data (69), and by increasing the inverse Debye length in the electrostatic term from 1.48 to 2.52 nm^{-1} to reflect altered electrostatic repulsion between linker DNAs.

BD simulations reproduce the experimental trends, as well as the MC results (Fig. 8 *A*). Fibers without LHs show the same compaction as the experimental and MC

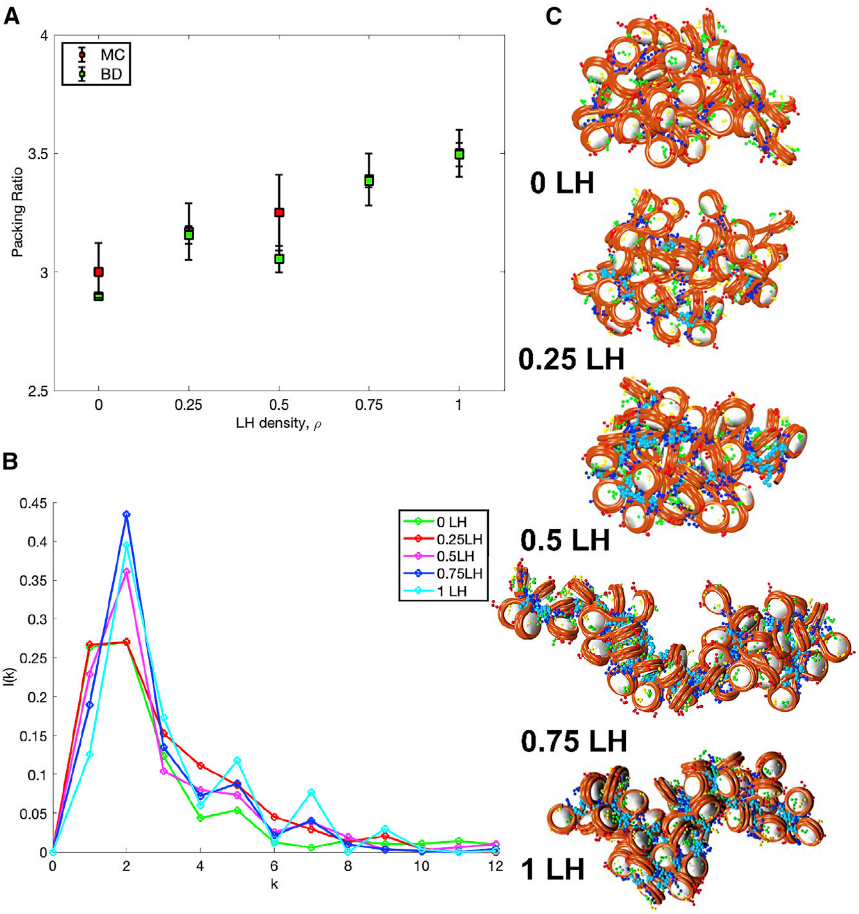

FIGURE 9 The effect of ρ on compaction of a 50-nucleosome fiber. (*A*) Packing ratio as a function of ρ (LH density) as obtained from BD simulations (*green*) and MC simulations (*red*). (*B*) Internucleosome interactions for each ρ from BD. (*C*) Representative fiber configurations from BD at each ρ. To see this figure in color, go online.

values. Such compaction increases with the addition of LHs and further with the addition of Mg^{2+} (Fig. 8 *B*). For low salt with LHs, BD conformations are slightly more open than experimental measurements suggest, while MC configurations are much more compact. For physiological salt with LHs, both BD and MC fibers are less compact than the experiment suggests. This is likely due to our fixed (rather than dynamic) LH treatment.

Role of LH in chromatin transitions

The role of LH in compacting chromatin is well known (70). Recently, we further studied how LH density, ρ, regulates chromatin architecture in relationship with the development of lymphoma (43). There, we showed that ρ triggers a transition from a flexible and globular to a rigid and straight configuration that could be related to the overexpression of genes that drive lymphoma when LHs are mutated or deficient.

Here, we reproduce this transition by studying the same chromatin fibers for different ρ values. In particular, we study 50-nucleosome arrays with an irregular DNA linker length distribution found in mammalian cells (71), and $\rho = 0$, 0.25, 0.5, and 0.75 and 1 LH per nucleosome, with LHs randomly distributed along with the fiber. Simulations of 100 ns are started from equilibrated structures obtained from MC simulations. Five independent trajectories are run for each system starting from different random seeds, and the last 10 ns of each replica's trajectory are used for analysis.

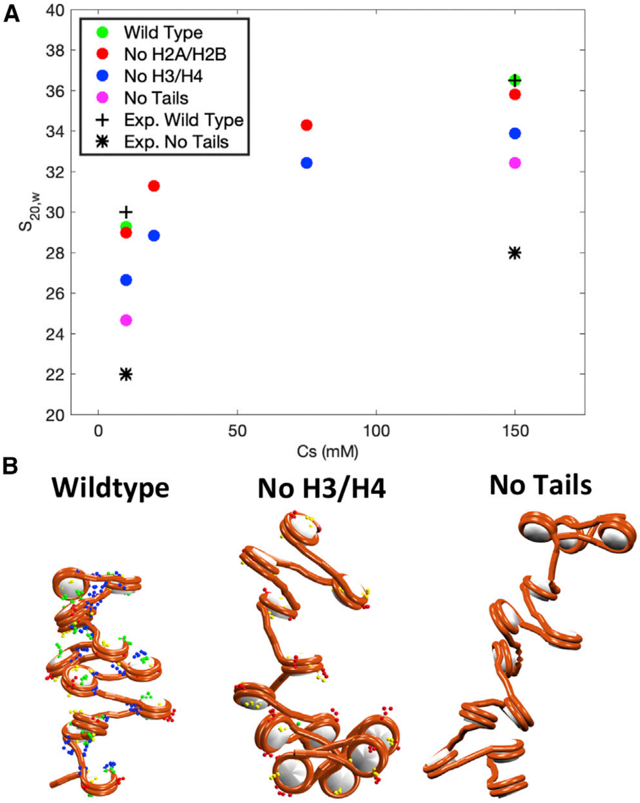

FIGURE 10 Effect of histone tails on chromatin folding at different monovalent salt concentrations for 12-nucleosome fibers. (*A*) Sedimentation coefficients as a function of salt concentration and wild-type/truncated tails. Green: all tails wild-type at 10 and 150 mM NaCl. Red: truncated H2A/H2B tails at 10, 20, 75, and 150 mM NaCl. Blue: truncated H3/H4 tails at 10, 20, 75, and 150 mM NaCl. Magenta: all tails truncated at 10 and 150 mM NaCl. (*B*) Fiber configurations obtained at 150 mM NaCl with all tails wild-type, H3/H4 tails truncated, and all tails truncated. To see this figure in color, go online.

To characterize the transition, we measure the fiber packing ratio as the number of nucleosomes contained in 11 nm of fiber length as

$$P_R = \frac{11 * N_C}{Fiber - length}, \tag{12}$$

where N_C is the total number of nucleosomes, and the fiber length is calculated by defining the fiber axis with a cubic smoothing spline interpolation to the nucleosomes *x*, *y*, and *z* coordinates (see (72)).

We also measure the internucleosome interaction patterns. That is, we determine the tail-mediated interactions between each pair i,j of cores in the chromatin fiber, constructing an interaction intensity matrix $I'(i,j)$ that describes the fraction of configurations that nucleosome pairs interact with one another. An interaction is defined when any nucleosome tail *i* is less than 2 nm apart from any charged bead or tail of nucleosome *j*. The internucleosome matrix I' is further decomposed into a normalized 1D plot that represents the relative intensity of interactions between nucleosomes separated by *k* neighbors as

$$I(k) = \frac{\sum_{i=1}^{N_C} I'(i, i \pm k)}{\sum_{j=1}^{N_C} I(j)}. \tag{13}$$

As we see from Fig. 9 *A*, BD simulations accurately reproduce the compaction trends obtained by MC simulations. As expected, fiber compaction increases with ρ.

TABLE 2 Chromatin compaction assessment by sedimentation coefficients of 12-nucleosome fibers with different tails

Array type	NaCl (mM)	Computed $S_{20,w}$ (S)	Experimental $S_{20,w}$ (S)
All tails wild-type	10	29.3 ± 0.2	30
	150	36.5 ± 0.7	36.5
Truncated H2A/H2B	10	29.0 ± 0.1	29.5
	20	31.3 ± 0.6	31
	75	34.3 ± 0.8	32
	150	35.8 ± 1.7	34
Truncated H3/H4	10	26.6 ± 0.6	26
	20	28.8 ± 0.6	27.5
	75	32.4 ± 1.0	28.5
	150	33.9 ± 1.0	31
All tails truncated	10	24.6 ± 0.1	22
	150	32.4 ± 0.4	28

See Fig. 10.

The internucleosome interaction plots (Fig. 9 *B*) show that lower ρ produces high k ± 1 contacts due to higher linker DNA bending and a more globular and disordered structure, whereas higher ρ increases the $k \pm 2$ interactions, which indicates a strong zigzag topology. Fiber configurations (Fig. 9 *C*) reveal this structural transition from a globular and loose fiber to a straight and compact fiber as ρ increases.

Role of histone tails in chromatin folding

Many experimental studies have shown how truncation of all tails or specific tails, such as H3 and H4, affect chromatin compaction (73–75). As we have shown (45), histone tails mediate chromatin compaction and higher-order folding through internucleosome interactions, which are regulated by salt conditions and the presence of LHs.

To further validate our BD approach, we reproduce sedimentation coefficient values obtained in (73) on the effect of histone tails in chromatin folding. In particular, we study 12-nucleosome fibers with NRL = 207 bp and without LHs in the following conditions: 1) all tails wild-type at 10 and 150 mM NaCl; 2) H2A and H2B tails truncated at 10, 20, 75, and 150 mM NaCl; 3) H3 and H4 tails truncated at 10, 20, 75, and 150 mM NaCl; and 4) all tails truncated at 10 and 150 mM NaCl. To truncate any tails, we set selected tail beads charges to zero.

Each system is run for 700 ns with $\Delta t = 1$ ps starting from ideal zigzag structures. Data are collected over the last 10 ns of five independent replicas started from different random seeds.

As is evident by the sedimentation coefficient values (Fig. 10 *A*) and chromatin fiber configurations (Fig. 10*B*), truncation of all tails produces a dramatic opening of the chromatin fibers. Similar results are obtained when H3/H4 tails are truncated, but the extent of the effect is smaller. In the case of H2A/H2B truncation, fibers look similar to the wild-type fibers, with almost the same sedimentation coefficients. These results are in agreement with our previous findings demonstrating the importance of histone tails in chromatin folding and, particularly, that H3 and H4 tails have a major role by mediating internucleosome interactions through cross fiber contacts that enhance long-range interactions (45).

Moreover, as shown in Table 2, our trends on sedimentation coefficients reproduce the experimental trends of (73) reasonably. Our BD computed values generally reflect slightly more compact fibers.

SUMMARY AND OUTLOOK

We have developed an efficient BD simulation protocol for GPU computing architectures to simulate chromatin fibers using our mesoscale model at nucleosome resolution. With the CUDA implementation, our BD is faster compared with a CPU implementation by two orders of magnitude, which allows us to study kb-range chromatin fibers with hundreds of nucleosomes in a few days.

Our mesoscale model for chromatin fibers has proven valuable for many applications including metaphase chromatin folding (38), compartmentalization (76), mouse cell differentiation (40), lymphoma-associated genome folding (43), yeast cell genome silencing mechanism (77), and gene folding (39,41). Although we originally started our chromatin modeling using BD (32), we moved to MC as our model increased in complexity with the introduction of flexible histone tails and LH variants with different orientations, as MC was more efficient for large systems. Here, we demonstrate that, with the advance of GPU computing, BD simulations of large nucleosome fibers are feasible with our nucleosome-resolution chromatin mesoscale model. We show the agreement of our BD results with experimental measurements such as diffusion coefficients and chromatin sedimentation on different systems, including nucleosomes at different salt concentrations and with/without LHs, ions, and tails, as well as the effect of different LH densities. With BD, dynamic chromatin problems at the nucleosome level can be studied, including chromatin flexibility, effect of binding proteins on chromatin mobility, and dynamics of gene loops.

