JCTC Journal of Chemical Theory and Computation

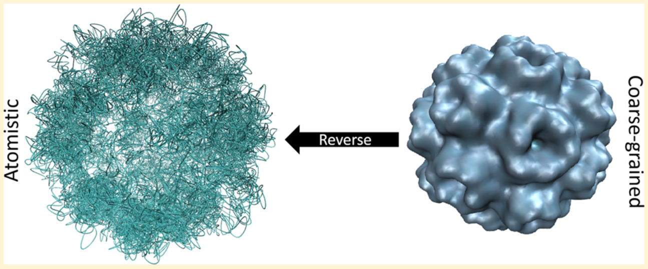
Article

pubs.acs.org/JCTC

Reverse Coarse-Graining for Equation-Free Modeling: Application to Multiscale Molecular Dynamics

Andrew Abi Mansour and Peter J. Ortoleva*

Department of Chemistry and Center for Theoretical and Computational Nanoscience, Indiana University, Bloomington, Indiana 47405, United States



ABSTRACT: Constructing atom-resolved states from low-resolution data is of practical importance in many areas of science and engineering. This problem is addressed in this article in the context of multiscale factorization methods for molecular dynamics. These methods capture the crosstalk between atomic and coarse-grained scales arising in macromolecular systems. This crosstalk is accounted for by Trotter factorization, which is used to separate the all-atom from the coarse-grained phases of the computation. In this approach, short molecular dynamics runs are used to advance in time the coarse-grained variables, which in turn guide the all-atom state. To achieve this coevolution, an all-atom microstate consistent with the updated coarse-grained variables must be recovered. This recovery is cast here as a nonlinear optimization problem that is solved with a quasi-Newton method. The approach yields a Boltzmann-relevant microstate whose coarse-grained representation and some of its fine-scale features are preserved. Embedding this algorithm in multiscale factorization is shown to be accurate and scalable for simulating proteins and their assemblies.

1. 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 these systems must somehow account for the coevolution of coarse-grained (CG) and microscropic (atomistic) variables. Multiscale coevolution,[1−5] an equation-free method,[6−9] has been proposed as an alternative to purely coarse-grained methods;[10−15] coevolution methods operate via a cycle consisting of microscopic and coarse-grained phases. These methods do not involve deriving CG dynamical equations in closed form. Instead, the CG dynamics follow directly from the microscopic dynamics. In contrast, traditional CG methods evolve large-scale structural variables via phenomenological equations,[10,11,13] and they do not provide information on the evolving microstate(s).

The focus of this study is multiscale factorization,[16,17] an equation-free coevolution method applied to molecular dynamics (MD) for macromolecular systems such as proteins and their assemblies. A necessary condition for an efficient multiscale simulation is the separation of time scales between the atomistic fluctuations and coherent, slow changes captured by the CG variables.[2−4,18,19] Furthermore, the MD phase of the multiscale computation should be sufficiently long to generate a representative ensemble of fluctuations in the CG momenta (i.e., longer than the "stationarity time"[16]). To complete the multiscale cycle, a microstate consistent with the updated CG variables must be constructed before the MD phase of the computation is resumed. This multiscale approach is summarized in the flowchart shown in Figure 1. In this way, the entire multiscale computation follows directly from an interatomic force field and avoids the need for introducing phenomenological CG governing equations and the uncertainty associated with them. While there has been recent progress on coevolving such multiscale systems in time,[5,20] recovering a microstate consistent with the CG variables remains a challenge because it is an ill-posed problem[21,22] characterized by an information gap between the CG and fine-grained (FG) descriptions. This reverse coarse-graining is cast here as a nonlinear optimization problem the solution to which enables an efficient and scalable fine-graining algorithm that is hereafter referred to as microstate sparse reconstruction (MSR). In this

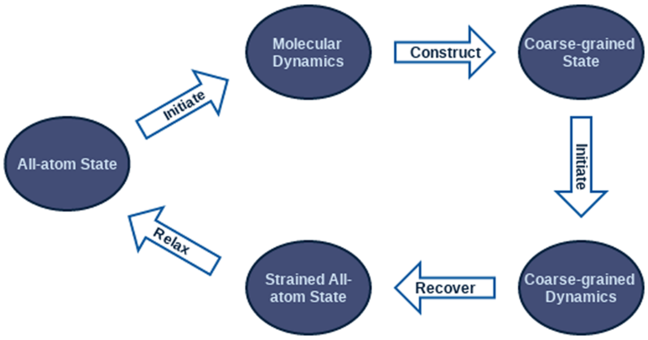


Figure 1. In multiscale coevolution methods, a molecular dynamics run is initiated to collect information needed to advance the coarse-grained state in time. Afterward, the all-atom state is recovered to begin another dual-phase step.

Received: April 6, 2016
Published: September 15, 2016

DOI: 10.1021/acs.jctc.6b00348
J. Chem. Theory Comput. 2016, 12, 5541−5548

article, it is shown how embedding MSR in MF yields an accurate and scalable method for simulating macromolecular systems.

The mathematical framework for MSR is outlined in Section 2, and its implementation for distributed systems is described in Section 3. MSR is demonstrated for several macromolecular systems in Section 4, and conclusions are drawn in Section 5.

2. THEORY

For an efficient multiscale simulation, it is necessary to reduce the all-atom description represented by N atoms to a set of N_{CG} variables ($N_{\mathrm{CG}} \ll N$) that capture the coherent deformation of the system. These CG variables must be chosen such that they evolve on a time scale much greater than that of fluctuating atoms. Let ϕ denote a set of CG variables such that

$$\phi_\alpha = \mathbf{Q}\mathbf{r}_\alpha \tag{1}$$

where $\mathbf{r}_\alpha$ is a vector of all atomic positions with α corresponding to the x, y, or z axis and $\mathbf{Q}$ is a matrix of dimensions $N_{\mathrm{CG}} \times N$ that depends on the atomic positions of a reference configuration denoted $\mathbf{r}^0$. Initially, the reference structure introduces a configuration determined by data collected from X-ray, cryo-EM, or other experimental techniques. However, at later times, the reference structure is taken from a previous time step in a discrete time evolution sequence. Thus, the CG variables specify how the structure is deformed from this reference configuration in the course of the simulation. While coarse-graining can be uniquely defined for a system, the inverse problem of finding a microstate from a given CG description is ill-posed and therefore has no unique solution. This problem is addressed below.

