OpenMSCG: A Software Tool for Bottom-up Coarse-graining

Yuxing Peng,[1] Alexander J. Pak,[2] Aleksander E. P. Durumeric,[3] Patrick G. Sahrmann,[4] Sriramvignesh Mani,[4] Jaehyeok Jin,[4] Timothy D. Loose,[4] Jeriann R. Beiter,[4] and Gregory A. Voth[4, a)]

[1] NVIDIA Corporation, 2788 San Tomas Expressway, Santa Clara, CA 95051, USA

[2]Department of Chemical and Biological Engineering, Colorado School of Mines, Golden, Colorado 80401, USA

[3]Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany

[4]Department of Chemistry, Chicago Center for Theoretical Chemistry, James Franck Institute, and Institute for Biophysical Dynamics, The University of Chicago, Chicago, Illinois 60637, USA

[a)] Author to whom correspondence should be addressed: gavoth@uchicago.edu

ABSTRACT

The “bottom-up” approach to coarse-graining – for building accurate and efficient computational models to simulate large-scale and complex phenomena and processes – is an important approach in computational chemistry, biophysics, and materials science. As one example, the multiscale coarse-graining (MS-CG) approach to developing CG models can be rigorously derived using statistical mechanics applied to fine-grained, i.e., all-atom simulation data for a given system. Under a number of circumstances, a systematic procedure such as MS-CG modeling is particularly valuable. Here we present the development of the OpenMSCG software, a modularized open-source software that provides a collection of successful and widely applied bottom-up CG methods, including Boltzmann Inversion (BI), Force-Matching (FM), Ultra-Coarse-Graining (UCG), Relative Entropy Minimization (REM), Essential Dynamics Coarse-Graining (ED-CG), and Heterogeneous Elastic Network Modeling (HeteroENM). OpenMSCG is a high-performance and comprehensive toolset that can be used to derive CG models from large-scale fine-grained simulation data in file formats from common molecular dynamics (MD) software packages, such as GROMACS, LAMMPS and NAMD. OpenMSCG is modulized in the Python programming framework, which allows users to create and customize modeling “recipes” for reproducible results, thus greatly improving the reliability, reproducibility, and sharing of bottom-up CG models and their applications.

1 Introduction

Coarse-grained (CG) modeling and simulation has become one of the most important methods for *in silico* studies of complex phenomena across biology, chemistry, physics and materials science.[1-4] CG models, in which each particle represents a grouping of atoms at a lower resolution and with reduced interactions and detail, can greatly extend the achievable length and time scales of molecular dynamics (MD) simulations compared to atomistic, or fine-grained (FG), models. This allows for insighs into collective behavior and can improve the sampling of the processes of interest in the simulated system . Despite decades of significant theoretical and methodological efforts, systematically developing accurate and efficient CG models remains a challenge, especially from the "bottom-up", i.e., by an application[2, 4] of the principles of statistical mechanics rather than through an alternative "top-down" fitting of the CG model from certain properties.[3, 5, 6]

After defining the CG "sites" for a given system from the fine-grained model (also referred to as "mapping"), the central task in CG modeling is to develop the effective interaction force-field, i.e., a set of energy functions and corresponding parameters, for these CG sites. There are two fundamental strategies for the development of CG force-fields. The strategy we focus on in this paper is the bottom-up approach, in which the models and parameters are derived from microscopic interaction data (e.g., collected from atomistic simulations) using statistical mechanics.[1, 2, 4] One of the most successful methods using this bottom-up CG'ing strategy is the Multiscale Coarse-graining (MS-CG) method, which was originally proposed by Izvekov and Voth[7, 8] and was then further justified and generalized.[9-11] In MS-CG modeling, reference forces on CG sites are mapped from atomistic forces collected from all-atom (AA) simulations, and the ensemble average of the least-squares difference between the CG forces and reference forces is variationally minimized, a process sometimes referred to as “force-matching” (FM). (We note that this widely used and somewhat unfortunate terminology perhaps does not give the original MS-CG approach the credit it deserves, because the original MS-CG work is more of a force “renormalization” from the FG resolution to the CG one; for papers in which FM is used solely at the FG, all-atom level to define an approximate force field from *ab initio* or other higher level data, see refs[12, 13].) In fact, the MS-CG approach can be viewed as an early form of machine learning (ML) in which interactions at the CG level are derived from a fitting of FG interactions to numerically renormalize and thereby express them. Extended functionalities have also been developed for the MS-CG framework, such as three-body non-bonded interactions,[14, 15] CG virtual site and center-of-charge and mappings,[16, 17] and the (rather non-trivial) extension of the MS-CG theory to quantum Boltzmann statistical mechanics.[18]