Although BD is not likely efficient for folding fibers from ideal zigzag structures into equilibrium structures, it is suitable to study the dynamics of systems around equilibrium. Moreover, BD can be used to generate and simulate efficiently fiber topologies based on Hi-C maps. Our CUDA code ideas could also be applied to MC sampling to quickly and efficiently fold fibers into equilibrated structures in parallel. Together, MC and BD implementations for GPU architectures will allow the study of Mb-range chromatin fibers, a size not reached until now for nucleosome resolution chromatin models with explicit histone tails and LHs.

DATA AVAILABILITY

The working BD code has been deposited in GitHub under the Schlick lab group: https://github.com/Schlicklab/Brownian_Dynamics_CUDA.

SUPPORTING MATERIAL

Supporting material can be found online at https://doi.org/10.1016/j.bpj.2022.09.013.

AUTHOR CONTRIBUTIONS

Z.L. designed the research, developed the BD simulation C++ code with MPI and CUDA implementation, performed data analysis, prepared figures, and wrote the manuscript. S.P.-L. performed results analysis, wrote the manuscript, prepared figures, and performed some MC simulations. T.S. designed the research, reviewed the data and results analysis, wrote the manuscript, and obtained funding.

ACKNOWLEDGMENTS

This work was supported by the National Institutes of Health, National Institutes of General Medical Sciences Award R35-GM122562, the National Science Foundation RAPID Award (2030377) from the Divisions of Mathematical Sciences and Chemistry, and National Science Foundation Award 2151777 from the Division of Mathematical Sciences, and Philip-Morris International to T.S. The NYU IT High Performance Computing resources, services, and staff expertise are gratefully acknowledged.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Szerlong, H. J., and J. C. Hansen. 2011. Nucleosome distribution and linker DNA: connecting nuclear function to dynamic chromatin structure. *Biochem. Cell. Biol.* 89:24–34. https://doi.org/10.1139/O10-139.
2. Fussner, E., R. W. Ching, and D. P. Bazett-Jones. 2011. Living without 30nm chromatin fibers. *Trends Biochem. Sci.* 36:1–6. https://doi.org/10.1016/j.tibs.2010.09.002.
3. Klemm, S. L., Z. Shipony, and W. J. Greenleaf. 2019. Chromatin accessibility and the regulatory epigenome. *Nat. Rev. Genet.* 20:207–220. https://doi.org/10.1038/s41576-018-0089-8.
4. Rowley, M. J., and V. G. Corces. 2018. Organizational principles of 3D genome architecture. *Nat. Rev. Genet.* 19:789–800. https://doi.org/10.1038/s41576-018-0060-8.
5. Bendandi, A., S. Dante, W. Rocchia…, 2020. Chromatin compaction multiscale modeling: a complex synergy between theory, simulation, and experiment. *Front. Mol. Biosci.* 7:15. https://doi.org/10.3389/fmolb.2020.00015.
6. Dans, P. D., J. Walther, M. Orozco…, 2016. Multiscale simulation of DNA. *Curr. Opin. Struct. Biol.* 37:29–45. https://doi.org/10.1016/j.sbi.2015.11.011.
7. Portillo-Ledesma, S., and T. Schlick. 2020. Bridging chromatin structure and function over a range of experimental spatial and temporal scales by molecular modeling. *WIREs Comput. Mol. Sci.* 10:1434. https://doi.org/10.1002/wcms.1434.
8. Jimenez-Useche, I., N. P. Nurse, C. Yuan…, 2014. DNA methylation effects on tetra-nucleosome compaction and aggregation. *Biophys. J.* 107:1629–1636. https://doi.org/10.1016/j.bpj.2014.05.055.
9. Yildirim, A., L. Boninsegna, F. Alber…, 2021. Uncovering the principles of genome folding by 3D chromatin modeling. *Cold Spring Harbor Perspect. Biol.* 14:a039693. https://doi.org/10.1101/cshperspect.a039693.
10. Ozer, G., A. Luque, and T. Schlick. 2015. The chromatin fiber: multiscale problems and approaches. *Curr. Opin. Struct. Biol.* 31:124–139. https://doi.org/10.1016/j.sbi.2015.04.002.
11. Bascom, G. D., and T. Schlick. 2017. Mesoscale modeling of chromatin fibers. *In* Nuclear Architecture and Dynamics, *volume 2*. C. Lavelle and J.-M. Victor, eds.. Academic Press, pp. 123–147.
12. Huertas, J., E. J. Woods, and R. Collepardo-Guevara. 2022. Multiscale modelling of chromatin organisation: resolving nucleosomes at near-atomistic resolution inside genes. *Curr. Opin. Cell Biol.* 75:102067. https://doi.org/10.1016/j.ceb.2022.02.001.
13. Brackey, C. A., D. Marenduzzo, and N. Gilbert. 2020. Mechanistic modeling of chromatin folding to understand function. *Nat. Methods.* 17:767–775. https://doi.org/10.1038/s41592-020-0852-6.
14. Moller, J., and J. J. de Pablo. 2020. Bottom-up meets top-down: the crossroads of multiscale chromatin modeling. *Biophys. J.* 118:2057–2065. https://doi.org/10.1016/j.bpj.2020.03.014.
15. Lin, X., Y. Qi, B. Zhang…, 2021. Multiscale modeling of genome organization with maximum entropy optimization. *J. Chem. Phys.* 155:010901. https://doi.org/10.1063/5.0044150.
16. Chiang, M., C. A. Brackley, N. Gilbert…, 2022. Predicting genome organisation and function with mechanistic modelling. *Trends Genet.* 38:364–378. https://doi.org/10.1016/j.tig.2021.11.001.
17. Portillo-Ledesma, S., Z. Li, and T. Schlick. 2022. Genome modeling: from chromatin fibers to genes. *Curr. Opin. Struct.*, Submitted.
18. Annunziatella, C., A. M. Chiariello, M. Nicodemi…, 2018. Molecular dynamics simulations of the strings and binders switch model of chromatin. *Methods.* 142:81–88. https://doi.org/10.1016/j.ymeth.2018.02.024.
19. Woods, D. C., F. Rodríguez-Ropero, and J. Wereszczynski. 2021. The dynamic influence of linker histone saturation within the poly-nucleosome array. *J. Mol. Biol.* 433:166902. https://doi.org/10.1016/j.jmb.2021.166902.
20. Tan, C., J. Jung, Y. Sugita…, 2022. Implementation of residue-level coarsegrained models in GENESIS for large-scale molecular dynamics simulations. *PLoS Comput. Biol.* 18:e1009578. https://doi.org/10.1371/journal.pcbi.1009578.
21. Aumann, F., F. Lankas, J. Langowski…, 2006. Monte Carlo simulation of chromatin stretching. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 73:041927. https://doi.org/10.1103/PhysRevE.73.041927.
22. Arya, G., and T. Schlick. 2007. Efficient global biopolymer sampling with end-transfer configurational bias Monte Carlo. *J. Chem. Phys.* 126:044107. https://doi.org/10.1063/1.2428305.
23. Lequieu, J., A. Córdoba, J. J. de Pablo…, 2019. 1CPN: a coarse-grained multi-scale model of chromatin. *J. Chem. Phys.* 150:215102. https://doi.org/10.1063/1.5092976.
24. Ehrlich, L., C. Münkel, J. Langowski…, 1997. A brownian dynamics model for the chromatin fiber. *Comput. Appl. Biosci.* 13:271–279. https://doi.org/10.1093/bioinformatics/13.3.271.
25. Di Pierro, M., B. Zhang, J. N. Onuchic…, 2016. Transferable model for chromosome architecture. *Proc. Natl. Acad. Sci. USA.* 113:12168–12173. https://doi.org/10.1073/pnas.1613607113.
26. Shi, G., L. Liu, D. Thirumalai…, 2018. Interphase human chromosome exhibits out of equilibrium glassy dynamics. *Nat. Commun.* 9:3161. https://doi.org/10.1038/s41467-018-05606-6.
27. Liu, X., and E. Chow. 2014. Large-scale hydrodynamic brownian simulations on multicore and manycore architectures. *In* 2014 IEEE 28th Int. Parallel Distrib. Process. Symp. IEEE, pp. 563–572.
28. Lee, T.-S., D. S. Cerutti, D. M. York…, 2018. GPU-accelerated molecular dynamics and free energy methods in Amber18: performance enhancements and new features. *J. Chem. Inf. Model.* 58:2043–2050. https://doi.org/10.1021/acs.jcim.8b00462.
29. Ufimtsev, I. S., and T. J. Martínez. 2008. Quantum chemistry on graphical processing units. 1. Strategies for two-electron integral evaluation. *J. Chem. Theor. Comput.* 4:222–231. https://doi.org/10.1021/ct700268q.
30. Schlick, T., and S. Portillo-Ledesma. 2021. Biomolecular modeling thrives in the age of technology. *Nat. Comput. Sci.* 1:321–331. https://doi.org/10.1038/s43588-021-00060-9.
31. Schlick, T., S. Portillo-Ledesma, E. Xue…, 2021. Biomolecular modeling and simulation: a prospering multidisciplinary field. *Annu. Rev. Biophys.* 50:267–301. https://doi.org/10.1146/annurev-biophys-091720-102019.
32. Beard, D. A., and T. Schlick. 2001. Computational modeling predicts the structure and dynamics of chromatin fiber. *Structure.* 9:105–114. https://doi.org/10.1016/s0969-2126(01)00572-x.
33. Arya, G., and T. Schlick. 2006. Role of histone tails in chromatin folding revealed by a mesoscopic oligonucleosome model. *Proc. Natl. Acad. Sci. USA.* 103:16236–16241. https://doi.org/10.1073/pnas.0604817103.
34. Schlick, T., and O. Perisić. 2009. Mesoscale simulations of two nucleosome-repeat length oligonucleosomes. *Phys. Chem. Chem. Phys.* 11:10729–10737. https://doi.org/10.1039/b918629h.

35. Collepardo-Guevara, R., and T. Schlick. 2012. Crucial role of dynamic linker histone binding and divalent ions for DNA accessibility and gene regulation revealed by mesoscale modeling of oligonucleosomes. *Nucleic Acids Res.* 40:8803–8817. https://doi.org/10.1093/nar/gks600.

36. Collepardo-Guevara, R., and T. Schlick. 2014. Chromatin fiber polymorphism triggered by variations of DNA linker lengths. *Proc. Natl. Acad. Sci. USA.* 111:8061–8066. https://doi.org/10.1073/pnas.1315872111.

37. Bascom, G. D., T. Kim, and T. Schlick. 2017. Kilobase pair chromatin fiber contacts promoted by living-system-like DNA linker length distributions and nucleosome depletion. *J. Phys. Chem. B.* 121:3882–3894. https://doi.org/10.1021/acs.jpcb.7b00998.

38. Grigoryev, S. A., G. Bascom, T. Schlick…, 2016. Hierarchical looping of zigzag nucleosome chains in metaphase chromosomes. *Proc. Natl. Acad. Sci. USA.* 113:1238–1243. https://doi.org/10.1073/pnas.1518280113.

39. Bascom, G. D., K. Y. Sanbonmatsu, and T. Schlick. 2016. Mesoscale modeling reveals hierarchical looping of chromatin fibers near gene regulatory elements. *J. Phys. Chem. B.* 120:8642–8653. https://doi.org/10.1021/acs.jpcb.6b03197.

