Bridging the Coarse-grained to Microscopic information gap: A numerical optimization method

Andrew Abi Mansour and Peter J. Ortoleva*

Department of Chemistry, Indiana University, Bloomington

E-mail: ortoleva@indiana.edu

Abstract

Atom-resolved states must be constructed as part of a multiscale algorithm that coevolves the system at the atomic and coarse-grained (CG) scales. The CG description does not capture the constraints on distances and angles imposed by stiff bonded interactions. Thus, in isothermal simulations, using only CG information to construct the initial conditions yields microstates of negligible Boltzmann weight. In this paper, we present a reversible CG to all-atom mapping algorithm that overcomes this difficulty. The result is a scalable algorithm for simulating mesoscopic systems with atomic precision, over long periods of time, and with great efficiency over conventional MD. The mapping algorithm is implemented in parallel for distributed memory systems as part of the Deductive Multiscale Simulator software. It is demonstrated for Lactoferrin, an assembly of Nudaurelia Capensis Omega proteins, and Cowpea Chlorotic Mottle virus capsid.

Introduction

Mesoscopic systems such as nanocapsules, viruses, and ribosomes evolve through the coupling of processes across multiple scales in space and time. Therefore, a theory of the dynamics of

*To whom correspondence should be addressed
these systems must somehow account for the coevolution of coarse-grained (CG) and microscopic variables. Methods based on deductive multiscale analysis (DMA) have shown great promise in achieving efficiency\textsuperscript{11}–\textsuperscript{15} over conventional molecular dynamics (MD) via coevolving the micro and CG states. Some of these methods have used the space-warping approach\textsuperscript{6} to coarse grain the atomistic configuration, evolve the CG variables, and then reconstruct the microstate. The drawback with this approach is that bonds tend to be highly strained when overall deformations implied by the CG state are expressed. This leads to computational difficulties because most of these configurations are typically unlikely because of their negligible Boltzmann weight. Therefore, to construct thermal averages, these configurations must be modified using energy minimization and thermalization\textsuperscript{12} (see Figures 1 and 2), and similarly for single evolution scenarios developed using Trotter factorization\textsuperscript{13}. This creates a computational burden and thereby detracts from much of the efficiency that would have otherwise been afforded by the multiscale approach. Therefore, while there is much discussion about the microstate to CG state mapping\textsuperscript{6–11}, the information gap between these levels has been a challenge for understanding multiscale systems.

In this paper, we extend the space-warping method introduced earlier\textsuperscript{6} by accounting for CG to microstate transformation in a way that respects short scale effects such as the constraints imposed by stiff bonds. This is achieved by constraining the fine-scaling algorithm to a set of equations that enforce constants bond lengths and angles. Constraint algorithms are often applied to MD simulations inorder to increase the time step a solver can take. This is because at the atomic scale, Newton’s equations of motions are highly stiff, and MD integrators use explicit methods\textsuperscript{14} to numerically integrate these equations. Therefore, the time step is restricted to the timescale of bond fluctuations. Constrained MD attempts to resolve this problem by neglecting the bond fluctuations. This is achieved by adding constraint forces to Newton’s equations of motion such that these forces prevent any two bonded atoms from changing their bond lengths, and any three consecutively bonded atoms from changing their angle (See Figure 6). Constraint algorithms are usually based on some form of the method of undetermined Lagrange multipliers\textsuperscript{15–18}, and they perform well for specific systems and when the time step of evolution is quite small. The first algorithm to
satisfy geometrical bond constraints was SHAKE\textsuperscript{16,17}. This algorithm was limited to mechanical systems with a tree structure. A later extension of the method, QSHAKE (Quaternion SHAKE) was developed to amend this\textsuperscript{19}. It works satisfactorily for rigid loops such as aromatic ring systems but fails for flexible loops\textsuperscript{20} (e.g. proteins having a disulfide bridge). Further extensions include RATTLE\textsuperscript{21}, WIGGLE\textsuperscript{22}, and MSHAKE\textsuperscript{18}. RATTLE is similar to SHAKE in that it uses the Velocity Verlet time integration scheme, but it offers higher precision. WIGGLE extends SHAKE and RATTLE by using an initial estimate for the Lagrange multipliers based on the particle velocities. MSHAKE computes corrections on the constraint forces to achieve better convergence. Another modification is the P-SHAKE algorithm\textsuperscript{23} for rigid or semi-rigid molecules. P-SHAKE computes and updates a pre-conditioner which is applied to the constraint gradients before the SHAKE iteration, causing the Jacobian to become diagonal or strongly diagonally dominant. The de-coupled constraints converge much faster (quadratically as opposed to linearly) at a cost of $O(N^2)$ flops.