2.1. Microstate Reconstruction.

A challenge in multiscale equation-free methods is recovering an all-atom configuration consistent with the updated CG variables. This is formulated here as an optimization problem that minimizes the norm of the difference between $\mathbf{r}_\alpha$ and $\mathbf{r}_\alpha^0$ over all three Cartesian components, subject to the constraints imposed by the updated CG variables (eq 1). Thus, the CG constraints act as a perturbation that guides the all-atom microstate to a configuration which minimizes deviation from the reference configuration and is consistent with the imposed CG description. Unless the CG variables have not significantly evolved in time, the new microstate should not be the same as the reference structure. In the former case, either the CG time step is too small or the macromolecular system is exploring an ensemble of microstates lying close to that of the previous time step.

To take microstate effects such as those imposed by stiff bonds into account, FG constraints are included in order to enforce constant bond lengths and harmonic angles (Figure 2). This preserves key aspects of the microstructure.

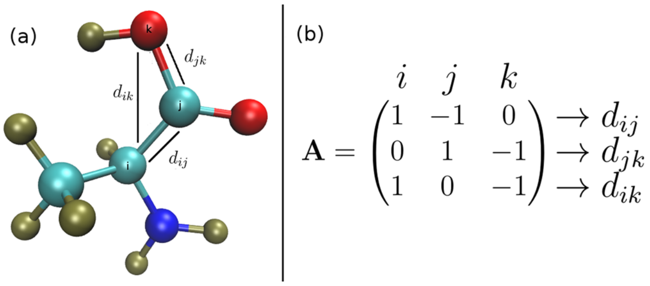
The optimization problem is formulated in terms of a quadratic function as follows

$$\min_{\mathbf{r}_\alpha} f(\mathbf{r}) = \frac{1}{2} \sum_{\alpha'} (\mathbf{r}_{\alpha'} - \mathbf{r}_{\alpha'}^0)^T (\mathbf{r}_{\alpha'} - \mathbf{r}_{\alpha'}^0) \tag{2}$$

which is subject to the following constraints

$$\phi_\alpha - \mathbf{Q}\mathbf{r}_\alpha = \mathbf{0} \tag{3}$$

$$\sum_\alpha D(\mathbf{A}\mathbf{r}_\alpha)(\mathbf{A}\mathbf{r}_\alpha) - D(\mathbf{l}_\alpha)\mathbf{l}_\alpha = \mathbf{0} \tag{4}$$



$$\mathbf{A} = \begin{array}{c} \begin{array}{ccc} i & j & k \end{array} \\ \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{pmatrix} \begin{array}{l} \to d_{ij} \\ \to d_{jk} \\ \to d_{ik} \end{array} \end{array}$$

Figure 2. Adjacency-like matrix $\mathbf{A}$ (b) for the three atoms labeled i, j, and k shown in (a). To preserve the two bond lengths and harmonic angle of the triplet of atoms (i–j–k), the three interatomic distances d_{ij}, d_{jk}, and d_{ik} are constrained to their equilibrium values extracted from MD.

where $D(\mathbf{v})$ denotes a diagonal matrix whose entries are equal to those of vector $\mathbf{v}$, $\mathbf{l}_\alpha$ represents a vector of interatomic distances (for the atomic bonds and harmonic angles) computed from the MD phase in every α direction, and $\mathbf{A}$ is an adjacency-like matrix that captures the location of each atomic index in every equation of the FG constraints, i.e., for the kth constraint spanning atoms i and j, row k in $\mathbf{A}$ has +1 entry at column i and -1 at column j, whereas all remaining columns have zero entries (Figure 2). For convenience, the interatomic distance squared was used in the equations of the FG constraints.

The Lagrangian $\mathcal{L}$ of the above optimization problem is

$$\mathcal{L}(\mathbf{r}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = f(\mathbf{r}) + \sum_\alpha \mu_\alpha^T (\phi_\alpha - \mathbf{Q}\mathbf{r}_\alpha) + \lambda^T \sum_\alpha (D(\mathbf{A}\mathbf{r}_\alpha)(\mathbf{A}\mathbf{r}_\alpha) - D(\mathbf{l}_\alpha)\mathbf{l}_\alpha) \tag{5}$$

The Lagrange multipliers represented by the vector $\boldsymbol{\mu}_\alpha$ ensure the recovered microstate is consistent with the updated CG state in every α direction, and those represented by the vector $\boldsymbol{\lambda}$ enforce constant bond lengths and harmonic angles. While the Lagrangian does not necessarily admit a unique minimum, this does not contradict the physics of the problem since there is an ensemble of microstates consistent with a given CG description. The numerical scheme for minimizing the Lagrangian in eq 5 is covered in the next section.

3. IMPLEMENTATION

MSR was implemented using ProtoMD,[23] a prototyping toolkit written in python for multiscale MD. The parallelization of the algorithm was done with the aid of PETSc[24,25] (Portable, Extensible Toolkit for Scientific computation), whereas SWIG[26] (Simplified Wrapper and Interface Generator) was used to interface the C++ modules (that use PETSc) with the python code (ProtoMD source code). The MSR code is freely available on github.[27] The vdW forces were modeled using a Lennard-Jones potential that is slightly modified by shifting the interatomic distance by 1 Å to prevent the repulsive term from diverging to infinity when this distance approaches 0.

3.1. Coarse-Graining.

The space-warping method is used as a dimensionality reduction technique[28,29] in this study because the CG variables obtained with this method are slowly varying in time for the systems considered here. In this method, the mapping matrix $\mathbf{Q}$ in eq 1 is constructed from a set of products of three Legendre polynomials that are functions of the x, y, and z positions of the reference configuration of orders

k_x, k_y, and k_z, respectively. The total order of the method is k_m such that $k_\mathrm{m} \geq k_x + k_y + k_z$. A brief review of the space-warping method and the particular form of $\mathbf{Q}$ used here is covered in Appendix A.