40. Gómez-García, P. A., S. Portillo-Ledesma, M. Lakadamyali…, 2021. Mesoscale modeling and single-nucleosome tracking reveal remodeling of clutch folding and dynamics in stem cell differentiation. *Cell Rep.* 34:108614. https://doi.org/10.1016/j.celrep.2020.108614.

41. Bascom, G. D., C. G. Myers, and T. Schlick. 2019. Mesoscale modeling reveals formation of an epigenetically driven HOXC gene hub. *Proc. Natl. Acad. Sci. USA.* 116:4955–4962. https://doi.org/10.1073/pnas.1816424116.

42. Portillo-Ledesma, S., L. H. Tsao, T. Schlick…, 2021. Nucleosome clutches are regulated by chromatin internal parameters. *J. Mol. Biol.* 433:166701. https://doi.org/10.1016/j.jmb.2020.11.001.

43. Yusufova, N., A. Kloetgen, A. M. Melnick…, 2021. Histone H1 loss drives lymphoma by disrupting 3D chromatin architecture. *Nature.* 589:299–305. https://doi.org/10.1038/s41586-020-3017-y.

44. Myers, C. G., D. E. Olins, T. Schlick…, 2020. Mesoscale modeling of nucleosome-binding antibody PL2-6: mono- versus bivalent chromatin complexes. *Biophys. J.* 118:2066–2076. https://doi.org/10.1016/j.bpj.2019.08.019.

45. Arya, G., and T. Schlick. 2009. A tale of tails: how histone tails mediate chromatin compaction in different salt and linker histone environments. *J. Phys. Chem. A.* 113:4045–4059. https://doi.org/10.1021/jp810375d.

46. Collepardo-Guevara, R., G. Portella, M. Orozco…, 2015. Chromatin unfolding by epigenetic modifications explained by dramatic impairment of internucleosome interactions: a multiscale computational study. *J. Am. Chem. Soc.* 137:10205–10215. https://doi.org/10.1021/jacs.5b04086.

47. Bascom, G. D., and T. Schlick. 2018. Chromatin fiber folding directed by cooperative histone tail acetylation and linker histone binding. *Biophys. J.* 114:2376–2385. https://doi.org/10.1016/j.bpj.2018.03.008.

48. Luque, A., R. Collepardo-Guevara, T. Schlick…, 2014. Dynamic condensation of linker histone C-terminal domain regulates chromatin structure. *Nucleic Acids Res.* 42:7553–7560. https://doi.org/10.1093/nar/gku491.

49. Perišić, O., S. Portillo-Ledesma, and T. Schlick. 2019. Sensitive effect of linker histone binding mode and subtype on chromatin condensation. *Nucleic Acids Res.* 47:4948–4957. https://doi.org/10.1093/nar/gkz234.

50. Zhang, Q., D. A. Beard, and T. Schlick. 2003. Constructing irregular surfaces to enclose macromolecular complexes for mesoscale modeling using the discrete surface charge optimization (DiSCO) algorithm. *J. Comput. Chem.* 24:2063–2074. https://doi.org/10.1002/jcc.10337.

51. Jian, H., A. V. Vologodskii, and T. Schlick. 1997. A combined wormlike-chain and bead model for dynamic simulations of long linear DNA. *J. Comput. Phys.* 136:168–179. https://doi.org/10.1006/jcph.1997.5765.

52. Stigter, D. 1977. Interactions of highly charged colloidal cylinders with applications to double-stranded DNA. *Biopolymers.* 16:1435–1448. https://doi.org/10.1002/bip.1977.360160705.

53. Arya, G., Q. Zhang, and T. Schlick. 2006. Flexible histone tails in a new mesoscopic oligonucleosome model. *Biophys. J.* 91:133–150. https://doi.org/10.1529/biophysj.106.083006.

54. Ermak, D. L., and J. A. McCammon. 1978. Brownian dynamics with hydrodynamic interactions. *J. Chem. Phys.* 69:1352–1360. https://doi.org/10.1063/1.436761.

55. Iniesta, A., and J. García de la Torre. 1990. A second-order algorithm for the simulation of the Brownian dynamics of macromolecular models. *J. Chem. Phys.* 92:2015–2018. https://doi.org/10.1063/1.458034.

56. Rotne, J., and S. Prager. 1969. Variational treatment of hydrodynamic interaction in polymers. *J. Chem. Phys.* 50:4831–4837. https://doi.org/10.1063/1.1670977.

57. Carrasco, B., J. García de la Torre, and P. Zipper. 1999. Calculation of hydrodynamic properties of macromolecular bead models with overlapping spheres. *Eur. Biophys. J.* 28:510–515. https://doi.org/10.1007/s002490050233.

58. Yao, J., P. T. Lowary, and J. Widom. 1990. Direct detection of linker DNA bending in defined-length oligomers of chromatin. *Proc. Natl. Acad. Sci. USA.* 87:7603–7607. https://doi.org/10.1073/pnas.87.19.7603.

59. Yao, J., P. T. Lowary, J. Yao…, 1991. Linker DNA bending induced by the core histones of chromatin. *Biochemistry.* 30:8408–8414. https://doi.org/10.1021/bi00098a019.

60. Bednar, J., R. A. Horowitz, C. L. Woodcock…, 1995. Chromatin conformation and salt-induced compaction: three-dimensional structural information from cryoelectron microscopy. *J. Cell Biol.* 131:1365–1376. https://doi.org/10.1083/jcb.131.6.1365.

61. Nozaki, T., K. Kaizu, …, K. Maeshima. 2013. Flexible and Dynamic Nucleosome Fiber in Living Mammalian Cells.

62. Hihara, S., C.-G. Pack, K. Maeshima…, 2012. Local nucleosome dynamics facilitate chromatin accessibility in living mammalian cells. *Cell Rep.* 2:1645–1656. https://doi.org/10.1016/j.celrep.2012.11.008.

63. Hansen, J. C., J. Ausio, K. E. Van Holde…, 1989. Homogeneous reconstituted oligonucleosomes, evidence for salt-dependent folding in the absence of histone H1. *Biochemistry.* 28:9129–9136. https://doi.org/10.1021/bi00449a026.

64. Howe, L., M. Iskandar, and J. Ausió. 1998. Folding of chromatin in the presence of heterogeneous histone H1 binding to nucleosomes. *J. Biol. Chem.* 273:11625–11629. https://doi.org/10.1074/jbc.273.19.11625.

65. Garcia-Ramirez, M., F. Dong, and J. Ausio. 1992. Role of the histone "tails" in the folding of oligonucleosomes depleted of histone H1. *J. Biol. Chem.* 267:19587–19595. https://doi.org/10.1016/S0021-9258(18)41815-7.

66. Butler, P. J., and J. O. Thomas. 1998. Dinucleosomes show compaction by ionic strength, consistent with bending of linker DNA. *J. Mol. Biol.* 281:401–407. https://doi.org/10.1006/jmbi.1998.1954.

67. Robinson, P. J. J., L. Fairall, D. Rhodes…, 2006. EM measurements define the dimensions of the "30-nm" chromatin fiber: evidence for a compact, interdigitated structure. *Proc. Natl. Acad. Sci. USA.* 103:6506–6511. https://doi.org/10.1073/pnas.0601212103.

68. Grigoryev, S. A., G. Arya, T. Schlick…, 2009. Evidence for heteromorphic chromatin fibers from analysis of nucleosome interactions. *Proc. Natl. Acad. Sci. USA.* 106:13317–13322. https://doi.org/10.1073/pnas.0903280106.

69. Baumann, C. G., S. B. Smith, C. Bustamante…, 1997. Ionic effects on the elasticity of single DNA molecules. *Proc. Natl. Acad. Sci. USA.* 94:6185–6190. https://doi.org/10.1073/pnas.94.12.6185.

70. Happel, N., and D. Doenecke. 2009. Histone H1 and its isoforms: contribution to chromatin structure and function. *Gene.* 431:1–12. https://doi.org/10.1016/j.gene.2008.11.003.

71. Voong, L. N., L. Xi, X. Wang…, 2016. Insights into nucleosome organization in mouse embryonic stem cells through chemical mapping. *Cell.* 167:1555–1570.e15. https://doi.org/10.1016/j.cell.2016.10.049.

72. Perišić, O., R. Collepardo-Guevara, and T. Schlick. 2010. Modeling studies of chromatin fiber structure as a function of DNA linker length. *J. Mol. Biol.* 403:777–802. https://doi.org/10.1016/j.jmb.2010.07.057.

73. Moore, S. C., and J. Ausió. 1997. Major role of the histones H3-H4 in the folding of the chromatin fiber. *Biochem. Biophys. Res. Commun.* 230:136–139. https://doi.org/10.1006/bbrc.1996.5903.

74. Hansen, J. C., C. Tse, and A. P. Wolffe. 1998. Structure and function of the core histone N-termini: more than meets the eye. *Biochemistry.* 37:17637–17641. https://doi.org/10.1021/bi982409v.

75. Dorigo, B., T. Schalch, T. J. Richmond…, 2003. Chromatin fiber folding: requirement for the histone H4 N-terminal tail. *J. Mol. Biol.* 327:85–96. https://doi.org/10.1016/s0022-2836(03)00025-1.

76. Rao, S. S. P., S.-C. Huang, E. L. Aiden…, 2017. Cohesin loss eliminates all loop domains. *Cell.* 171:305–320.e24. https://doi.org/10.1016/j.cell.2017.09.026.

77. Swygert, S. G., D. Lin, T. Tsukiyama…, 2021. Local chromatin fiber folding represses transcription and loop extrusion in quiescent cells. *Elife.* 10:e72062. https://doi.org/10.7554/eLife.72062.

Biophysical Journal, Volume *122*

Supplemental information

Brownian dynamics simulations of mesoscale chromatin fibers

Zilong Li, Stephanie Portillo-Ledesma, and Tamar Schlick

Brownian Dynamics Simulations of Mesoscale Chromatin Fibers
Supporting Information

Zilong Li[1], Stephanie Portillo-Ledesma[1], and Tamar Schlick[1,2,3]

[1]Department of Chemistry, 100 Washington Square East, Silver Building, New York University, New York, NY 10003 U.S.A.

[2]Courant Institute of Mathematical Sciences, New York University, 251 Mercer St., New York, NY 10012 U.S.A.

[3]New York University-East China Normal University Center for Computational Chemistry, New York University Shanghai, Shanghai 200122 China.

May 2022

1 Overview on overall strategy, difficulties, and approach

1.1 Strategy

To efficiently perform Brownian Dynamics (BD) simulations, we combine these key elements: our chromatin mesoscale model, our chromatin BD algorithm developed earlier [1], and the CUDA (Compute Unified Device Architecture) platform to run simulations on GPUs.

- Our mesoscale model for chromatin has evolved for 20 years [2, 3] to accurately reproduce experimental results on chromatin folding. It contains coarse grained units for the nucleosome cores [4], irregular linker DNA [5], histone tails [6], and linker histone (LH) [7]. Besides, it can treat histone tail acetylation [8, 9] and several LH binding modes and variants [10]. This makes it an excellent model to study chromatin at the nucleosome level.
- Our BD algorithm for chromatin was developed by Beard and Schlick [1] with Rotne-Prager hydrodynamic tensor using the second-order integration.
- We develop the code with the CUDA application for computing on GPUs, significantly speeding up calculations by massive parallelization.

1.2 Difficulties

The main difficulties encountered during the development were associated to the calculation of the hydrodynamic interactions, presence of non-calculation latencies associated to GPU computing, scientific correctness, and sampling issue.