An alternative constraint method called LINCS (Linear Constraint Solver) was developed in 1997 by Hess, Bekker, Berendsen and Fraaije\textsuperscript{24}. LINCS approximates the inverse of the Jacobian of constraints with a power series using geometric progression for each Newton step. This approximation works only for molecules with low connectivity since the eigenvalues of the Jacobian have to be smaller than 1.

The objective of the present study is to develop a robust and scalable algorithm that computes the all-atom configuration from a given CG state without any restrictions on the connectivity of the macromolecule. Rather than modifying the CG dynamical equations, here a CG-to-atom resolved state map that takes into account the information needed to avoid unphysical bond strains is presented. The algorithm exploits the connectivity of atomic bonds and angles in a given macromolecule. This is based on the fact that atoms that are spatially far from each other are independent and therefore do not contribute to the same constraint equation. Therefore, all the assembled matrices are naturally sparse, which we exploit in solving the linearized systems. This is the source of the efficiency and scalability of the constrained mapping algorithm, which allows efficient parallelization of the code on distributed and shared memory systems. The study is based on im-
plementing the proposed algorithm in the framework of Deductive Multiscale Simulator (DMS) package\textsuperscript{4,25,26}. The formulation is provided, as are the specific algorithm implementation and parallelization, results and discussion are given, and conclusions are drawn.

**Formulation**

The development starts with a set of space warping CG variables (denoted $\phi$) that were used to simulate large-scale macromolecular conformational changes\textsuperscript{6}. The fine graining relationship, i.e. obtaining the atom-resolved configuration consistent with the CG variables, takes the form

$$ r_i = \sum_k U_k(r^0_i)\phi_k + \sigma_i. $$

Here $r$ is a vector of all $3N$ atomic coordinates, $\sigma$ is a vector of $3N$ residual displacements from coherent deformation generated by the first term of the RHS of Eq. (1), and $U$ is a matrix (of size $N \times N_{CG}$) of mass-weighted orthonormalized Legendre functions\textsuperscript{26}. These CG variables describe the overall features of the system whereas $r$ describes the atomic configuration directly. The basis functions depend on the reference all-atom configuration $r^0$. The reference structure introduces a configuration as determined by X-ray and Cryo-EM data. Thus, the CG variables specify how the structure is deformed from this reference configuration (i.e. due to the temperature and fluid condition of interest). This method was extended\textsuperscript{27} later on by maximizing the amount of information contained in $\phi$, and in particular by minimizing the mass-weighted square residual. The result is a reversible $r \leftrightarrow \phi$ map, which provides a unique CG state given $r$, but an ensemble of $r$ given $\phi$.

For conciseness, Eq. (1) is recast in matrix form:

$$ r = K\phi + \sigma, $$

(2)
where

$$K = \begin{pmatrix} U & 0 & 0 \\ 0 & U & 0 \\ 0 & 0 & U \end{pmatrix}_{3N \times N_{CG}}$$

The constraint mapping approach begins by defining the objective function $f$ via

$$f(r) = \frac{1}{2} (r - K\phi - \sigma)'(r - K\phi - \sigma).$$

This function is convex and therefore admits a single unique minimum at $r_u = K\phi + \sigma$, the unconstrained map defined in Eq. (3). Here we introduce two sets of constraints that enforce constant bond lengths and angles, respectively. Thus, as $\phi$ changes, the minimization of $f$ with respect to $r$ subject to these constraints yields atom-resolved configurations consistent with the CG state. The first set of constraints is written in the form:

$$|r_i - r_j| = d_{ij},$$

such that $d_{ij}$ is the equilibrium bond length between atoms $i$ and $j$. The second set of constraints ensures the three-body angle, defined between two consecutive bonds (see Figure 6), is respected

$$\theta_{ijk} = \theta_{ijk}^{eq},$$

such that the indices $i, j, k$ span every two consecutive bonds and $\theta_{ijk}^{eq}$ is the value of the angle between these two bonds at equilibrium. This constraint can be satisfied by setting $|r_i - r_k|$ equal to $d_{ik}$. For convenience, the inter-atomic distance squared is used in all constraint equations. The optimization problem can then be stated as follows:

$$\min_{r} f(r),$$

5
subject to

\[ |\mathbf{r}_i - \mathbf{r}_j|^2 = d_{ij}^2, \]  
\[ |\mathbf{r}_i - \mathbf{r}_k|^2 = d_{ik}^2. \]  