Despite its successes, the conventional MS-CG approach, which usually uses a pairwise numerically derived potential to represent the interactions between pairs of CG sites, may not be flexible enough to accurately model a variety of chemical and physical characteristics of the given system, resulting in a loss of accuracy and limited transferability.[4] However, the concept of Ultra-Coarse-graining (UCG) theory[19-21] has been developed to address this issue (and others) by introducing internal “states” to CG sites to incorporate the effects of important degrees of freedom that may be missing at the CG resolution. In the UCG framework, the internal states interact through an associated CG site state-dependent potential that can also be obtained by using the MS-CG procedure (possibly with other steps or fitting as well). To efficiently incorporate the internal states into UCG models, one of the most successful approaches [21]assumes a separation of time scales is assumed such that the internal state relaxation time is shorter than the CG site translational relaxation times. In the rapid local equilibrium (RLE) limit, the internal states can be regarded as being in quasi-equilibrium, and the UCG site is expressed as the linear combination of these internal states with their substate probabilities. Hence, the UCG internal sites can dynamically switch their internal states and corresponding force-field parameters, on-the-fly, depending on their surrounding environment and local structural conditions. Recent work has shown that UCG theory significantly improves, e.g., the accuracy of structural properties for interfacial systems[22] and energetics for hydrogen bonding,[23] while also enhancing the transferability of bottom-up CG models.[24].

Another limiting case for UCG modeling is the case in which the internal UCG states make rare transitions relative to the timescale of the CG site motions. [20] The resulting UCG dynamics is somewhat like a trajectory for the CG sites interacting with a moving Kinetic Monte Carlo algorithm for the internal UCG state transitions. This approach has facilitated the simulation of biomolecular active matter, e.g., actin filaments that hydrolyze ATP during their motions.[25, 26]

In between the two limiting UCG cases described in the last two paragraphs, one can develop a more *ad hoc* equation of motion for the UCG MD dynamics coupled to internal CG site state changes. However, while being more ad hoc such an algorithm has led to some noteworthy successes, e.g., in the simulation of the assembly of the HIV-1 virus capsid from over 1000 proteins.[27, 28]

Another systematic “bottom-up” CG’ing approach is relative entropy minimization (REM), which quantitatively minimizes the log difference between the configurational phase-space probability distributions of the CG and reference FG model (e.g., all-atom), which is referred to as the relative entropy. On the basis of the mathematical relationship between the relative entropy and linear model parameters, an iterative approach has been developed to variationally minimize the relative entropies by refining the CG potentials stepwise.[29-32]

In this paper and in the OpenMSCG software, the term “MS-CG” is taken to refer to not only to the conventional “FM” approach as described earlier, but also to the general set of “bottom-up” CG’ing ideas in multiscale theory. These include the three major approaches described above as well as other methods such as Boltzmann inversion (BI)[33] and heterogeneous elastic network modeling (HENM).[34]

It is important to note that “bottom-up” CG models are derived from FG interaction data at a given thermodynamic state point and are therefore not rigorously transferable from one system to another or one state point to another.[2, 4, 24, 35-38] (Indeed, there is no sound theoretical basis from statistical mechanics in which any top-down CG model can claim transferability between systems either.) Instead, the vision of our multiscale coarse-graining methodology is to provide a systematic modeling workflow that can be easily and effectively applied to a variety of studied systems, each individually. Moreover, a successful MS-CG model usually needs a combination of multiple methods, and requires adequate knowledge, experience, and domain insight from the researcher. Therefore, a sustainable and sharable development environment is of great interest and value for the CG’ing community.