- In the BD algorithm, we have a very large hydrodynamic interactions (HI) matrix that consumes memory with space complexity $O(n^2)$ and computational time associated with its Cholesky factorization with a time complexity of $O(n^3)$ for n particles.
- In the CUDA code, there are non-calculation latencies coming from using threads belonging to different blocks in the GPU. This increases data sharing time, consuming most of the wall time.
- The Brownian Dynamics strategy has been proposed over 40 years [11] and applied on many models but mostly coarse grained models. Our mesoscale model was used in MC simulations, but some artificial parameters cannot be used directly in BD (i.e., excluded volume).
- Similar to other simulation techniques, we faced the equilibrium sampling problem. As the system size increases, the time to fold fibers increases considerably.

1.3 Approach

- We assign many threads to calculate the large HI and force matrices and let each thread handle one calculation, converting the complexity from $O(n^2)$ to $O(1)$. For the Cholesky decomposition, we use the well-developed library "cuSolver" that calculates the matrix column by column in parallel and reduces the complexity from $O(n^3)$ to $O(n)$.
- To reduce non-calculation latencies as much as possible, we reduce the data transfer between GPU and CPU by completing most of the calculations in GPU, even those that were not computing intensive in CPU.
- To improve the scientific correctness we run at least 5 independent replicas for every simulation and compared with different experimental data with different perspectives and setups. Many sets of parameters were tested and justified according to the comparison of the experimental data when developing the BD code.
- The current solution for the sampling problem is to use Monte Carlo simulations to obtain equilibrated and folded fibers and then use BD to study their dynamics.

2 Chromatin Mesoscale Model

2.1 Components and their Connection

The coarse-grained chromatin model consists of nucleosome cores [4], treated as disks; and linker DNA [12], histone tails [13], and LHs [7, 10], treated as beads (Figure 1).

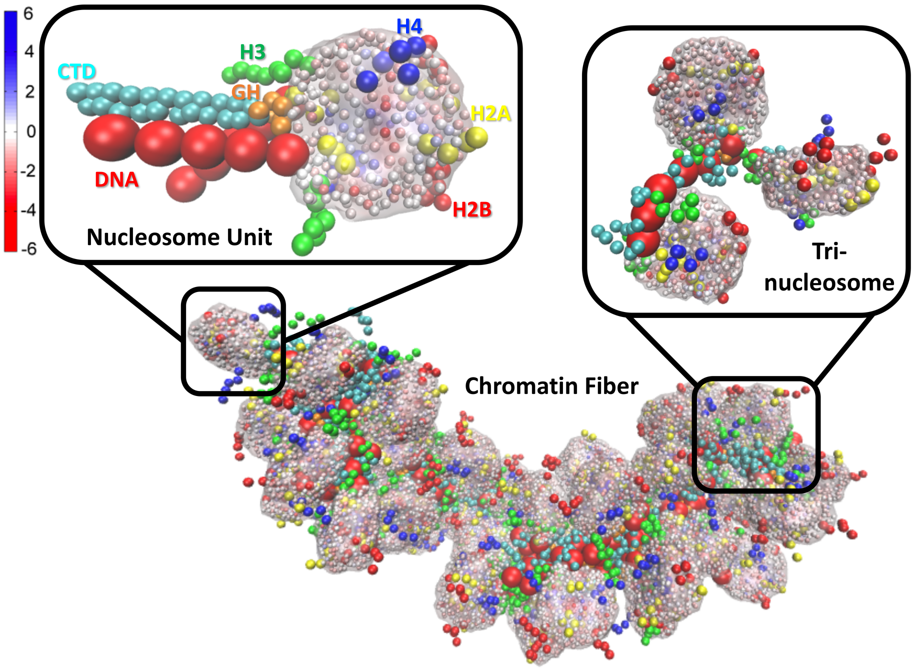

Figure 1: Chromatin mesoscale model. 50-nucleosome chromatin fiber with enlarged basic unit (chromatosome) showing each element, and enlarged tri-nucleosome showing the connection between nucleosomes by linker DNA. Linker DNA is shown as red beads, histone tails as green (H3), yellow (H2A), red (H2B), and blue (H4) beads, LHs are shown as orange (globular head) and cyan (C-terminal domain) beads, and nucleosome cores are shown with their distributed charge beads.

Our collective chromatin model can be regarded as springs connecting balls. In particular, linker DNAs connect nucleosome cores, and histone tails and LHs are attached to the nucleosome cores. Each bead can move freely during the simulation, except for the 300 charge beads in the nucleosome cores, one fixed tail bead per histone tail, and the 6 LH globular head (GH) beads, which must move together with the nucleosome core. Hence, the coarse-grained chromatin model has flexible linker DNA, histone tails and LHs, and rigid nucleosome cores.

Linker DNA – Nucleosome Core Connection The nucleosome core with 1.7 turns of DNA wrapped around is modeled as a disk [14]. Each linker DNA bead connecting nucleosomes represents roughly 9 base-pairs (detailed conversion in Equation 16). To simulate the wrapped DNA supercoil as observed experimentally, the linker DNA is not connected to the center, but to the imaginary points on the nucleosome core disk, as shown in Figure 2. Each nucleosome core disk has a radius of $r_0 = 4.8\ nm$, and parameters $w_0 = 1.8\ nm$, $\theta_0 = 108^o$ determined by the geometry of the wrapped DNA supercoil. We assign an Euler body-centered coordinate frame to each linker DNA bead and each core disk. For the Euler frame $\{\boldsymbol{a_i}, \boldsymbol{b_i}, \boldsymbol{c_i}\}$ of the core disks, the unit vectors $\boldsymbol{a_i}$ and $\boldsymbol{b_i}$ are parallel to the plane of the flat surface of the disk, while $\boldsymbol{c_i}$ is perpendicular to it. Thus, the location of the two imaginary points can be represented by:

$$\mathbf{r}_i^- = \mathbf{r}_i - r_0(-sin(\theta_0)\mathbf{a}_i + cos(\theta_0)\mathbf{b}_i) + w_0\mathbf{c}_i, \tag{1}$$

$$\mathbf{r}_i^+ = \mathbf{r}_i - (r_0\mathbf{b}_i + w_0\mathbf{c}_i). \tag{2}$$

The Euler frames on the linker DNA are updated during each step of the simulation so that the unit vector $\boldsymbol{a_i}$ is always pointing to the next bead (to the imaginary point on the nucleosome core disk if the next "bead" is a nucleosome core disk).

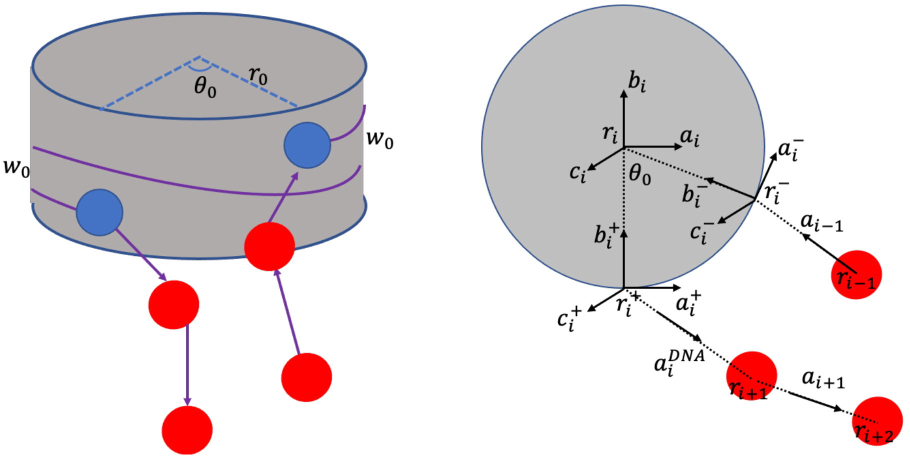

Figure 2: Relative positions of linker DNA and nucleosome core. Red beads represent linker DNA, the grey disk is the nucleosome core, and blue beads are imaginary points embedded on the nucleosome core to connect linker DNA. At left, we show the DNA supercoil and linker DNA entering and leaving the nucleosome. At right, we show the assigned Euler vectors.

Histone Tails – Nucleosome Core Connection The histone tail geometry and parameters are adopted from our previous work in [13], as shown in Figure 1. Each nucleosome core disk has 10 histone tails: two N-terminal domains of H2A (yellow beads), H2B (red beads), H3 (green beads), and H4 (blue beads), and two C-terminal domains of H2A (yellow beads). Each tail is treated as a spring with one bead fixed on the nucleosome core.

Linker Histone – Nucleosome Core Connection The LH geometry and parameters are adopted from our earlier works in [7, 10]. The coarse-grained model contains 6 beads for the GH and 22 beads for the C-terminal domain, as shown in Figure 1.

2.2 Euler Angle

Since we have defined Euler frames on the linker DNA and nucleosome cores, Euler Angles describing the relative position between beads are also calculated, and are used to calculate bending and twisting energies and force terms discussed in the following sections.

As discussed in Section 2.1, the Euler frames are updated during each step of the simulation so that the unit vector $\boldsymbol{a_i}$ always pointing to the next bead. Thus, β, shown in Figure 3, is calculated as:

$$\beta_i = \arccos(\mathbf{a}_i \cdot \mathbf{a}_{i+1}). \tag{3}$$

The other two Euler angles, α and γ, also shown in Figure 3, are defined as follows: $\boldsymbol{p}$ is the vector perpendicular to $\boldsymbol{a_i}$ and $\boldsymbol{a_{i+1}}$, α is the angle between $\boldsymbol{b_i}$ and $\boldsymbol{p}$, and γ is the angle between $\boldsymbol{p}$ and $\boldsymbol{b_{i+1}}$:

$$\alpha_i = \arccos\left(\frac{\mathbf{a}_{i+1} \cdot \mathbf{b}_i}{\sin(\beta_i)}\right), \tag{4}$$

$$\gamma_i = \arccos\left(\frac{\mathbf{b}_i \cdot \mathbf{b}_{i+1} + \mathbf{c}_i \cdot \mathbf{c}_{i+1}}{1 + \cos(\beta_i)}\right) - \alpha_i. \tag{5}$$

For the nucleosome core, the linker DNA is connected to the two imaginary points on the disk. Thus, the Euler frames assigned on these two points are used instead of these assigned to the center of the core. An additional set of Euler angles $\{\alpha_i^+, \beta_i^+, \gamma_i^+\}$ is also calculated representing the Euler transformation from $\{\boldsymbol{a}_i^+, \boldsymbol{b}_i^+, \boldsymbol{c}_i^+\}$ to $\{\boldsymbol{a}_i^{DNA}, \boldsymbol{b}_i^{DNA}, \boldsymbol{c}_i^{DNA}\}$, as shown in Figure 2.

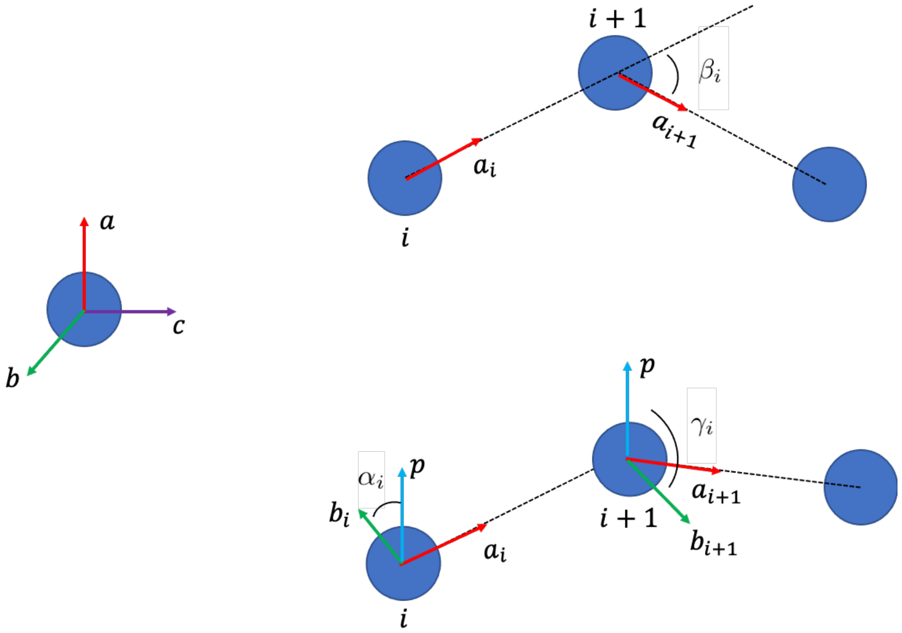

Figure 3: Schematic representation of Euler Angle related with Euler frames.