(7) \hspace{1cm} (8)

For efficient parallelization, these equations are formulated in matrix form. To achieve that, the entire macromolecule is represented as a Graph. In Graph theory\[^{28}\], a set of vertices (atoms) can link in pairs to form edges (bonds). To take the non-bonded angle-determining distance \(d_{ik}\) into account, we introduce an adjacency-like matrix \(A\) that captures the location of each atomic index in every constraint equation, i.e., for the \(l^{th}\) constraint spanning atoms \(i\) and \(j\), row \(l\) in \(A\) has +1 entry at column \(i\), and −1 at column \(j\), while all remaining columns have zero entries. The Lagrangian \(\mathcal{L}\) of the optimization problem can be written as

\[
\mathcal{L} = f(\mathbf{r}) + \lambda^t \left( \sum_{d=1}^{3} \mathbf{D}(\mathbf{A}\mathbf{r}_d)\mathbf{A}\mathbf{r}_d - \mathbf{l}_{eq} \right),
\]

(9)

for a vector of Lagrange multipliers \(\lambda\), and vector \(\mathbf{l}_{eq} = \sum_{d=1}^{3} \mathbf{D}(\mathbf{A}\mathbf{r}_d^{eq})\mathbf{A}\mathbf{r}_d^{eq}\) that represents the equilibrium inter-atomic distances squared. To simplify the notation, we defined \(\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)\). Here \(\mathbf{D}(\mathbf{v})\) represents the diagonal matrix form of vector \(\mathbf{v}\). By using the Lagrange method for equality constraints\[^{29}\], the Lagrangian with respect to \(\mathbf{r}\) and \(\lambda\) was minimized i.e. the equations \(\nabla_{\mathbf{r}} \mathcal{L} = 0\) and \(\nabla_{\lambda} \mathcal{L} = 0\) were solved such that at the optimum solution \((\mathbf{r}^*, \lambda^*)\) the Hessian of \(\mathcal{L}\) is positive definite. This leads to the following non-linear coupled equations:

\[
\mathbf{r} - \mathbf{r}_u - \mathbf{J}_r^t \lambda = \mathbf{0},
\]

(10)

\[
\sum_{d=1}^{3} \mathbf{D}(\mathbf{A}\mathbf{r}_d)\mathbf{A}\mathbf{r}_d - \mathbf{l}_{eq} = \mathbf{0}.
\]

(11)

where \(\mathbf{J}_r\) is the Jacobian of Eqs. (7, 8). When \(\lambda\) is zero, the unconstrained solution is recovered. Unlike constrained MD\[^{30}\], this procedure does not modify the ensuing Newtonian Physics, and furthermore yields the same coarse-graining map \(\mathbf{r} \Leftrightarrow \phi\) noted above, i.e., the definition of \(\phi\)
is unaltered (see Appendix).

**Methodology**

Finding an analytical solution to the optimization problem presented in the previous section is not feasible. Therefore, the following numerical approach was implemented. First the constraints equations were recast in the form

\[ c = \sum_{d=1}^{3} D(Ar_d) Ar_d - l_{eq}. \]  

(12)

The vector function \( c \) depends on \( \lambda \), in a way that can be made explicit by substituting \( r \) from Eq. (??) to give

\[
\begin{align*}
    c(\lambda) &= \sum_{d=1}^{3} D(Ar_d) Ar_d + D(Ar_d) AJ_t r_d \lambda \\
    &+ D(AJ_t r_d \lambda) Ar_d + D(AJ_t r \lambda) AJ_t r_d \lambda - l_{eq}
\end{align*}
\]

(13)

Here \( J_{r,0}^{t}, J_{r,1}^{t}, J_{r,2}^{t} \) are the \( x, y, \) and \( z \) component block matrices of \( J_{r}^{t} \), i.e.

\[
J_r^{t} = \begin{pmatrix}
J_{r,0}^{t} \\
J_{r,1}^{t} \\
J_{r,2}^{t}
\end{pmatrix}_{3N \times N_c}
\]

(14)

Thus, the optimization problem is reduced to solving a system of non-linear equations for \( \lambda \). Newton’s method can therefore be used to solve Eqs. (??) by setting \( \lambda \) equal to \( 0 \), updating \( r \) through Eqs. (??), and then by equating \( r_u \) to \( r \). This procedure is iterative i.e. it is repeated until the change from iteration to iteration falls below a specified tolerance. The Jacobian \( J_{\lambda} \) of Eq. (??) is found to be

\[
J_{\lambda} |_{\lambda=0} = 2D(Ar_d) AJ_t^{t}.
\]