3.2. Regularizing the Lagrangian. Using Newton's method to minimize the Lagrangian in eq 5 is not possible because its Hessian is ill-conditioned.[30] Instead, an L_2 regularization (Appendix B) is imposed to obtain an approximate numerical solution. First, the Lagrangian is recast in the form

$$\mathcal{L}(\mathbf{r}) = f(\mathbf{r}) + \sum_\alpha \mu_\alpha^T(\phi_\alpha - \mathbf{Q}\mathbf{r}_\alpha) + \lambda^T \sum_\alpha \left(D(\mathbf{A}\mathbf{r}_\alpha)(\mathbf{A}\mathbf{r}_\alpha) - D(\mathbf{l}_\alpha)\mathbf{l}_\alpha\right) + \frac{1}{2}\beta^2\lambda^T\lambda \tag{6}$$

where β is a regularization parameter set to 1 and the penalty term $\lambda^T\lambda$ keeps the norm of λ to a minimum. The Lagrangian in eq 6 is minimized by setting its gradient to zero with respect to the atomic positions and Lagrange multipliers. This yields

$$\mathbf{r}_\alpha = \mathbf{r}_\alpha^0 - \mathbf{J}_{\mathbf{r}_\alpha}^T\lambda + \mathbf{Q}^T\boldsymbol{\mu}_\alpha \tag{7}$$

$$\boldsymbol{\phi}_\alpha = \mathbf{Q}\mathbf{r}_\alpha \tag{8}$$

$$\sum_\alpha D(\mathbf{A}\mathbf{r}_\alpha)(\mathbf{A}\mathbf{r}_\alpha) = \sum_\alpha D(\mathbf{l}_\alpha)\mathbf{l}_\alpha - \beta^2\lambda \tag{9}$$

where $\mathbf{J}_{\mathbf{r}_\alpha}$ is the Jacobian of eq 9 with respect to the atomic positions, and it is given by

$$\mathbf{J}_{\mathbf{r}_\alpha} = 2 \times D(\mathbf{A}\mathbf{r}_\alpha)\mathbf{A} \tag{10}$$

Equations 7–9 are solved with a quasi-Newton method in a way analogous to MD constraint algorithms.[31] This is achieved by decoupling the Lagrange multipliers from the atomic positions. First, eq 7 is recast in terms of the unconstrained atomic positions vector $\mathbf{r}_\alpha^u$ (which is initially set to $\mathbf{r}_\alpha^0$) such that

$$\mathbf{r}_\alpha = \mathbf{r}_\alpha^u - \mathbf{J}_{\mathbf{r}_\alpha^u}^T\lambda + \mathbf{Q}^T\boldsymbol{\mu}_\alpha \tag{11}$$

The Jacobian $\mathbf{J}_{\mathbf{r}_\alpha^u}$ is evaluated at $\mathbf{r}_\alpha = \mathbf{r}_\alpha^u$. The Lagrange multipliers are then decoupled and updated separately via

$$\mathbf{J}_\lambda\lambda = \sum_\alpha D(\mathbf{l}_\alpha)\mathbf{l}_\alpha - D(\mathbf{A}\mathbf{r}_\alpha^u)(\mathbf{A}\mathbf{r}_\alpha^u) \tag{12}$$

$$\mathbf{Q}\mathbf{Q}^T\boldsymbol{\mu}_\alpha = \boldsymbol{\phi}_\alpha - \mathbf{Q}\mathbf{r}_\alpha^u \tag{13}$$

Using the chain rule, the Jacobian $\mathbf{J}_\lambda$ is found to be

$$\mathbf{J}_\lambda = -\sum_\alpha \mathbf{J}_{\mathbf{r}_\alpha^u}\mathbf{J}_{\mathbf{r}_\alpha^u}^T + \beta^2\mathbf{I} \tag{14}$$

Once eqs 12 and 13 are solved, the Lagrange multipliers are used in eq 11 to update the atomic positions vector $\mathbf{r}_\alpha$. The unconstrained positions vector $\mathbf{r}_\alpha^u$ is then equated to $\mathbf{r}_\alpha$, the Lagrange multipliers are set to zero, and the procedure is repeated until the maximum atomic displacement is below 10^{-2} Å. This is summarized in Algorithm 1.

3.3. Sparse Storage. The efficiency and scalability of MSR stem from the sparse structure of the Jacobians $\mathbf{J}_{\mathbf{r}_\alpha}$ and $\mathbf{J}_\lambda$. Thus, these matrices are stored in compressed sparse row format[32] using PETSc.[24,25] For example, the sparsity pattern for $\mathbf{J}_\lambda$ is shown in Figure 3 for lactoferrin protein.[33] The system consists of 10 560 atoms and is characterized by 10 674 bonds and 19 249 harmonic angles. The size of $\mathbf{J}_\lambda$ is therefore 29 923 × 29 923. As shown in Figure 3, $\mathbf{J}_{\mathbf{r}_\alpha}$ is almost completely sparse.

```
while $error \geq tol$ do
    Construct $\mathbf{J}_{\mathbf{r}_\alpha^u}$ $\forall\ \alpha$;
    Assemble $\mathbf{J}_\lambda$;
    Compute $\lambda$ and $\mu$ by solving Eqs. (12 - 13);
    Update atomic positions via Eq. (11);
    Compute error;
end
```

Algorithm 1: MSR updates the atomic positions and Lagrange multipliers in an alternating way such that both the CG and FG constraints are satisfied to within a certain tolerance.

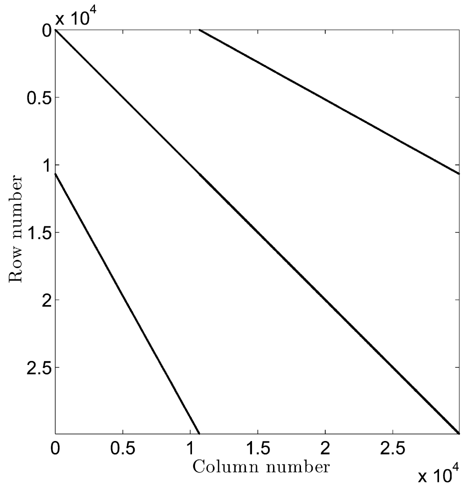


Figure 3. Sparsity pattern of $\mathbf{J}_\lambda$ for lactoferrin. In total, there are 29 923 FG constraints and 59 846 non-zero entries, which makes $\mathbf{J}_\lambda$ 99.99% sparse.