3 Energy

There are 5 types of energy terms calculated during a BD simulation that make up the effective force. These are stretching, bending, twisting, electrostatic, and excluded volume energies. The details of each energy term and the beads they applied to are given below.

3.1 Stretching

The stretching energy between two beads is based on Hooke's law, and given by:

$$E_{S_i} = \frac{h}{2}(l_i - l_0)^2, \tag{6}$$

where h is the stretching constant, l_i is the distance between two beads, and l_0 is the equilibrium length.

The stretching energy is calculated for linker DNA and nucleosome core, histone tails, LHs, and histone tails and nucleosome core.

Linker DNA – Nucleosome Core The stretching energy between linker DNA and nucleosome cores, more specifically between a DNA bead and a reference point on the nucleosome core disk (Figure 2), is the summation of stretching energy between each two adjacent beads.

$$E_S = \frac{h}{2}\sum_{i=1}^{N-1}(l_i - l_0)^2, \tag{7}$$

where $h = \frac{100 k_B T}{l_0{}^2}$, with k_B the Boltzmann's constant, T the temperature, and $l_0 = 3\ nm$.

Histone tails For the histone tails, each tail is treated as a spring, so the stretching energy is the summation of all the stretching energies calculated by each tail on all the nucleosome cores as follows:

$$E_{tS} = \sum_{i=1}^{N_c}\sum_{j=1}^{N_t}\sum_{k=1}^{N_j-1}\frac{k_{jk}}{2}(l_{ijk} - l_{jk0})^2, \tag{8}$$

where, N_c is the number of cores, N_t the number of tails per core, and N_j the number of beads per tail (each tail has different number of beads, as shown in Figure 1). The parameters k_{jk} and l_{jk0} are taken from our previous work in [13], and are shown in Table 1.

Table 1: Histone tail parameters for the stretching energy term

Tail	Bond *j*-*k*	k_{jk} $(kcal/mol/\text{Å})$	l_{jk0} [Å]
N-ter H3	1-2	0.09	14.80
	2-3	0.06	13.40
	3-4	0.07	14.50
	4-5	0.07	15.00
	5-6	0.07	14.80
	6-7	0.07	13.90
	7-8	0.11	13.70
N-ter H4	1-2	0.10	13.20
	2-3	0.10	13.90
	3-4	0.06	13.70
	4-5	0.20	14.40
N-ter H2A	1-2	0.08	13.40
	2-3	0.09	14.50
	3-4	0.03	11.00
C-ter H2A	1-2	0.07	14.10
	2-3	0.07	12.60
N-ter H2B	1-2	0.08	13.50
	2-3	0.10	12.70
	3-4	0.08	15.20
	4-5	0.08	14.20

Linker Histone For linker histones H1E, whose coarse grained model contains 6 beads for the GH and 22 beads for the C-terminal domain (CTD), as shown in Figure 1, the stretching energy is calculated as the summation of the stretching energy between all adjacent beads in the CTD and the single bead in the GH connected to the CTD as follows:

$$E_{lhS} = \sum_{i=1}^{N_{lh}}\sum_{j=1}^{N_i}\frac{k_j}{2}(l_{ij} - l_{j0})^2, \tag{9}$$

where N_{lh} is the number of nucleosome cores that have LH attached and N_i is the number of beads used to calculate the stretching energy. The constant $k_j = 0.1\ kcal/mol/\text{Å}$ and $l_{j0} = 15$ Å for the CTD beads, and $l_{j0} = 0$ for the GH beads connected to the CTD.

3.2 Bending

The bending energy is also based on the Hooke's law, which is given by:

$$E_{B_i} = \frac{g}{2}(\beta_i - \beta_0)^2, \tag{10}$$

where g is the bending constant, β_i the Euler angle described in Section 2.2, and β_0 the equilibrium angle.

The bending energy for linker DNA and nucleosome cores, histone tails, and LHs is calculated as described below.

Linker DNA – Nucleosome Core The bending energy between the linker DNA and nucleosome cores is calculated based on the Euler frames as follows:

$$E_B = \frac{g}{2}\sum_{i=1}^{N-1}\beta_i^2 + \frac{g}{2}\sum_{i\in I_C}(\beta_i^+)^2, \tag{11}$$

where g is the bending rigidity of the DNA with $g = \frac{L_P k_B T}{l_0}$, and $L_P = 50\ nm$ is the persistence length of the DNA. Again, β^+ represents the Euler transformation from $\{\boldsymbol{a}_i^+, \boldsymbol{b}_i^+, \boldsymbol{c}_i^+\}$ to $\{\boldsymbol{a}_i^{DNA}, \boldsymbol{b}_i^{DNA}, \boldsymbol{c}_i^{DNA}\}$ shown in Figure 2.

Histone Tails Similar to the stretching energy calculation, the bending energy for histone tails is the summation of all the bending energies calculated by each tail on all the nucleosome cores.

$$E_{tB} = \sum_{i=1}^{N_c}\sum_{j=1}^{N_t}\sum_{k=1}^{N_j-2}\frac{k_{\beta_{jk}}}{2}(\beta_{ijk} - \beta_{jk0})^2, \tag{12}$$

Parameters $k_{\beta_{jk}}$ and β_{jk0} are taken from our early work in [13], as in Table 2.

Table 2: Histone tails parameters for the bending energy term

	Angle	$k_{\beta_{jk}}$	β_{jk0}
Tail	*i-j-k*	$(kcal/mol/rad^2)$	$[^o]$
N-ter H3	1-2-3	1.10	115.90
	2-3-4	1.00	116.70
	3-4-5	1.70	117.30
	4-5-6	1.20	123.00
	5-6-7	1.20	111.80
	6-7-8	1.50	114.90
N-ter H4	1-2-3	1.00	112.50
	2-3-4	1.10	116.30
	3-4-5	0.50	111.60
N-ter H2A	1-2-3	1.10	121.20
	2-3-4	0.60	100.10
C-ter H2A	1-2-3	1.00	113.80
N-ter H2B	1-2-3	0.90	118.40
	2-3-4	0.60	118.90
	3-4-5	1.60	124.50

Linker Histone The bending energy is calculated as the summation of the bending energy between all adjacent beads in the CTD and the single bead of the GH connected to the nucleosome:

$$E_{lhB} = \sum_{i=1}^{N_{lh}}\sum_{j=1}^{N_i-1}\frac{k_{\beta_j}}{2}(\beta_{ij} - \beta_{j0})^2, \tag{13}$$

where $k_{\beta_j} = 1$ and $\beta_{j0} = 110^o$. For positions of the 3 adjacent beads on linker histone defined as r_{i-1}, r_i, and r_{i+1}, the angle β_i is defined as:

$$\beta_i = \arccos(\frac{(r_i - r_{i-1})\cdot(r_{i+1} - r_i)}{||(r_i - r_{i-1})||||(r_{i+1} - r_i)||}). \tag{14}$$

3.3 Twisting

The twisting energy is only applied to linker DNA and nucleosome cores and is given by:

$$E_T = \frac{s}{2l_0} \sum_{i=1}^{N-1} (\alpha_i + \gamma_i - \phi_0)^2, \tag{15}$$

where $s = 3.0 \times 10^{-12}$ *erg nm* is the torsional rigidity constant of DNA as obtained from experiments [15], α and γ are Euler angles, and ϕ_0 is a twist deviation penalty term.

To justify the twisting due to DNA, ϕ_0 is calculated as follows: considering the average rise between B-DNA base pairs 3.4 Å, and average twist 34.95^o; let the number of beads between two nucleosomes be nb, then:

$$\begin{aligned} number\_of\_base\_pair &= \frac{(nb+1) \times 3}{rise} \\ number\_of\_turns &= \frac{number\_of\_base\_pairs}{10.3} \\ DNA\_rotation &= round(number\_of\_turns) \\ rotation\_per\_base\_pair &= \frac{DNA\_rotation \times 360}{number\_of\_base\_pairs} \\ DNA\_twist &= rotation\_per\_base\_pair - twist \\ whole\_linker\_twist &= number\_of\_base\_pairs \times DNA\_twist \\ \phi_0 &= \frac{whole\_linker\_twist}{nb+1} \end{aligned} \tag{16}$$

3.4 Electrostatics

The electrostatic energy is calculated using the Debye-Hückel screened electrostatic potential, which is given by:

$$U_{DH}(q_i, q_j, r_{i,j}) = \frac{q_i q_j}{4\pi\varepsilon_0 \varepsilon r_{i,j}} exp(-\kappa r_{i,j}), \tag{17}$$

where q_i, q_j are the charges on the two beads, $r_{i,j}$ the distance between the two beads, ε the dielectric constant, and κ the inverse Debye length.

Electrostatic energies are calculated for all pairs not connected by virtual bonds. For the DNA – nucleosome core interaction, we use a cut off distance of 25 *nm*, and for all the other type of interactions, we use a cut off distance of 7 *nm*.

Linker DNA – Nucleosome Core The electrostatic energy between linker DNA and nucleosome cores is calculated using the charge on the linker DNA beads and the 300 charge beads on the nucleosome cores as follows:

$$E_C = \sum_{\substack{j>i+1 \\ i,j \in I_l}}^{N} U_{DH}(q_i, q_j, r_{i,j}) + \sum_{\substack{j>i+1 \\ i \in I_l \\ j \in I_c}}^{N} \sum_{k=1}^{N_c} U_{DH}(q_i, q_{jk}, r_{i,jk}) + \sum_{\substack{j>i \\ i,j \in I_c}}^{N} \sum_{k=1}^{N_c} \sum_{l=1}^{N_c} U_{DH}(q_{ik}, q_{jl}, r_{ik,jl}). \tag{18}$$

Here, N_c is the number of charge beads on the nucleosome cores. The first, second, and third term correspond to the interaction between linker DNA beads, interaction between linker DNA beads and nucleosome cores, and the interaction between nucleosome cores, respectively.