(15)
Solving the linear system, however, proves to be impractical due to the high condition number of $J_\lambda$. Shifting the matrix by a factor of $\alpha$ significantly reduces the condition number and makes the problem tractable. Newton’s iteration becomes

$$(J_\lambda + \alpha I)\lambda = -c(0),$$

with $I$ being the identity matrix. Given that $\lambda$ will eventually vanish if the system converges to the correct solution, the shift term $\alpha I$ does not alter the numerical solution but makes the procedure tractable. With this scheme, only one linear system is solved per Newton iteration. A Krylov subspace iterative solver based on the Conjugate Gradient method is used to solve Eq. (16).

Since the Jacobian matrices ($J_r$ and $J_\lambda$) contain a large number of zero entries (see Figure 3), they were stored in compressed sparse row format. In sparse storage, a low density matrix is compressed into a set of three vectors. The first stores the non-zero values, while the second and third store the row and column indices, respectively.

### Implementation

#### Parallelization

The matrix form of Eqs. (16, 22, 29) and the size of the biological systems of interest (virus capsid, ribosomes, etc.) make the algorithm a good candidate for parallelization. In the current implementation, the algorithm is parallelized for distributed memory systems. Since the most expensive part of the algorithm is the numerical solution of the linear system (Eq. (16)), the sparse matrices $A$, $J_r$, and $J_\lambda$ were constructed or assembled in parallel. This was done with the aid of PETSC (portable extensible scientific toolkit) library, which uses message passing interface (MPI) to perform linear algebra subroutines in parallel. The library supports sparse storage of matrices and vectors distributed on multiple nodes. In particular, an MPI matrix is partitioned into a set of sequential matrices, each of which is stored locally on one specific node. The partitioning scheme used is
shown in Figure 5. MPI vectors such as the coordinates $\mathbf{r}$ and the lagrange multipliers $\lambda$ were stored distributively on all processors. With the input read on node 0 and parallelized across all available processors (see Figure 4), the algorithm proceeds by constructing the RHS of Eq. (??) and the Jacobian $\mathbf{J}_r$, assembling the Jacobian $\mathbf{J}_\lambda$, and then solving the linear system with a scalable iterative solver. Since $\mathbf{J}_\lambda$ is symmetric, and the shifting factor makes it positive definite, the Conjugate Gradient method (KSPCG in PETSC) was used to solve the linear system of Eq. (??).

Results and discussion

In order to assess the accuracy, efficiency, and scalability of the constrained mapping algorithm, three systems were simulated in vacuum. The first is Lactoferrin (ID 1LFG) undergoing a structural transition\cite{13}. It was experimentally shown that this iron binding protein has two free energy minimizing conformations\cite{35}: diferric with closed proximal lobes (PDB code 1LFG), and apo with open ones (PDB code 1LFH). Here, we start with an open lactoferrin structure and simulate its closing in vacuum. The second system is Nωv (ID 1OHF) triangular structure\cite{36} shrinking after equilibration in an explicit solvent at pH = 5.4, and the third is the native CCMV viral capsid (ID 1CWP) shrinking due to strong protein-protein interactions. All three systems were previously simulated\cite{13} using DMS\cite{3,13,26,27} and NAMD\cite{37} under NVT conditions. Here we show that the constrained map captures the system dynamics accurately (Figures 8-10) and with a greater efficiency (Figure 7) over both DMS and conventional MD.

Conclusion

We have shown that the coarse-grained to all-atom information gap can be decreased by taking bond lengths and angles into account when constructing atom-resolved states from CG variables. The mapping algorithm proposed in this paper provided efficiencies not only over traditional MD but also over the multiscale approach without constraints. These efficiencies will enable the simulation of mesoscopic systems such as viruses over longer periods of time. Furthermore, the mapping
algorithm is flexible so that future implementations can also account for experimental nanocharacterization data (e.g., atomic force microscopy, time of flight techniques, and micro-fluidics).