3.4. Parallelization. The size of the biological systems of interest (such as virus-like particles or assemblies of proteins) makes MSR a good candidate for parallelization. In the current implementation, MSR is parallelized for distributed memory systems. This was done with the aid of PETSc,[24,25] which uses message passing interface (MPI)[34] to perform linear algebra computations in parallel. The library supports sparse storage for matrices distributed on multiple nodes. Once the input coordinates and topology indices are distributed on all processors, the algorithm proceeds by constructing the RHS of eqs 12 and 13 and the Jacobians $\mathbf{J}_{\mathbf{r}_\alpha}$, assembling the Jacobian $\mathbf{J}_\lambda$, and then solving eqs 12 and 13. A direct Choleski solver was used to solve eq 13, whereas an iterative solver based on the improved stabilized version of the biconjugate gradient squared method (IBiCGStab in PETSc) was used to solve eq 12 with the incomplete LU (PCILU in PETSc) chosen as a preconditioner.

4. RESULTS AND DISCUSSION

Pertussis toxin (PDB code 1PRT)[35] was used as a demonstration system to assess the accuracy and efficiency of MSR. This protein was simulated under NVT conditions at 300 K using the CHARMM22 force field[36] and the TIP3P water model.[37] NaCl counterions of concentration 0.15 M were added for charge neutrality. The system consisted of 603 775 atoms in a box of dimensions 16 nm × 16 nm × 24 nm. An equilibration run with position restraints imposed on the protein was performed for 100 ps; after thermal equilibrium was established, the system was simulated without any

DOI: 10.1021/acs.jctc.6b00348

restraints for 3.2 ns during which the protein underwent a conformational change (Figure 4).

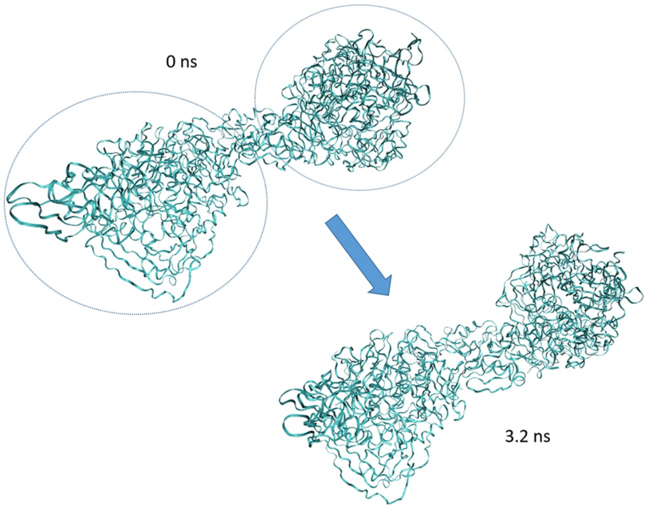


Figure 4. Pertussis toxin (PRT) protein undergoes a conformational change under NVT conditions: its two lobes (circled) shrink in time as the protein contracts in aqueous solution of salinity equal to 0.15 M.

4.1. Fine-Grained Constraint Error.

The error from the FG constraints (eq 4) was assessed as follows. First, the space-warping method[28,29] was used to coarse-grain the protein at t = 0 ps using Legendre polynomials of maximum order $k_m = 5$. A microstate was then recovered using the method outlined in ref 28 (Appendix A). The recovered microstate was characterized with high bond energies that had to be annealed using extensive energy minimization (the steepest descent method was employed for demonstration), followed by thermalization to bring bond energies to values consistent with the thermal conditions. In contrast, MSR recovers a microstate with modest bond and harmonic angle energies without the need for thermalization and within a fewer number of iterations. This is demonstrated in Figure 5, which shows the FG error (taken to be the absolute value of the left-hand side of eq 4) rapidly vanishes. Another characterization of the microstructure used here is the radial distribution function, $g(r)$, computed for pertussis toxin (Figure 6). MSR reproduces the RDF with good accuracy: $g(r)$ is characterized by two sharp peaks at 1.1047 and 1.6030 Å, and it vanishes below a distance of separation of 1 Å.

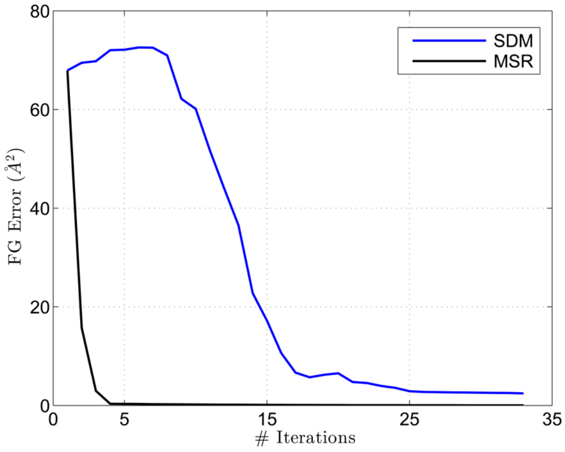


Figure 5. Convergence of the fine-grained (FG) error in eq 4 is linear. This error drops below 0.1 Å within five iterations using MSR (in black). In contrast, minimizing the potential energy using the steepest descent method (SDM) takes more iterations to bring the FG error close to 2 Å (in blue).