Histone Tails

Histone Tails – Histone Tails The electrostatic energy for tails is calculated between the beads from different tails or the non-adjacent beads from the same tail as follows:

$$E_{tC} = \sum_{\substack{i=1 \\ j>i+2 \\ i,j \in t_a}}^{N_t} U_{DH}(q_i, q_j, r_{i,j}) + \sum_{\substack{i=1 \\ j=1 \\ i,j \in t_a, t_b}}^{N_t} U_{DH}(q_i, q_j, r_{i,j}) \tag{19}$$

Histone Tails – Linker DNA The electrostatic energy is calculated between all histone tail beads and all linker DNA beads as follows:

$$E_{tlC} = \sum_{i \in I_l}^{N} \sum_{j \in I_t}^{N_t} \sum_{k=1}^{N_j} U_{DH}(q_i, q_j, r_{i,j}) \tag{20}$$

Linker Histone

Linker Histone – Linker Histone The electrostatic energy is calculated between the beads from different LHs as follows:

$$E_{lhC} = \sum_{i=1}^{N_{lh}} \sum_{j=1}^{N_i} \sum_{k \neq i}^{N_{lh}} \sum_{l=1}^{N_k} U_{DH}(q_j, q_l, r_{j,k}) \tag{21}$$

Linker Histone – Linker DNA The electrostatic energy is calculated between all LH beads and all linker DNA beads as follows:

$$E_{lhlC} = \sum_{i=1}^{N} \sum_{j=1}^{N_{lh}} \sum_{k=1}^{N_j} U_{DH}(q_i, q_k, r_{i,j}) \tag{22}$$

Linker Histone – Nucleosome Core The electrostatic energy is calculated between all LH beads and the charged beads on non-parental nucleosome cores as follows:

$$E_{lhcC} = \sum_{\substack{i=1 \\ i \in I_{c_a}}}^{N_{lh}} \sum_{j=1}^{N_i} \sum_{\substack{k=1 \\ k \notin I_{c_a}}}^{N_c} U_{DH}(q_j, q_k, r_{i,j}) \tag{23}$$

Linker Histone – Histone Tail The electrostatic energy is calculated between all LH beads and all histone-tail beads as follows:

$$E_{lhtC} = \sum_{i=1}^{N_t} \sum_{j=1}^{N_i} \sum_{k=1}^{N_{lh}} \sum_{l=1}^{N_k} U_{DH}(q_j, q_l, r_{i,j}) \tag{24}$$

3.5 Excluded Volume

The excluded volume energy is calculated using the Lennard-Jones potential given by:

$$U_{LJ}(\sigma, k_{ev}, r_{i,j}) = k_{ev}\left[\left(\frac{\sigma}{r_{i,j}}\right)^{12} - \left(\frac{\sigma}{r_{i,j}}\right)^{6}\right], \tag{25}$$

where k_{ev} is the excluded volume interaction energy parameter, and σ is the effective diameter of the two interacting beads.

We use a cutoff distance of 4 nm to calculate excluded volume energies between linker DNA - linker DNA, linker DNA - nucleosome cores, histone tails - histone tails, histone tails - linker DNA, linker histone - linker histone, linker histone - linker DNA, linker histone - nucleosome cores, and linker histone - histone tails.

Linker DNA – Nucleosome Core There are two types of excluded volume interactions, the interaction between linker DNA and nucleosome cores, and the interaction between different nucleosome cores. The excluded volume is given by:

$$E_V = \sum_{\substack{j>i+1 \\ i \in I_l \\ j \in I_c}}^{N} \sum_{k=1}^{N_c} U_{LJ}(\sigma_{lc}, k_{ev}, r_{i,jk}) + \sum_{\substack{j>i \\ i,j \in I_c}}^{N} \sum_{k=1}^{N_c} \sum_{l=1}^{N_c} U_{LJ}(\sigma_{cc}, k_{ev}, r_{ik,jl}) \tag{26}$$

Here, $k_{ev} = 0.001 k_B T$, $\sigma_{lc} = 2.4$ nm, and $\sigma_{cc} = 1.2$ nm.

Histone Tails

Histone Tails – Histone Tails The excluded volume energy for tails is calculated between the beads from different tails or the non-adjacent beads from the same tail as follows:

$$E_{tV} = \sum_{\substack{i=1\\ j>i+2\\ i,j\in t_a}}^{N_t} U_{LJ}(\sigma, k_{ev}, r_{i,j}) + \sum_{\substack{i=1\\ j=1\\ i,j\in t_a,t_b}}^{N_t} U_{LJ}(\sigma, k_{ev}, r_{i,j}) \tag{27}$$

Histone Tails – Linker DNA The excluded volume energy is calculated between all histone tails beads and all linker DNA beads as follows:

$$E_{tlV} = \sum_{i\in I_l}^{N} \sum_{j\in I_t}^{N_t} \sum_{k=1}^{N_j} U_{LJ}(\sigma, k_{ev}, r_{i,k}) \tag{28}$$

Linker Histone

Linker Histone – Linker Histone The excluded volume energy is calculated between the beads from different LHs and non-adjacent beads on the same LH as follows:

$$E_{lhV} = \sum_{i=1}^{N_{lh}} \sum_{j=1}^{N_i} \sum_{k\neq i}^{N_{lh}} \sum_{l=1}^{N_k} U_{LJ}(\sigma, k_{ev}, r_{j,l}) + \sum_{i=1}^{N_{lh}} \sum_{j=1}^{N_i-2} \sum_{k=i+2}^{N_i} U_{LJ}(\sigma, k_{ev}, r_{j,k}) \tag{29}$$

Linker Histone – Linker DNA The excluded volume energy is calculated between all LH beads and all linker DNA beads as follows:

$$E_{lhlV} = \sum_{i=1}^{N} \sum_{j=1}^{N_{lh}} \sum_{k=1}^{N_j} U_{LJ}(\sigma, k_{ev}, r_{i,k}) \tag{30}$$

Linker Histone – Nucleosome Core The excluded volume energy is calculated between the beads on the LH and the charge beads on the non-attached nucleosome cores as follows:

$$E_{lhcV} = \sum_{\substack{i=1\\ i\in I_{c_a}}}^{N_{lh}} \sum_{j=1}^{N_i} \sum_{\substack{k=1\\ k\notin I_{c_a}}}^{N_c} U_{LJ}(\sigma, k_{ev}, r_{i,k}) \tag{31}$$

Linker Histone – Histone Tail The excluded volume energy is calculated between all LH beads and all histone tails beads as follows:

$$E_{lhtV} = \sum_{i=1}^{N_t} \sum_{j=1}^{N_i} \sum_{k=1}^{N_{lh}} \sum_{l=1}^{N_k} U_{LJ}(\sigma, k_{ev}, r_{j,l}) \tag{32}$$

Table 3: Linker histone parameters for the electrostatic and excluded volume energy terms

	charge	$k_{ev_{hh}}$	$k_{ev_{hc}}$	$k_{ev_{hl}}$	$k_{ev_{ht}}$
Global Head					
1	-3.29	1.4360	1.7134	2.2180	1.6180
2	4.22	1.4720	1.7368	2.2360	1.6360
3	8.48	1.4460	1.7199	2.2230	1.6230
4	0.28	1.5380	1.7797	2.2690	1.6690
5	2.08	1.6180	1.8317	2.3090	1.7090
6	3.27	1.5280	1.7732	2.2640	1.6640
C-term					
1	3.36	1.8000	1.8000	2.7000	1.8000
2	0.00	1.8000	1.8000	2.7000	1.8000
3	5.04	1.8000	1.8000	2.7000	1.8000
4	1.68	1.8000	1.8000	2.7000	1.8000
5	3.36	1.8000	1.8000	2.7000	1.8000
6	3.36	1.8000	1.8000	2.7000	1.8000
7	3.36	1.8000	1.8000	2.7000	1.8000
8	1.68	1.8000	1.8000	2.7000	1.8000
9	5.04	1.8000	1.8000	2.7000	1.8000
10	3.36	1.8000	1.8000	2.7000	1.8000
11	3.36	1.8000	1.8000	2.7000	1.8000
12	1.68	1.8000	1.8000	2.7000	1.8000
13	3.36	1.8000	1.8000	2.7000	1.8000
14	5.04	1.8000	1.8000	2.7000	1.8000
15	5.04	1.8000	1.8000	2.7000	1.8000
16	1.68	1.8000	1.8000	2.7000	1.8000
17	3.36	1.8000	1.8000	2.7000	1.8000
18	3.36	1.8000	1.8000	2.7000	1.8000
19	3.36	1.8000	1.8000	2.7000	1.8000
20	3.36	1.8000	1.8000	2.7000	1.8000
21	5.04	1.8000	1.8000	2.7000	1.8000
22	5.04	1.8000	1.8000	2.7000	1.8000

4 Force and Torque

4.1 Force

The force **F** on the system is defined as the negative gradient with respect to the position vector collection:

$$\mathbf{F} = -\nabla_r E, \tag{33}$$

where E is the sum of all the interaction energies, and each component of the force $\mathbf{F}_i = -\nabla_{r_i} E$. Following are the details on how each force term is calculated.

Stretching For each bond i, the magnitude of stretching force is calculated by the derivative of Equation 6:

$$F_{S_i} = h(l_i - l_0). \tag{34}$$

Let B_a and B_b be the two DNA beads connected by bond i, and the direction pointing from B_a to B_b be $\vec{r}_{ab}$, then the forces applied to the DNA beads are $F_{B_a} = F_{S_i} \frac{\vec{r}_{ab}}{|\vec{r}_{ab}|}$ and $F_{B_b} = -F_{S_i} \frac{\vec{r}_{ab}}{|\vec{r}_{ab}|}$ (Figure 4).

Bending For each β, the magnitude of bending force is calculated by the derivative of Equation 10:

$$F_{B_i} = g(\beta_i - \beta_0) \tag{35}$$

Similar to the stretching force, after calculating F_{B_i}, we calculate the projection along each pair of DNA beads, and then the force applied on each bead. Instead of two beads involved in stretching, there are three beads involved in bending (Figure 4).

Twisting The twisting force, which describes the change in the torsions $(\alpha_{i-2}+\gamma_{i-2})$, $(\alpha_{i-1}+\gamma_{i-1})$, and $(\alpha_i + \gamma_i)$ created by a change in the position of the i^{th} particle, is defined similarly as we did before [1]:

$$F_T = \frac{s}{l_0}(\chi_i + \xi_i - \chi_{i-1} - \xi_{i-1}), \tag{36}$$

where the vectors χ_i and ξ_i are given by:

$$\chi_i = \frac{(\alpha_i + \gamma_i)}{l_i} \tan\frac{\beta_i}{2}(\cos\alpha_i \vec{c}_i - \sin\alpha_i \vec{b}_i) \tag{37}$$

$$\xi_i = \frac{(\alpha_{i-1} + \gamma_{i-1})}{l_i} \tan\frac{\beta_{i-1}}{2}(\cos\gamma_{i-1} \vec{c}_i + \sin\gamma_{i-1} \vec{b}_i) \tag{38}$$

Electrostatics For each pair of beads i, j, the electrostatic force is calculated by the derivative of Equation 17:

$$F_C = -\frac{q_i q_j(\kappa r_{i,j} + 1)}{4\pi\varepsilon_0 \varepsilon r_{i,j}^2} exp(-\kappa r_{i,j}) \tag{39}$$

Similar to the stretching force, we calculate the projections and apply the force to the two beads affected (Figure 4).

Exclude Volume For each pair of beads i, j, the exclude volume force is calculated by the derivative of Equation 25:

$$F_V = k_{ev}\left[\frac{6\sigma^6}{r_{i,j}^7} - \frac{12\sigma^{12}}{r_{i,j}^{13}}\right] \tag{40}$$

Similar to the stretching force, we calculate the projections and apply the force to the two beads affected (Figure 4).