To address these goals, we report here the development and release of an open-source software package, OpenMSCG, for the purpose of high performance, sharable and reproducible MS-CG modeling. The release of OpenMSCG includes modeling tools for the BI, FM, UCG, and REM methods, and it supports a variety of data formats used as input/output for MD software packages such as Gromacs[39], NAMD,[40] and LAMMPS[41]. OpenMSCG is wrapped as a Python3 package and can be used as a software development kit (SDK) for researchers to customize their workflow by importing these modules into their own scripts as building blocks. Therefore, researchers can apply OpenMSCG as standardized templates to create and publish their “recipes,” detailing all steps from the beginning to the end, of any modeling work, which can enable reliable and collaborative sharing throughout the CG'ing community.

The remaining body of this article is organized as follows: Section 2 briefly describes the architecture, workflow, and features of OpenMSCG, as well as the implemented MS-CG methods. Several benchmark studies, including a liquid-vapor interface, heterogeneous fluid mixtures, and HIV-1 CA/SP1 protein-protein interactions, are reported and discussed using different CG modeling approaches in Section 3, as well as a comparison of OpenMSCG with related software packages. Finally, Section 4 provides conclusions and perspectives for future development.

2 Methods and Implementations

2.1 Overview

In general, a bottom-up CG model is derived to generate the configurational equilibrium probability distribution of the all-atom model when mapped to the CG particle phase space:

$$\mathrm{P}_{CG}(\boldsymbol{R},\boldsymbol{P}) = \iint \mathrm{P}_{AA}(\boldsymbol{r},\boldsymbol{p})\delta[\boldsymbol{M}(\boldsymbol{r}) - \boldsymbol{R}]\delta[\boldsymbol{M}(\boldsymbol{p}) - \boldsymbol{P}]\mathrm{d}\boldsymbol{r}\mathrm{d}\boldsymbol{p} \qquad (1)$$

where $\boldsymbol{M}$ is the mapping function, by which the CG phase space configurations $(\boldsymbol{R},\boldsymbol{P})$ are constructed from the atomistic configurations $(\boldsymbol{r},\boldsymbol{p})$. A common mapping function is the center-of-mass of atoms that are grouped by certain strategies, such as geometry-based or sequence-based approaches. In this case, the mapping function can be considered as an $n \times N$ matrix, $\widehat{\boldsymbol{M}}$, for the linear transformation of $3n$ atomistic coordinates into $3N$ CG coordinates.

In the canonical ensemble, Eq. 1 can be rewritten as:

$$\frac{1}{N!\,h^{3N}} Z_{CG}^{-1} e^{-\beta \mathrm{U}_{CG}(\boldsymbol{R})} e^{-\beta \boldsymbol{P}^2/2\boldsymbol{M}} = \frac{1}{n!\,h^{3n}} Z_{AA}^{-1} \int e^{-\beta \mathrm{U}_{AA}(\boldsymbol{r})} \delta[\boldsymbol{M}(\boldsymbol{r}) - \boldsymbol{R}] d\boldsymbol{r} \times \int e^{-\beta \boldsymbol{p}^2/2\boldsymbol{m}} \delta[\boldsymbol{M}(\boldsymbol{p}) - \boldsymbol{P}] \mathrm{d}\boldsymbol{p} \tag{2}$$

where

$$Z_{CG} = \frac{1}{N!\,h^{3N}} \int e^{-\beta \mathrm{U}_{CG}(\boldsymbol{R})} d\boldsymbol{R} \int e^{-\beta \boldsymbol{P}^2/2\boldsymbol{M}} \mathrm{d}\boldsymbol{P} \tag{3}$$

and

$$Z_{AA} = \frac{1}{n!\,h^{3n}} \int e^{-\beta \mathrm{U}_{AA}(\boldsymbol{r})} d\boldsymbol{r} \int e^{-\beta \boldsymbol{p}^2/2\boldsymbol{m}} \mathrm{d}\boldsymbol{p} \tag{4}$$

are the partition functions of the CG model and the all-atom model, respectively. Once the mapping function is decided, the right side of Eq. 2 can be obtained from all-atom MD simulations. Then, the MS-CG method seeks to obtain the CG particle effective potential, $\mathrm{U}_{CG}(\boldsymbol{R})$, which reproduces the configurational probability distributions of the mapped all-atom reference data. This “potential” is actually a potential of mean force, i.e., a free energy surface for the CG particles (aka the CG “sites” or “beads”).