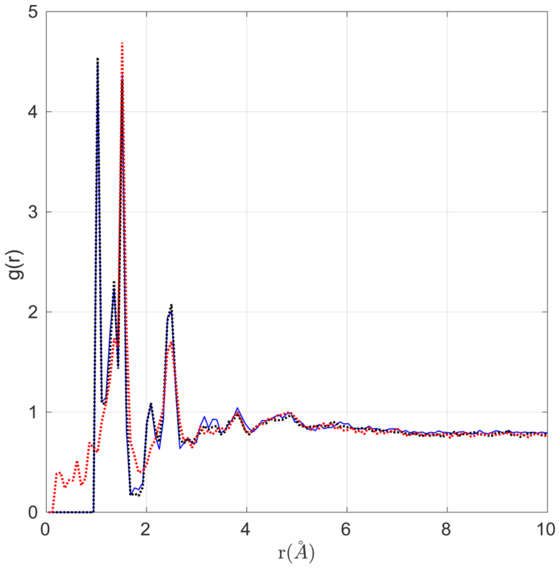


Figure 6. Radial distribution function (RDF) for the microstructure of pertussis toxin recovered at t = 100 ps using the center of mass of the protein as a course-grained (CG) variable. The RDF was computed from molecular dynamics (MD) simulation (blue curve), microstate sparse reonstruction (MSR; black dotted curve), and the space wapring method (SWM; red dotted curve). MSR reproduces the RDF accurately at all radial distances because it preserves the atomic bonds and harmonic angles, whereas the RDF computed with SWM deviates from that of MD at small distances of separation, as expected since this method is built on CG variables alone.

4.2. Coarse-Grained Constraint Error.

Convergence in the error of the CG constraints (eq 3) was assessed by taking a microstate of pertussis toxin at t = 10 ps and introducing noise (a random number between −1 and 1) to all atomic positions. The potential energy of the microstate before it was perturbed was approximately -1.04×10^5 kJ/mol, and after perturbation, its potential energy increased to approximately 3.29×10^9 kJ/mol. The space-warping method was then used to coarse-grain the unperturbed microstate using linear Legendre polynomials ($k_m = 1$). The constructed CG variables and the bond lengths and harmonic angles computed from MD were then used as input for MSR, which rapidly recovers a microstate consistent with the imposed constraints (Figure 7). Figure 8 shows the potential energy difference ($U - U_{min}$) of the recovered microstate is close to that of MD. The reference potential energy U_{min} was set to -1.16×10^5 kJ/mol.

4.3. Completeness of Coarse-Grained Description.

In order to analyze the variation in accuracy of MSR due to changes in the number of CG variables included, the RMSD of pertussis toxin with respect to the initial structure (Figure 9) was computed for all protein atoms and using various orders and numbers of Legendre polynomials (denoted k_m). A time series of the RMSD of the protein was generated using MD, with the structure of each frame aligned to the initial structure. The space-warping variables were then used to coarse-grain the protein, and a new microstate consistent with the CG variables was recovered using MSR and then compared to that obtained from MD. The metric chosen for the fine-graining error is the

DOI: 10.1021/acs.jctc.6b00348
J. Chem. Theory Comput. 2016, 12, 5541−5548

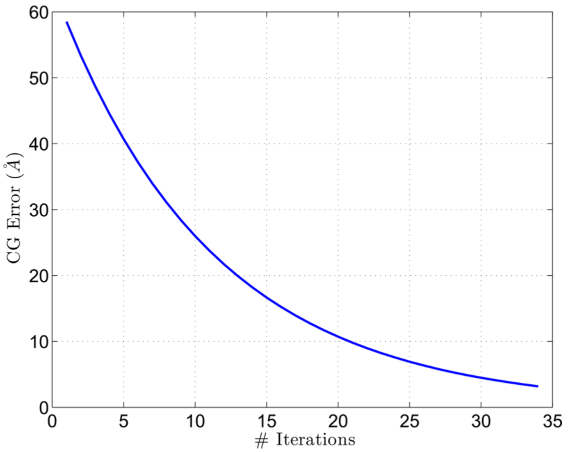


Figure 7. In MSR, convergence of the coarse-grained (CG) error in eq 3 is linear. This error drops close to 3 Å in 34 iterations.

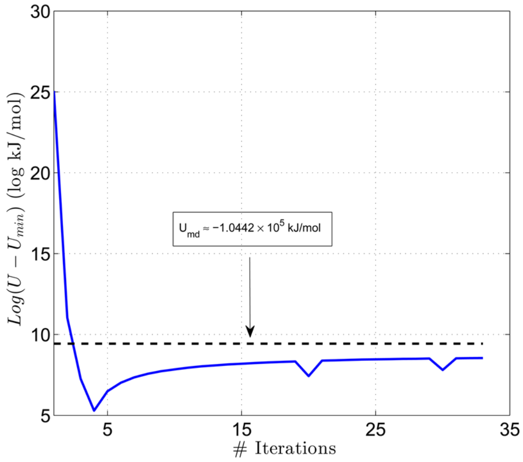


Figure 8. Potential energy computed with MSR for pertussis toxin rapidly converges to a value close to that of MD.

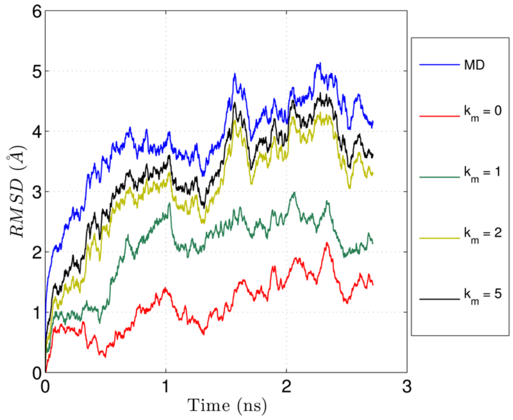


Figure 9. Error in RMSD decreases as the number of CG variables increases (represented by k_m), indicating that higher order space-warping variables capture finer scale features of the protein.

difference between the RMSD of the protein obtained from MD and that obtained from MSR. As k_m is increased, the number of CG variables increases, and the fine-graining error decreases as expected. However, beyond quadratic Legendre polynomials ($k_m = 2$), the rate of convergence in the RMSD error becomes slow for this problem. An alternative way of significantly accelerating this rate is by updating the reference structure. In practice, the latter requires reconstruction of the mapping matrix in eq 1, which can be computationally demanding for large systems represented by a high number of CG variables. Let ν represent the frequency of updating the reference structure (i.e., the reference structure is updated every ν CG steps, with each step set to 1 ps); then, the fine-graining error is expected to be proportional to ν. This is demonstrated in Figure 10, which shows that the fine-graining error increases as ν is increased from 50 to 200.