Appendix

Construction of CG variables from atomic positions

In the Formulation section, we modified Eq. (??) by imposing a set of constraints on the bond lengths and angles. We showed that the solution to the optimization problem can be obtained in terms of a vector of Lagrange multipliers (Eq. (??)). Here we show that this approach does not modify the CG variables derived from the unconstrained map.\(^\text{26}\) We begin by constructing \( \phi \) from \( r \) by doing a mass-weighted least-squares minimization of \( \sigma \) with respect to \( \phi \), i.e.

\[
\min_\phi \frac{1}{2} (M\sigma)' \sigma.
\]  

Here \( M \) is a diagonal matrix of all atomic masses. Since \( \sigma = r - U\phi - J_r^t\lambda \), the solution to Eq. (??) is

\[
MU'U\phi = MU' (r - J_r^t\lambda) \]

However, in constructing \( \phi \), the all-atom configuration used is from MD, which implies that the constraints are satisfied. The Jacobian \( J_r \) vanishes, and the CG variables are therefore unaltered.

The adjacency-like matrix

For a simple triatomic non-linearly bonded molecule (see Figure \[\text{6}\]), the adjacency-like matrix \( A \) defined in the Formulation section takes the form shown in Figure \[\text{11}\]. Here a total of three constraints must be taken into account for three bonded atoms.
Acknowledgement

This project was supported in part by the National Science Foundation (Collaborative Research in Chemistry Program), National Institutes of Health (NIBIB), METAcyt through the Center of Cell and Virus Theory, Indiana University College of Arts and Sciences, and the Indiana University information technology services (UITs) for high performance computing resources.

References

(1) Ortoleva, P. J. J. Chem. Phys. 2005, 109, 2770–2771.

(2) Pankavich, S.; Shreif, Z.; Ortoleva, P. J. Physica A 2008, 387, 4053–4069.

(3) Cheluvaraja, S.; Ortoleva, P. J. J. Chem. Phys. 2010, 132, 75102–75110.

(4) Singharoy, A.; Cheluvaraja, S.; Ortoleva, P. J. J. Chem. Phys. 2011, 134, 44104–44120.

(5) Singharoy, A.; Sereda, Y.; Ortoleva, P. J. J. Chem. Theory Comput. 2012, 8, 1379–1392.

(6) Jaqaman, K.; Ortoleva, P. J. J. Comput. Chem. 2002, 23, 484–491.

(7) Murtola, T.; Falck, E.; Karttunen, M.; Vattulainen, I. J. Chem. Phys. 2007, 126, 75101–75114.

(8) Shiha, A. Y.; Arkhipov, A.; Freddolino, P. L.; Schulten, K. J. Phys. Chem. B 2006, 110, 3674–3684.

(9) Shiha, A. Y.; Freddolinoa, P. L.; Arkhipova, A.; Schulten, K. J. Struct. Biol. 2007, 157, 579–592.

(10) Bahar, I.; Atilgan, R. A.; Erman, B. Folding and Design 1997, 2, 173–181.

(11) Haliloglu, T.; Bahar, I.; Erman, B. Phys. Rev. Letters 1997, 79, 3090–3093.

(12) Miao, Y.; Johnson, J. E.; Ortoleva, P. J. J. Phys. Chem. B 2010, 114, 11181–11195.

(13) Mansour, A. A.; Ortoleva, P. Preprint http://arxiv.org/abs/1305.6096 2013,
(14) Iserles, A. *A first course in the Numerical Analysis of differential equations*; Cambridge University Press, 2008.

(15) Miyamoto, S.; Kollman, P. A. *J. Comput. Chem.* **1992**, *13*, 952–962.

(16) Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. *J. Comput. Phys.* **1992**, *23*, 327–341.

(17) Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. *Molecular Simulation* **1994**, *13*, 395–405.

(18) Krautler, V.; van Gunsteren, W. F.; Hunenberge, P. H. *J. Comput. Chem.* **2001**, *22*, 501–508.

(19) Forester, T. R.; Smith, W. *J. Comput. Chem.* **1998**, *19*, 102–111.

(20) McBride, C.; Wilson, M. R.; Howard, J. A. K. *Molecular Physics* **1998**, *93*, 955–964.

(21) Andersen, H. C. *J. Comput. Phys.* **1983**, *52*, 24–34.

(22) Lee, S.-H.; Palmo, K.; Krimm, S. *J. Comput. Phys.* **2005**, *210*, 171–182.

(23) Gonnet, P. *J. Comput. Phys.* **2007**, *220*, 740–750.

(24) Hess, B.; Bekker, H.; Berendsen, H. J. C.; Fraaije., J. G. E. M. *J. Comput. Chem.* **1997**, *18*, 1463–1472.

(25) Joshi, H.; Singharoy, A.; Sereda, Y. V.; Cheluvaraja, S. C.; Ortoleva, P. J. *Prog. Biophys. Mol. Biol.* **2011**, *107*, 200–217.