Previous studies have focused on constructing $\mathrm{U}_{CG}(\boldsymbol{R})$ using multi-body interactions. The most common design of $\mathrm{U}_{CG}(\boldsymbol{R})$, which is implemented in OpenMSCG, holds a form similar to that of all-atom molecular mechanics:[42, 43]

$$\mathrm{U}_{CG}(\boldsymbol{R}) = \sum \mathrm{U}_{pair}(\boldsymbol{r}) + \sum \mathrm{U}_{2b}(\boldsymbol{b}) + \sum \mathrm{U}_{3b}(\boldsymbol{\theta}) + \sum \mathrm{U}_{4b}(\boldsymbol{\varphi}) \tag{5}$$

where $\boldsymbol{r}, \boldsymbol{b}, \boldsymbol{\theta},$ and $\boldsymbol{\varphi}$ are geometric terms for non-bonded pairwise distances, 2-body bond lengths, 3-body bending angles and 4-body torsional dihedral angles, respectively. In some cases, pair-wise interactions only for non-bonded CG-sites fail to capture certain structural properties, such as the tetrahedral structure of liquid water due to hydrogen bonding. Therefore, we also introduced an optional Stillinger–Weber (SW) style three-body non-bonded term,[14, 44] which can be used to improve such structural correlations in CG systems, for example, the recently reported BUMPer water model.[45, 46] Depending upon the particular functional form for $\mathrm{U}_{CG}(\boldsymbol{R})$ that is chosen, a set of parameters, $\{\boldsymbol{\lambda}\}$, must be fitted or optimized to satisfy Eq. 2. In MS-CG methodology, a practical functional form for these parameters is the $\boldsymbol{k}$-order B-spline,[47, 48] given by

$$\boldsymbol{S}_k(x) = \sum_{i=1}^{n} c_i \boldsymbol{B}_{i,k}(x) \tag{6}$$

where $\boldsymbol{B}_{i,k}(x)$ is a B-spline basis function with coefficient c_i, which defines the controlling knots of the spline, are to be fitted or optimized during the parameterization.

The framework for MS-CG modeling described above has been implemented in the OpenMSCG software framework following the hierarchy that is illustrated in Figure 1 and summarized below.