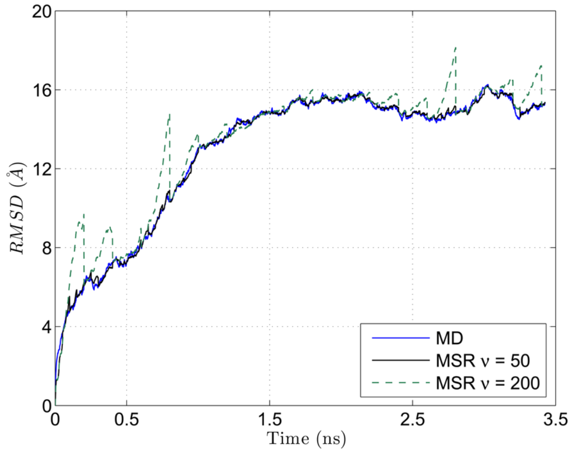


Figure 10. MSR reproduces the RMSD of pertussis toxin with relatively small error for $\nu = 50$. As ν is increased to 200, the error in RMSD significantly increases.

4.4. Scalability.

To analyze the parallel performance and scalability of MSR, a cowpea chlorotic mottle virus capsid (PDB code 1CWP) was used as a demonstration system (Figure 11). The capsid was generated using BIOMT transformations.[38] This virus-like particle (VLP) supports a structure consisting of 450 840 atoms. Linear space-warping variables (using $k_m = 1$) were then computed for the VLP before it was perturbed by adding noise (random number between −10 and 10) to all of its atomic positions. The microstate was then recovered in 10 MSR iterations for a total of 1 348 680 FG constraints using an increasing number of cores. Simulations were performed on Indiana University's Karst cluster, using a dual Intel Xeon E5-2650 v2 8-core processor and a total of 32 GB of RAM. MSR shows promising strong scalability over a total of 16 cores (Figure 12). Further optimization of the algorithm using OpenMP or GPU-based acceleration should lead to higher speedups.

5. CONCLUSIONS

Microstate reconstruction is a key element of multiscale MD algorithms. MSR, an efficient method that constructs microstates consistent with the evolved CG description and thermal conditions is presented and demonstrated for proteins and their assemblies. Using these microstates to generate dynamical information needed to update the CG state in MF yields accurate and efficient multiscale simulation of molecular systems. MSR can be further improved by taking vdW interactions into account when reconstructing the microstate. This can be achieved by incoporating historical information (such as several microstates obtained from MD) into the optimization problem. The numerical implementation of MSR leads to highly sparse matrices; consequently, the prallel

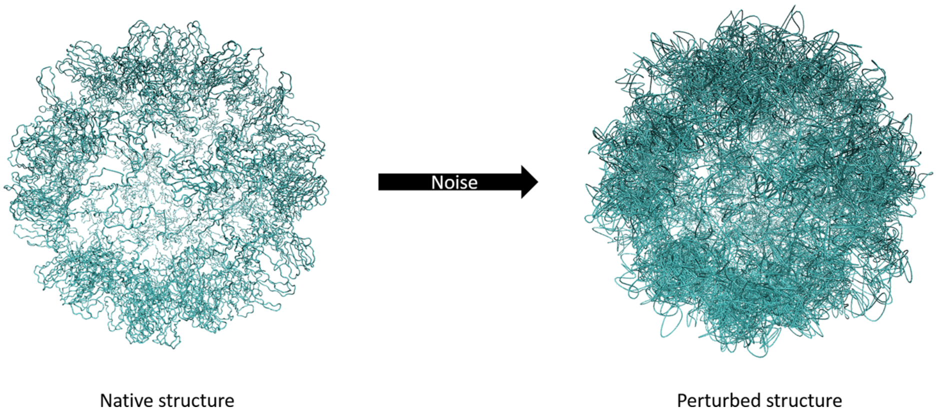


Figure 11. Snapshot of the cowpea chlorotic mottle virus capsid in its native (left) and perturbed (right) forms.

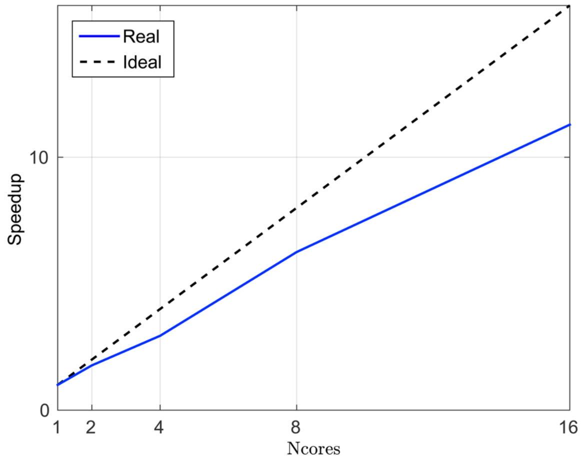


Figure 12. Strong scalability of MSR is shown for the human papilloma virus capsid using a total of 16 cores. The measured speedup is with respect to one core (serial run) based on the CPU time.

implementation shows good scaling with the size of the simulated systems. This suggests that the algorithm is suitable for supramillion-atom systems such as virus-like particles and other nanomaterials.

APPENDIX A: SPACE-WARPING METHOD

The space-warping method is a coarse-graining technique suitable for representing macromolecules in low-dimensional manifolds.[28,29] The method is based on writing the N-atom coordinates (denoted $\mathbf{r}$) in terms of a set of CG variables (denoted ϕ) via a Fourier-like expansion

$$\mathbf{r}_i = \mathbf{r}_c^0 + \sum_{\underline{k}} \mathbf{B}_{\underline{k}}(\mathbf{r}_i^0)\boldsymbol{\phi}_{\underline{k}} + \boldsymbol{\sigma}_i \tag{15}$$

where $\underline{k}$ is a triplet of indices (k_x, k_y, and k_z) that vary between 0 and N_{CG}, and the "Fourier modes" of order $\underline{k}$ are represented by $\boldsymbol{\phi}_{\underline{k}}$, a three-dimensional vector that serves as a CG variable that captures large-scale macromolecular conformational changes; $\mathbf{r}_i$ is the position of atom i; $\mathbf{B}$ is a matrix (of size $N \times N_{\mathrm{CG}}$) of the product of three Legendre functions of orders k_x, k_y, and k_z along the x, y, and z axes, respectively; this matrix depends on the positions of a reference all-atom configuration (denoted $\mathbf{r}^0$) of center of mass designated $\mathbf{r}_c^0$; $\boldsymbol{\sigma}_i$ is a three-dimensional vector of residual displacements that result from the difference between the positions generated by the coherent deformations. The Legendre polynomials are constructed over a normalized orthogonal box shown in Figure 13.