4.2 Force Projection

As mentioned above, the forces are projected onto two or three beads. Below, we give the details of force projection:

Force projection onto two beads As shown in Figure 4a, the force vector is calculated as:

$$\begin{aligned} \boldsymbol{F}_i &= F\frac{\boldsymbol{r}_{ij}}{|\boldsymbol{r}_{ij}|} \\ \boldsymbol{F}_j &= F\frac{\boldsymbol{r}_{ji}}{|\boldsymbol{r}_{ji}|} \end{aligned} \tag{41}$$

Force projection onto three beads As shown in Figure 4b, the force vector is calculated as:

$$\begin{aligned} \boldsymbol{F}_i &= \frac{F}{|\boldsymbol{r}_{i,i+1}|}\frac{\boldsymbol{r}_{i+1,i} \times (\boldsymbol{r}_{i+1,i} \times \boldsymbol{r}_{i+1,i+2})}{|\boldsymbol{r}_{i+1,i} \times (\boldsymbol{r}_{i+1,i} \times \boldsymbol{r}_{i+1,i+2})|} \\ \boldsymbol{F}_{i+2} &= \frac{F}{|\boldsymbol{r}_{i+1,i+2}|}\frac{\boldsymbol{r}_{i+2,i+1} \times (\boldsymbol{r}_{i+1,i} \times \boldsymbol{r}_{i+1,i+2})}{|\boldsymbol{r}_{i+2,i+1} \times (\boldsymbol{r}_{i+1,i} \times \boldsymbol{r}_{i+1,i+2})|} \\ \boldsymbol{F}_{i+1} &= -\boldsymbol{F}_i - \boldsymbol{F}_{i+2} \end{aligned} \tag{42}$$

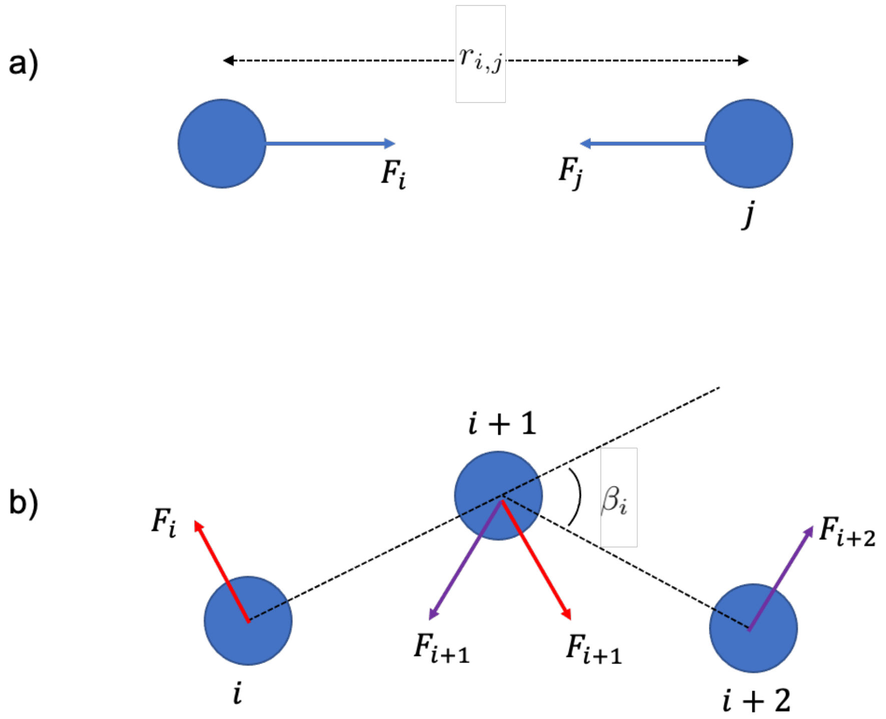

Figure 4: Force projection after calculating the magnitude of the forces. a). Force projection onto two beads, which is used in the case of stretching, electrostatics, and excluded volume forces. b) Force projection onto three beads, which is used in the case of bending force.

4.3 Torque

There are two types of torques applied during the simulation: torques due to the forces and mechanical torques due to the twisting potential in Equation 15.

Torque due to force The torques due to the forces are calculated by the equation:

$$\tau = \mathbf{r} \times \mathbf{F}, \tag{43}$$

where $\boldsymbol{r}$ is the positional vector away from the center, and $\boldsymbol{F}$ is the force vector calculated above. Note that linker DNA, histone tails, and linker histones are represented only by beads so that there is no torque due to forces applied on them. For nucleosome cores, the torques are generated by the two imaginary points when calculating stretching, bending, and twisting forces, and by the 300 charge beads when calculating electrostatics and excluded volume forces.

Mechanical torque The mechanical torques for linker DNA are due to the twisting potential and are only applied in the longitudinal direction $\boldsymbol{a}$, given by:

$$\tau_i = -\frac{s}{l_0}(\alpha_i + \gamma_i - \alpha_{i-1} - \gamma_{i-1}), \tag{44}$$

Besides the torque acting in direction $\boldsymbol{a}$, there are also torques in directions $\boldsymbol{b}$ and $\boldsymbol{c}$ acting on nucleosome cores. Thus, the total torque for nucleosome cores can be written as:

$$\tau_i = \tau_i^F + \tau_i^B + \tau_i^T, \tag{45}$$

where τ_i^F is the torque due to forces, τ_i^B the torque due to bending potential, and τ_i^T the torque due to twisting potential.

4.4 Testing

We have tested the correctness of the derivatives by checking the ratio of the Taylor expansion described below:

A Taylor series expansion for a multivariate function E at $\mathbf{x}_k + \mathbf{p}$, where $\mathbf{p}$ is the displacement vector:

$$E(\mathbf{x}_k + \mathbf{p}) \approx E(\mathbf{x}_k) + \mathbf{g}_k^T \mathbf{p} + \frac{1}{2}\mathbf{p}^T H_k \mathbf{p}. \tag{46}$$

where $\mathbf{x}_k$ is the current approximation to the solution vector $\mathbf{x}^*$, and $\mathbf{g}_k$ and H_k are the gradient and Hessian evaluated at $\mathbf{x}_k$ respectively.

For $\mathbf{p} = \varepsilon \mathbf{Y}$, with $\mathbf{Y}$ is a random perturbation vector and ε is a scalar, we have:

$$E(\mathbf{x}_k + \varepsilon \mathbf{Y}) = E(\mathbf{x}_k) + \varepsilon \mathbf{g}_k^T \mathbf{Y} + \frac{\varepsilon^2}{2}\mathbf{Y}^T H_k \mathbf{Y} + O(\varepsilon^3). \tag{47}$$

If only the gradient is tested, we have:

$$E(\mathbf{x}_k + \varepsilon \mathbf{Y}) = E(\mathbf{x}_k) + \varepsilon \mathbf{g}_k^T \mathbf{Y} + O(\varepsilon^2). \tag{48}$$

When we divide ε by 2, we have:

$$E\Big(\mathbf{x}_k + \frac{\varepsilon}{2}\mathbf{Y}\Big) = E(\mathbf{x}_k) + \frac{\varepsilon}{2}\mathbf{g}_k^T \mathbf{Y} + O\Big(\Big(\frac{\varepsilon}{2}\Big)^2\Big). \tag{49}$$

Then, the ratio tested is $O(\varepsilon^2)/O((\frac{\varepsilon}{2})^2) = 4$.

We divide ε by 2 at every step and test if indeed our truncation errors decrease with the expected rate (i.e., for a correct gradient, if the error corresponding to ε is $E1$, then the error for $\varepsilon/2$ should be $E1/4$).

The following table shows that with decreasing ε, the ratio becomes closer to 4, supporting that the analytical solutions for the forces are correct.

Table 4: Ratio with decreasing ε

Ratio	ε
21499.5	0.5
4.20064	0.25
4.10839	0.125
4.05615	0.0625
4.02855	0.03125
4.0144	0.015625
4.00723	0.0078125
4.00362	0.00390625
4.00181	0.00195312
4.00091	0.000976562
4.00045	0.000488281
4.00023	0.000244141
4.0001	0.00012207
4.00002	6.10352×10^{-5}
4.00001	3.05176×10^{-5}

5 Hydrodynamic Interactions

5.1 Translational hydrodynamics

In Brownian dynamics, the movements of the particles of the chromatin system are coupled to one another through the action of the viscous medium. The hydrodynamic friction tensor, $\mathbf{H}$, is then introduced to approximate this viscous coupling. The diffusion tensor $\mathbf{D}$ is proportional to $\mathbf{H}$ in Brownian dynamics algorithm with $\mathbf{D} = k_B T \mathbf{H}$.

For an N-bead system, $\mathbf{D}$ is a $3N \times 3N$ matrix:

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} & \dots & \mathbf{D}_{1N} \\ \mathbf{D}_{21} & \mathbf{D}_{22} & \dots & \mathbf{D}_{2N} \\ \vdots & \vdots & & \vdots \\ \mathbf{D}_{N1} & \mathbf{D}_{N2} & \dots & \mathbf{D}_{NN} \end{bmatrix}, \tag{50}$$

where each $\mathbf{D}_{ij}$ is a 3×3 matrix for the interaction between bead$_i$ and bead$_j$.

There are two types of diffusion tensors introduced in [11], the Oseen and Rotne-Prager tensors. Our approach uses the Rotne-Prager tensor because in the BD simulation the diffusion tensor needs to be a positive definite matrix to apply the Cholesky decomposition. The Oseen tensor becomes non-positive definite when the separation between the particles is small. The Rotne-Prager tensor is given by:

$$\mathbf{D}_{ij} = \left\{ \begin{array}{ll} \frac{k_B T}{6\pi\eta a_i}\mathbf{I}, & \text{for } i = j \\ \frac{k_B T}{8\pi\eta r_{i,j}} \left[\left(\mathbf{I} + \frac{\mathbf{r}_{i,j}\mathbf{r}_{i,j}}{r_{i,j}^2} \right) + \frac{(a_i^2 + a_j^2)}{r_{i,j}^2} \left(\frac{1}{3}\mathbf{I} - \frac{\mathbf{r}_{i,j}\mathbf{r}_{i,j}}{r_{i,j}^2} \right) \right], & \text{for } i \neq j \end{array} \right\}, \tag{51}$$

where η is the solvent viscosity and a is the sphere radius of the particles.

The above formula is for the two non-overlapping particles, i, j. For overlapping particles, we use:

$$\mathbf{D}_{ij} = \frac{k_B T}{6\pi\eta a_{eff}} \left[\left(1 - \frac{9}{32} \frac{r_{i,j}}{a_{eff}} \right) \mathbf{I} + \frac{3}{32} \frac{\mathbf{r}_{i,j}\mathbf{r}_{i,j}}{a_{eff} r_{i,j}} \right]. \tag{52}$$

Here $a_{eff} = \sqrt{a_i^2 + a_j^2}$, which has been proven for $a_i = a_j$ [16], and has been proposed for $a_i \neq a_j$ [13, 17].

All the particles (including nucleosome cores) are treated as spheres for hydrodynamic interactions purposes. The choice of radius for linker DNA, nucleosome cores, histone tails and LHs are 1.5 nm, 5.0 nm, 0.6 nm, and 0.5 nm, respectively. The radius of the core (a_{core}) is based on the diffusion coefficient D measured in [18], and use the relation of $D = k_B T / 6\pi\eta a_{core}$. This radius choice of the core bead is nearly equal to the volume of a disk of radius 6 nm and width 5nm. The radius for other beads (DNA, tails, LHs) are based on our touching-bead model, and the values are calculated by $a_x = \dfrac{length\ of\ string_x}{2 \times number\ of\ beads}$, ($x$ = DNA, tail, LH).

5.2 Rotational hydrodynamics

Since we only calculate torque for the linker DNA and nucleosome cores, the rotational friction is only applied to these beads. The rotational frictional coefficients adopted from [1] can be expressed as:

$$\xi_{a_i} = \xi_{b_i} = \xi_{c_i} = 8\pi\eta a_{core}^3 \tag{53}$$

for nucleosome cores, and

$$\xi_{a_i} = 8\pi\eta a_{DNA}^3 \tag{54}$$

for linker DNA, since DNA beads only rotate about the $\mathbf{a}_i$ axis.

6 Brownian Dynamics Algorithm

A commonly used Brownian Dynamics algorithm was first proposed in 1978 [11] based on the Langevin equation, then improved in 1989 [19] with second-order algorithm based on the Runge-Kutta method to overcome the issue of inefficient procedure and unstable numerical behaviour introduced by the first-order approximation when the time step is small. In 2001, Beard and Schlick [1] further modified the second-order BD algorithm and this is adopted in this paper.