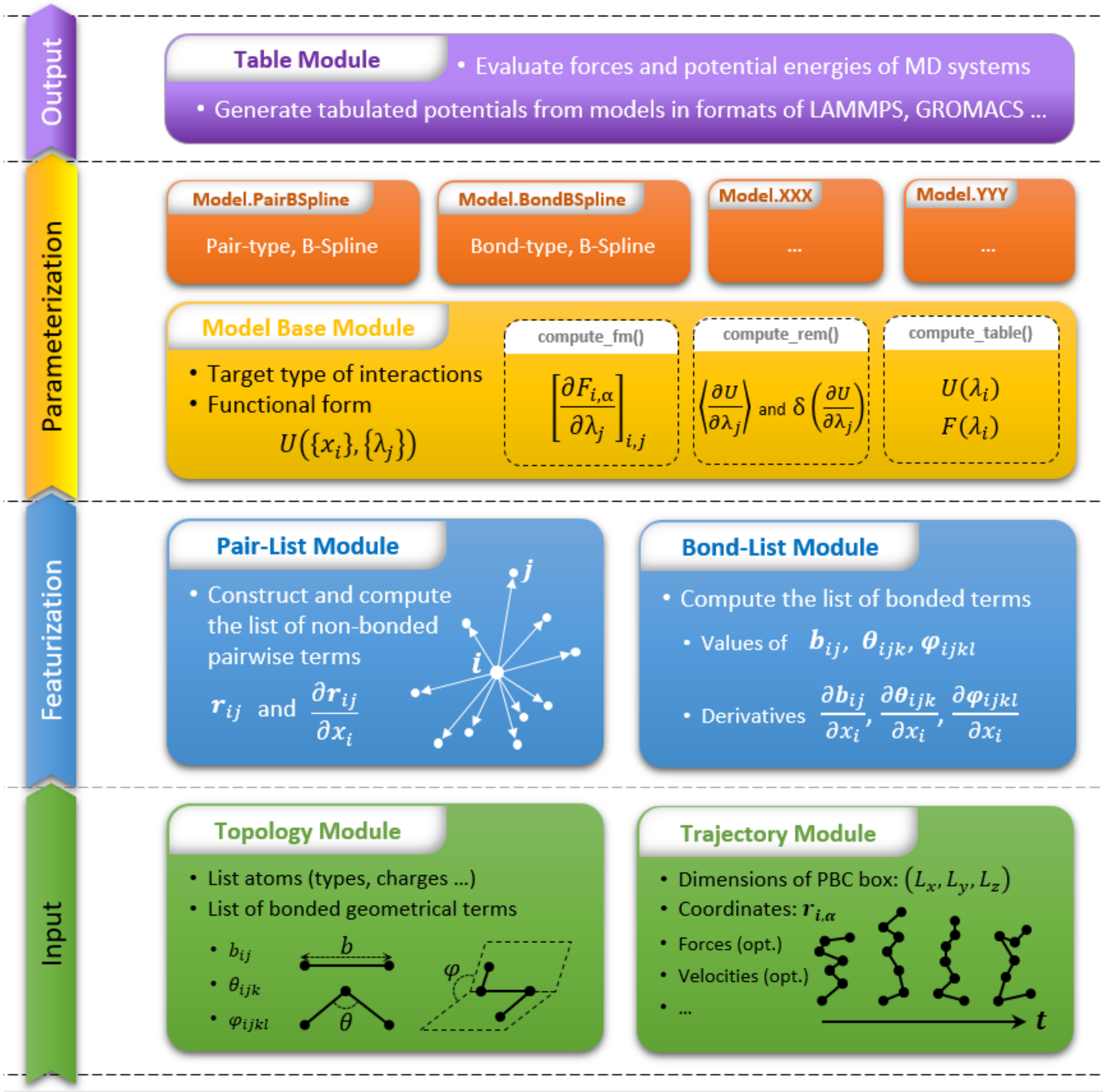

Output
Table Module
• Evaluate forces and potential energies of MD systems
• Generate tabulated potentials from models in formats of LAMMPS, GROMACS ...
Parameterization
Model.PairBSpline
Pair-type, B-Spline
Model.BondBSpline
Bond-type, B-Spline
Model.XXX
...
Model.YYY
...
Model Base Module
• Target type of interactions
• Functional form
$U(\{x_i\}, \{\lambda_j\})$
compute_fm()
$\left[\frac{\partial F_{i,\alpha}}{\partial \lambda_j}\right]_{i,j}$
compute_rem()
$\left\langle\frac{\partial U}{\partial \lambda_j}\right\rangle$ and $\delta\left(\frac{\partial U}{\partial \lambda_j}\right)$
compute_table()
$U(\lambda_i)$
$F(\lambda_i)$
Featurization
Pair-List Module
• Construct and compute the list of non-bonded pairwise terms
r_{ij} and $\frac{\partial r_{ij}}{\partial x_i}$
j
i
Bond-List Module
• Compute the list of bonded terms
• Values of b_{ij}, θ_{ijk}, φ_{ijkl}
• Derivatives $\frac{\partial b_{ij}}{\partial x_i}$, $\frac{\partial \theta_{ijk}}{\partial x_i}$, $\frac{\partial \varphi_{ijkl}}{\partial x_i}$
Input
Topology Module
• List atoms (types, charges ...)
• List of bonded geometrical terms
• b_{ij}
• θ_{ijk}
• φ_{ijkl}
b
θ
φ
Trajectory Module
• Dimensions of PBC box: (L_x, L_y, L_z)
• Coordinates: $r_{i,\alpha}$
• Forces (opt.)
• Velocities (opt.)
• ...
t

Figure 1. Overview of the OpenMSCG software illustrated as a four-layer framework: (1) Input, (2) Featurization, (3) Parameterization and (4) Output layers (from bottom to top).