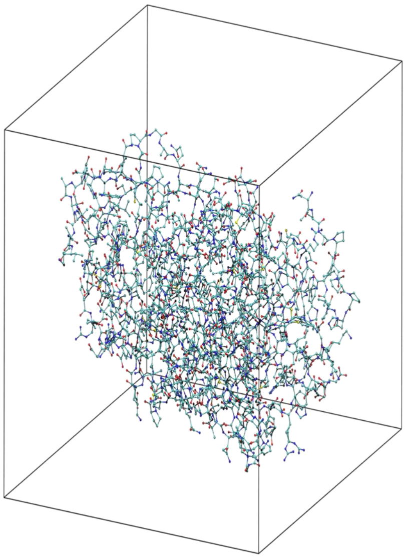


Figure 13. Space-warping method embeds a macromolecule in a normalized orthogonal box in which the basis functions (represented by $\mathbf{B}$) are constructed.

Coarse-graining is achieved by mass-weighted least-squares minimization, i.e., by minimizing $\sum_{i=1}^{N} m_i\boldsymbol{\sigma}_i^2$ with respect to $\boldsymbol{\phi}_{\underline{k}}$, with m_i being the mass of atom i. The result is a set of CG variables that serve as generalized centers of mass

$$\mathbf{B}^T\mathbf{M}\mathbf{B}\boldsymbol{\phi}_{\alpha} = \mathbf{B}^T\mathbf{M}\mathbf{r}_{\alpha} \tag{16}$$

where $\mathbf{M}$ is a diagonal matrix of the atomic masses, ϕ_α is a vector of CG coordinates of order $\underline{k}$, and $\mathbf{r}_\alpha$ is a vector of atomic positions (minus the center of mass of the macromolecule) in each α direction (with α representing x, y, or z). Thus, the coarse-graining matrix $\mathbf{Q}$ introduced in eq 1 is $(\mathbf{B}^T\mathbf{M}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{M}$.

The total order of the method is designated k_{m} such that $k_{\mathrm{m}} \geq k_x + k_y + k_z$. For example, if $k_{\mathrm{m}} = 0$, then $\{k_x, k_y, k_z\} = \{0,0,0\}$. The total number of CG variables in this case is 3 × 1, corresponding to ϕ_{000} being the center of mass of the macromolecule. As k_{m} increases, the CG variables capture additional information from the atomic scale, but they vary less slowly in time. Therefore, the space warping CG variables are classified into low-order and high-order variables. The former characterize the larger scale disturbances, whereas the latter capture short-scale ones.[18,29] For $k_m = 1$, $\{k_x, k_y, k_z\} = \{0,0,0\}$, $\{1,0,0\}$, $\{0,1,0\}$, $\{0,0,1\}$. In this case, the total number of CG variables is 3×4. It can be shown for $k_m = 1$, $\sum_{i=1}^{N} \mathbf{Q}_{k_xk_yk_z}(\mathbf{r}_i^0)$ is equal to 1 if $k_x = k_y = k_z = 0$, and it is equal to 0 otherwise. Thus, if the atomistic system has translated a distance d in all three directions, then this translation is captured by ϕ_{000} since $\Delta\phi = \mathbf{Q}\Delta\mathbf{r} = d\mathbf{Q}\mathbf{1}$, the first component of which is d, whereas the rest are 0. It can be further shown that ϕ_{100}, ϕ_{010}, and ϕ_{001} capture rotational motion,[28] whereas second-order CG variables $(k_x + k_y + k_z = 2)$ capture nonlinear transformations such as bending that macromolecular systems undergo.

APPENDIX B: REGULARIZATION OF INVERSE PROBLEMS

Inverse problems are often ill-posed.[21] In the context of reverse coarse-graining, solving inverse problems can be numerically challenging because the solution might exhibit numerical instabilities. For instance, if least-squares minimization is employed for recovering an all-atom state from the CG description using eq 1, then the reverse map $\mathbf{Q}^T\mathbf{Q}$ amplifies the high-frequency noise, which leads to numerical instabilities. In practice, regularization is therefore performed by incorporating additional constraints into the optimization problem. The most commonly used regularization is the L_2 norm of the solution vector (or in the present context, the vector of atomic positions $\mathbf{r}_\alpha$). In MSR, regularization is cast in terms of (1) the difference between the atomic positions, $\mathbf{r}_\alpha$, and the reference atomic positions, $\mathbf{r}_\alpha^0$, and (2) the Lagrange multipliers that enforce specific constraints on the FG constraints in eq 6.

AUTHOR INFORMATION

Corresponding Author

*E-mail: ortoleva@indiana.edu.

Funding

This work was supported by in part by the NSF division of material science research (grant 1533988, 80%) and the Indiana Clinical and Translational Sciences Institute, funded in part by grant UL1 TR001108 from the National Institutes of Health, National Center for Advancing Translational Sciences, Clinical and Translational Sciences Award (20%). The authors also acknowledge support by the IU College of Arts and Sciences via the Center for Theoretical and Computational Nanoscience, by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute, and in part by the Indiana METACyt Initiative. The Indiana METACyt Initiative at IU is also supported in part by Lilly Endowment, Inc. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Notes

The authors declare no competing financial interest.

REFERENCES

(1) Ortoleva, P. J. *J. Phys. Chem. B* **2005**, *109*, 21258−21266.

(2) Pankavich, S.; Shreif, Z.; Ortoleva, P. J. *Phys. A* **2008**, *387*, 4053−4069.