(26) Singharoy, A.; Joshi, H.; Ortoleva, P. J. *J. Chem. Inf. Model.* **2012**, *52*, 2638–2686.

(27) Pankavich, S.; Miao, Y.; Ortoleva, J.; Shreif, Z.; Ortoleva, P. J. *J. Chem. Phys.* **2008**, *128*, 234908–234921.

(28) Chartrand, G. *Introductory Graph Theory*; Dover Publications, 1985.

(29) Nocedal, J.; Wright, S. *Numerical Optimization*; Springer, 2006.

(30) Barth, E.; Kuczera, K.; Leimkuhler, B.; Skeel, R. D. *J. Comput. Chem.* **2004**, *16*, 1192–1209.
(31) Davis, T. A. *Direct Methods for Sparse Linear Systems*; SIAM, 2006.

(32) Balay, S.; Gropp, W. D.; McInnes, L. C.; Smith, B. F. Efficient Management of Parallelism in Object Oriented Numerical Software Libraries. 1997.

(33) Balay, S.; Brown, J.; Buschelman, K.; Eijkhout, V.; Gropp, W. D.; Kaushik, D.; Knepley, M. G.; McInnes, L. C.; Smith, B. F.; Zhang, H. *PETSc Users Manual*; 2013.

(34) Balay, S.; Brown, J.; Buschelman, K.; Gropp, W. D.; Kaushik, D.; Knepley, M. G.; McInnes, L. C.; Smith, B. F.; Zhang, H. PETSc Web page. 2013; http://www.mcs.anl.gov/petsc.

(35) Norris, G. E.; Anderson, B. F.; Baker, E. N. *Acta Crystallogr. Sect. B* 1991, 47, 998–1004.

(36) Taylor, D. J.; Wang, Q.; Bothner, B.; Natarajan, P.; Finn, M. G.; Johnson, J. E. *J. Chem. Commun.* 2003, 22, 2770–2771.

(37) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kale, L.; Schulten, K. *J. Comput. Chem.* 2005, 26, 1781–1802.

(38) Miao, Y.; Ortoleva, P. *J. Comput. Chem.* 2006, 125, 44901–44908.
Figure 1: A typical energy profile for an atomistic configuration constructed without respecting the bond lengths and angles; a substantial amount of time is spent doing energy minimization to bring the bond and angle energies back to their equilibrium values.
Figure 2: By imposing constant bond lengths and angles when constructing the atomistic configuration from a given CG state, the bond, angle, and total energies rapidly converge to their thermal averages without the need for extensive energy minimization. Therefore, constrained DMS (CDMS) performs a single CG time step faster than DMS.
Figure 3: The sparsity pattern of $\mathbf{J}_\lambda$ is shown for Lactoferrin. In total, there are 29,923 constraints to be satisfied. The total number of non-zero values is 388,311, which makes the sparsity equal to 99.956%.
Figure 4: A diagram that shows the input (atomic coordinates, equilibrium bond and angle-determining distances, and the constraint atomic indices) read on node 0 and then this information is distributed in parallel on all remaining \( n \) nodes. This allows the construction of the adjoint-like matrix \( A \) and the Jacobian \( J_R \), and the assembly of the Jacobian \( J_\lambda \), in parallel.
Figure 5: An $N \times m$ matrix is stored in parallel by distributing it on $n$ nodes, each storing a sequential matrix of size $N_{\text{local}} \times m$. If $N$ is divisible by $n + 1$, then $N_{\text{local}} = N/(n + 1)$.

Figure 6: A triatomic non-linear molecule with two bonds and one angle. A total of three constraints must be taken into account for this molecule.
Figure 7: Speedup of both DMS and CDMS over conventional MD. The efficiency of the constraint algorithm increases with the complexity of the system.

Figure 8: The radius of gyration of Lactoferrin decreases in time as the protein transits from its open to closed state.
Figure 9: The radius of gyration of the triangular structure of \( N\omega v \) decreases as a function of time.

Figure 10: The radius of gyration of the native CCMV capsid decreases as a function of time due to strong protein-protein interactions.
\[ A = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{pmatrix} \]

Figure 11: The columns specify the three atomic indices \( i, j, \) and \( k, \) while each row corresponds to a specific constraint: the first is for the bond between atoms \( i \) and \( j, \) the second is for the bond between atoms \( j \) and \( k, \) while the third is for the three-body angle constrained by the distance \( d_{ik}. \)