6.1 First-order estimate

6.1.1 translation

The first-order translational update (at nth step) is given by:

$$\mathbf{r}^{n+1,*} = \mathbf{r}^n + \frac{\Delta t}{k_B T}\mathbf{D}(\mathbf{r}^n)\mathbf{F}^n + \mathbf{R}^n \tag{55}$$

where $\mathbf{r}$ is the position vector, Δt is the time step, k_B is the Boltzmann's constant, T is the absolute temperature, $\mathbf{D}$ is the diffusion tensor, $\mathbf{F}$ is the force, and $\mathbf{R}$ is the stochastic random force, which is a Gaussian distributed random vector with zero mean and covariance of:

$$\left\langle \mathbf{R}^n(\Delta t)(\mathbf{R}^m(\Delta t))^T \right\rangle = 2\mathbf{D}^n \Delta t \delta_{nm} \tag{56}$$

6.1.2 rotation

The first-order rotational update (at nth step) is given by:

$$\begin{aligned} \Delta\Omega_{a_i}^{n,*} &= \frac{\Delta t}{\xi_{a_i}}(\tau_{a_i}^n + \omega_{a_i}^n) \\ \Delta\Omega_{b_i}^{n,*} &= \frac{\Delta t}{\xi_{b_i}}(\tau_{b_i}^n + \omega_{b_i}^n) \\ \Delta\Omega_{c_i}^{n,*} &= \frac{\Delta t}{\xi_{c_i}}(\tau_{c_i}^n + \omega_{c_i}^n) \end{aligned} \tag{57}$$

where $\{\Delta\Omega_{a_i}^{n,*}, \Delta\Omega_{b_i}^{n,*}, \Delta\Omega_{c_i}^{n,*}\}$ is the finite rotation of the ith particle about its local coordinate system $\{\mathbf{a}_i^n, \mathbf{b}_i^n, \mathbf{c}_i^n\}$, τ is the torque, ξ_{a_i,b_i,c_i} is the rotational friction coefficient, and ω_i is the stochastic terms, with zero mean Gaussian distributions and variance of:

$$\begin{aligned} \left\langle \omega_{a_i}^n(\Delta t)\omega_{a_i}^m(\Delta t) \right\rangle &= 2k_B T \xi_{a_i} \delta_{nm} \\ \left\langle \omega_{b_i}^n(\Delta t)\omega_{b_i}^m(\Delta t) \right\rangle &= 2k_B T \xi_{b_i} \delta_{nm} \\ \left\langle \omega_{c_i}^n(\Delta t)\omega_{c_i}^m(\Delta t) \right\rangle &= 2k_B T \xi_{c_i} \delta_{nm} \end{aligned} \tag{58}$$

where δ_{nm} is the Kroneker delta.

By applying the rotation matrix $\Delta\Omega_i^{n,*}$ calculated above to the Euler frames $\{\boldsymbol{a}_i^n, \boldsymbol{b}_i^n, \boldsymbol{c}_i^n\}$ at nth step, we can obtain the first-order estimate of the Euler frames $\{\boldsymbol{a}_i^{n+1,*}, \boldsymbol{b}_i^{n+1,*}, \boldsymbol{c}_i^{n+1,*}\}$. With the rotation only applying to nucleosome cores and the $\mathbf{a}_i$ axes of the linker DNAs.

6.2 Second-order estimate

With first-order estimate of the translation and rotation at time $(n+1)\Delta t$ ($\Delta\mathbf{r}^{n,*}$ and $\Delta\Omega_i^{n,*}$), we calculate the forces $\mathbf{F}^{n+1,*}$ and torques $\tau_i^{n+1,*}$ at the end of $n+1$ time step based on $\mathbf{r}^{n+1,*}$ and $\{\boldsymbol{a}_i^{n+1,*}, \boldsymbol{b}_i^{n+1,*}, \boldsymbol{c}_i^{n+1,*}\}$, then use them to construct an explicit second-order update.

6.2.1 translation

The second-order translational update (at nth step) is given by:

$$\mathbf{r}^{n+1} = \mathbf{r}^n + \frac{\Delta t}{2k_B T}\mathbf{D}(\mathbf{r}^n)(\mathbf{F}^n + \mathbf{F}^{n+1,*}) + \mathbf{R}^n \tag{59}$$

6.2.2 rotation

The second-order rotational update (at nth step) is given by:

$$\begin{aligned} \Delta\Omega_{a_i}^{n} &= \frac{\Delta t}{\xi_{a_i}}((\tau_{a_i}^n + \tau_{a_i}^{n+1,*})/2 + \omega_{a_i}^n) \\ \Delta\Omega_{b_i}^{n} &= \frac{\Delta t}{\xi_{b_i}}((\tau_{b_i}^n + \tau_{b_i}^{n+1,*})/2 + \omega_{b_i}^n) \\ \Delta\Omega_{c_i}^{n} &= \frac{\Delta t}{\xi_{c_i}}((\tau_{c_i}^n + \tau_{c_i}^{n+1,*})/2 + \omega_{c_i}^n) \end{aligned} \tag{60}$$

6.3 Cholesky decomposition

The Cholesky approach is used to compute $\mathbf{R}$ to satisfy the proprieties of Eq. (56). The Cholesky decomposition of the diffusion tensor $\mathbf{D}$ is determined by $\mathbf{D} = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is a lower triangular matrix and each element in $\mathbf{L}$ is given by:

$$l_{ij} = \left\{ \begin{array}{ll} (D_{ii} - \sum_{k=1}^{i-1} l_{ik}^2)^{\frac{1}{2}}, & \text{if } i = j \\ (D_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk})/s_{jj}, & \text{if } i > j \\ 0, & \text{if } i < j \end{array} \right\}, \tag{61}$$

7 Summary of parameters used in the simulations

We adopt parameters mainly from previous works in Schlick's lab [1, 13] and unified the units. Below is a summary of the parameters used in our program.

Table 5: Parameters used in the BD simulation

Parameter	Description	Value
T	Temperature	$293.15\ K$
Δt	time step	$10^{-12}\ s$
k_B	Boltzmann constant	$1.380649 \times 10^{-5}\ (nm^2 \cdot kg)/(s^2 \cdot K)$
η	viscosity of the surrounding fluid	$1.137076 \times 10^{-12}\ kg/(nm \cdot s)$
a_{core}	hydrodynamic radius of the core	$5.0\ nm$
a_{DNA}	hydrodynamic radius of DNA	$1.2\ nm$
ξ_{core}	rotational frictional coefficient of the core	$8\pi\eta d^3_{core}$
ξ_{DNA}	rotational frictional coefficient of DNA	$8\pi\eta a^3_{DNA}$
l_0	Equilibrium segment length	$3.0\ nm$
r_0	Radius of wound DNA supercoil	$4.8\ nm$
$2\omega_0$	Width of wound DNA supercoil	$3.6\ nm$
h	DNA Stretching constant	$100k_BT/l_0^2$
g	DNA Bending constant	$50k_BT/l_0$ (without Mg) or $30k_BT/l_0$ (with Mg)
C	DNA Twisting constant	$72.429k_BT/l_0$
k_e	Electrostatics parameter	$0.4151k_BT$
k_{ex}	Excluded volume parameter	$0.001k_BT$
$\sigma_{DNA-DNA}$	DNA-DNA excluded volume distance	$3.6\ nm$
$\sigma_{DNA-Core}$	DNA-Core excluded volume distance	$2.4\ nm$
$\sigma_{Core-Core}$	Core-Core excluded volume distance	$1.2\ nm$
$\sigma_{Tail-Tail}$	Tail-Tail excluded volume distance	$1.8\ nm$
$\sigma_{Tail-DNA}$	Tail-DNA excluded volume distance	$2.7\ nm$
$\sigma_{Tail-Core}$	Tail-Core excluded volume distance	$1.8\ nm$

References

[1] D. A. Beard and T. Schlick, "Computational Modeling Predicts the Structure and Dynamics of Chromatin Fiber," *Structure*, vol. 9, no. 2, pp. 105–114, 2001.

[2] S. Portillo-Ledesma and T. Schlick, "Bridging chromatin structure and function over a range of experimental spatial and temporal scales by molecular modeling," *WIREs Comput. Mol. Sci.*, vol. 10, no. 2, 2020.

[3] G. D. Bascom and T. Schlick, "Chromatin Fiber Folding Directed by Cooperative Histone Tail Acetylation and Linker Histone Binding," *Biophys. J.*, vol. 114, no. 10, pp. 2376–2385, 2018.

[4] Q. Zhang, D. A. Beard, and T. Schlick, "Constructing Irregular Surfaces to Enclose Macromolecular Complexes for Mesoscale Modeling Using the Discrete Surface Charge Optimization (DiSCO) Algorithm," *J. Comput. Chem.*, vol. 24, no. 16, pp. 2063–2074, 2003.

[5] R. Collepardo-Guevara and T. Schlick, "Chromatin fiber polymorphism triggered by variations of DNA linker lengths," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 111, no. 22, pp. 8061–8066, 2014.

[6] G. Arya and T. Schlick, "A tale of tails: how histone tails mediate chromatin compaction in different salt and linker histone environments," *J. Phys. Chem. A*, vol. 113, no. 16, pp. 4045–4059, 2009.

[7] A. Luque, R. Collepardo-Guevara, S. Grigoryev, and T. Schlick, "Dynamic condensation of linker histone C-terminal domain regulates chromatin structure," *Nucleic Acids Res.*, vol. 42, no. 12, pp. 7553–7560, 2014.

[8] R. Collepardo-Guevara, G. Portella, M. Vendruscolo, D. Frenkel, T. Schlick, and M. Orozco, "Chromatin Unfolding by Epigenetic Modifications Explained by Dramatic Impairment of Internucleosome Interactions: A Multiscale Computational Study," *J. Am. Chem. Soc.*, vol. 137, no. 32, pp. 10205–10215, 2015.

[9] G. D. Bascom and T. Schlick, “Chromatin Fiber Folding Directed by Cooperative Histone Tail Acetylation and Linker Histone Binding,” *Biophys. J.*, vol. 114, no. 10, pp. 2376–2385, 2018.

[10] O. Perisic, S. Portillo-Ledesma, and T. Schlick, “Sensitive effect of linker histone binding mode and subtype on chromatin condensation,” *Nucleic Acids Res.*, vol. 47, no. 10, pp. 4948–4957, 2019.

[11] D. L. Ermak and J. A. McCammon, “Brownian dynamics with hydrodynamic interactions,” *J. Chem. Phys.*, vol. 69, no. 4, pp. 1352–1360, 1978.

[12] H. Jian, A. V. Vologodskii, and T. Schlick, “A combined wormlike-chain and bead model for dynamic simulations of long linear DNA,” *J. Comput. Phys.*, vol. 136, no. 1, pp. 168–179, 1997.

[13] G. Arya, Q. Zhang, and T. Schlick, “Flexible histone tails in a new mesoscopic oligonucleosome model,” *Biophys. J.*, vol. 91, no. 1, pp. 133–150, 2006.

[14] D. A. Beard and T. Schlick, “Modeling salt-mediated electrostatics of macromolecules: The discrete surface charge optimization algorithm and its application to the nucleosome,” *Biopolymers*, vol. 58, no. 1, pp. 106–115, 2001.

[15] D. S. Horowitz and J. C. Wang, “Torsional rigidity of DNA and length dependence of the free energy of DNA supercoiling,” *J. Mol. Biol.*, vol. 173, no. 1, pp. 75–91, 1984.

[16] J. Rotne and S. Prager, “Variational Treatment of Hydrodynamic Interaction in Polymers,” *J. Chem. Phys.*, vol. 50, no. 11, pp. 4831–4837, 1969.

[17] B. Carrasco, J. García de la Torre, and P. Zipper, “Calculation of hydrodynamic properties of macromolecular bead models with overlapping spheres,” *Eur. Biophys. J.*, vol. 28, no. 6, pp. 510–515, 1999.

[18] J. Yao, P. T. Lowary, and J. Widom, “Direct detection of linker DNA bending in defined-length oligomers of chromatin,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 87, no. 19, pp. 7603–7607, 1990.

[19] A. Iniesta and J. García De La Torre, “A second-order algorithm for the simulation of the Brownian dynamics of macromolecular models,” *J. Chem. Phys.*, vol. 92, no. 3, pp. 2015–2018, 1990.