(3) Pankavich, S.; Shreif, Z.; Miao, Y.; Ortoleva, P. J. *J. Chem. Phys.* **2009**, *130*, 194115−194124.

(4) Cheluvaraja, S.; Ortoleva, P. J. *J. Chem. Phys.* **2010**, *132*, 075102.

(5) Abi Mansour, A.; Ortoleva, P. J. *J. Chem. Theory Comput.* **2016**, *12*, 1965−1971.

(6) Theodoropoulos, C.; Qian, Y.-H.; Kevrekidis, I. G. *Proc. Natl. Acad. Sci. U. S. A.* **2000**, *97*, 9840−9843.

(7) Gear, C. W.; Kevrekidis, I. G.; Theodoropoulos, C. *Comput. Chem. Eng.* **2002**, *26*, 941−963.

(8) Gear, C. W.; Hyman, J. M.; Kevrekidid, P. G.; Kevrekidis, I. G.; Runborg, O.; Theodoropoulos, C. *Commun. Math. Sci.* **2003**, *1*, 715−762.

(9) Kevrekidis, I.; Samaey, G. *Annu. Rev. Phys. Chem.* **2009**, *60*, 321−344.

(10) Bahar, I.; Atilgan, R. A.; Erman, B. *Folding Des.* **1997**, *2*, 173−181.

(11) Noid, W. G.; Chu, J.-W.; Ayton, G. S.; Krishna, V.; Izvekov, S.; Voth, G. A.; Das, A.; Andersen, H. C. *J. Chem. Phys.* **2008**, *128*, 244114−244124.

(12) Reith, D.; Putz, M.; Muller-Plathe, F. *J. Comput. Chem.* **2003**, *24*, 1624−1636.

(13) Shih, A. Y.; Arkhipov, A.; Freddolino, P. L.; Schulten, K. *J. Phys. Chem. B* **2006**, *110*, 3674−3684.

(14) Muller-Plathe, F. *ChemPhysChem* **2002**, *3*, 754−769.

(15) Rudd, R. E.; Broughton, J. Q. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1998**, *58*, 5893−5896.

(16) Abi Mansour, A.; Ortoleva, P. *J. Chem. Theory Comput.* **2014**, *10*, 518−523.

(17) Sereda, Y. V.; Espinosa-Duran, J. M.; Ortoleva, P. J. *J. Chem. Phys.* **2014**, *140*, 074102.

(18) Singharoy, A.; Cheluvaraja, S.; Ortoleva, P. J. *J. Chem. Phys.* **2011**, *134*, 044104.

(19) Singharoy, A.; Sereda, Y.; Ortoleva, P. J. *J. Chem. Theory Comput.* **2012**, *8*, 1379−1392.

(20) Gear, C. W.; Kevrekidis, I. G. *J. Sci. Compute.* **2003**, *24*, 1091−1106.

(21) Hansen, P. C. Problems with Ill-Determined Rank. In *Rank-deficient and Discrete Ill-posed Problems: Numerical Aspects of Linear Inversion*; SIAM: Philadelphia, PA, 1998; Vol. *4*, pp 69−98.

(22) Ensing, B.; Nielsen, S. O. Multiscale Molecular Dynamics and the Reverse Mapping Problem. In *Trends in Computational Nanomechanics: Transcending Length and Time Scales*; Dumitrica, T., Ed.; Springer Netherlands: Dordrecht, 2010; pp 25−59.

(23) Somogyi, E.; Mansour, A. A.; Ortoleva, P. J. *Comput. Phys. Commun.* **2016**, *202*, 337−350.

(24) Balay, S.; Gropp, W. D.; McInnes, L. C.; Smith, B. F. Efficient Management of Parallelism in Object Oriented Numerical Software Libraries. In *Modern Software Tools in Scientific Computing*; Birkhäuser Boston Inc.: Cambridge, MA, 1997; pp 163−202.

(25) Balay, S.; Brown, J.; Buschelman, K.; Gropp, W. D.; Kaushik, D.; Knepley, M. G.; McInnes, L. C.; Smith, B. F.; Zhang, H. *PETSc*. http://www.mcs.anl.gov/petsc (accessed August 21, 2016).

(26) Beazley, D. M. SWIG: An Easy to Use Tool for Integrating Scripting Languages with C and C++. In *Proceedings of the 4th Conference on USENIX Tcl/Tk Workshop*; Monterey, CA, July 10−13, 1996.

(27) Abi Mansour, A. *Github*. https://github.com/CTCNano/MSR (accessed August 21, 2016).

(28) Jaqaman, K.; Ortoleva, P. J. *J. Comput. Chem.* **2002**, *23*, 484−491.

(29) Singharoy, A.; Joshi, H.; Miao, Y.; Ortoleva, P. J. *J. Phys. Chem. B* **2012**, *116*, 8423−8434.

(30) Eldén, L. *BIT Numer. Math.* **1977**, *17*, 134−145.

(31) Barth, E.; Kuczera, K.; Leimkuhler, B.; Skeel, R. D. *J. Comput. Chem.* **1995**, *16*, 1192−1209.

(32) Davis, T. A. Basic Algorithms. In *Direct Methods for Sparse Linear Systems*; SIAM: Philadelphia, PA, 2006; pp 7−26.

(33) Norris, G. E.; Anderson, B. F.; Baker, E. N. *Acta Crystallogr., Sect. B: Struct. Sci.* **1991**, *47*, 998−1004.

(34) Gropp, W.; Lusk, E.; Doss, N.; Skjellum, A. *Parallel Comput.* **1996**, *22*, 789−828.

(35) Stein, P. E.; Boodhoo, A.; Armstrong, G. D.; Cockle, S. A.; Klein, M. H.; Read, R. J. *Structure* **1994**, *2*, 45−57.

(36) MacKerell, A. D., Jr.; Bashford, D.; Bellott, M.; Dunbrack, R. L., Jr.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; et al. *J. Phys. Chem. B* **1998**, *102*, 3586−3616.

(37) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *J. Chem. Phys.* **1983**, *79*, 926−935.

(38) Speir, J. A.; Munshi, S.; Wang, G.; Baker, T. S.; Johnson, J. E. *Structure* **1995**, *3*, 63−78.