(1) **Input Layer**. Modules in the input layer are used to read topology definitions, trajectories, as well as other input files. The topology reader module supports multiple file formats that include topology information, such as particle types and bonds. The trajectory reader module reads and processes simulation trajectories in file formats that are generated by common MD simulation packages, including GROMACS,[39] NAMD,[40] and LAMMPS.[41] These modules provide the data

feed into the featurization layer, which is described next. These modules can also be used to convert all-atom MD simulation trajectories into CG trajectories according to user-defined mapping rules; for this purpose, we have developed the CGMAP tool. Additionally, the resulting CG trajectories can be used for training of machine-learned CG force-fields (e.g., based on deep neural networks) with minimal user involvement.[49-51] While CGMAP defines mappings for both atomisitc configurational and force data, it is worth noting that the force mapping operator is not uniquely determined by the coordinate mapping operator.[10]

The aggforce software feature furthermore enables optimization of the force mapping operator to reduce the noise present in the atomistic forces, with a particular emphasis towards machine-learned CG force-fields.[52] We have developed the **CGFMP** tool to enable optimization of the force mapping, given a coordinate map, through interfacing with the aggforce software package.

(2) **Featurization Layer**. The CG configurations, $\boldsymbol{R}$, from the mapped CG trajectories are read and fed into this layer to calculate the geometrical terms, which are defined by the topology of the CG system. The generated geometric terms are fed into the parameterization layer described below. These modules apply standard MD algorithms for computational efficiency, such as the Verlet-List algorithm[53] for the construction of the pairwise neighbor lists. Forces are evaluated from defined potential energy functions by the chain rule:

$$f_I^\alpha = -\frac{\partial U_{CG}(\boldsymbol{R})}{\partial R_I^\alpha} = -\sum_i \frac{\partial U_i(\boldsymbol{\chi_i})}{\partial \boldsymbol{\chi_i}} \frac{\partial \boldsymbol{\chi_i}}{\partial R_I^\alpha} \tag{7}$$

where α is the x, y or z-component, and $\boldsymbol{\chi_i}$ and $U_i(\boldsymbol{\chi_i})$ are the geometric variable and decomposed potential function for the CG site I, respectively. The partial derivatives, $\partial \boldsymbol{\chi_i} / \partial R_I^\alpha$, are calculated and stored by the modules in this layer.

(3) **Parameterization Layer**. The parameterization layer provides several modeling modules, each of which can be declared to parameterize a given interaction type using a particular functional form, such as B-splines for nonbonded pair or Harmonic for bonded interactions. These modules are designed to be uniform with an Application Programming Interface (API) that facilitates their use across parameterization tools. These APIs export three groups of quantities relevant to parameter optimization or fitting:

a. $\boldsymbol{U}(\lambda_i)$ and $\boldsymbol{F}(\lambda_i)$ – values of potential energies and forces, respectively, on CG sites, which can be used for Monte Carlo (MC)/MD codes or energy/force analysis.

b. $\left[\frac{\boldsymbol{\partial F_{i,\alpha}}}{\boldsymbol{\partial \lambda_j}}\right]_{\boldsymbol{i,j}}$ – matrix of partial derivatives of forces on the ***i***th CG site with respect to the ***j***th parameter in the model, which is used in the FM method.

c. $\left\{\frac{\partial U}{\partial \lambda_j}\right\}$ – vector of partial derivatives of potential energies for each parameter in this model, which is used in the REM method.

(4) **Output Layer**. Tabulated numerical potentials are widely used in bottom-up CG simulations, due to their flexibility and low computational cost. OpenMSCG applies this idea and can output CG potentials into tabulated file formats that are used in common MD simulators. This does not mean that the models provided by OpenMSCG only support tabulated functional forms, as conventional analytical functions can be converted into tabulated forms. Similarly, tabulated potentials can be fitted to an empirical functional form using regression software tools, e.g., *SciPy*.[54]

2.2 Parameterization Tools

There are currently three MS-CG parameterization methods deployed in OpenMSCG, each of which has been applied successfully in prior MS-CG studies. The detailed theories and applications can be found in related literature, so in this section only a brief overview and implementations of them in OpenMSCG are summarized below:

(1) Direct **Boltzmann Inversion**. The simplest and most straightforward approach for CG modeling is to use the projected free energy surfaces (FES) as effective CG potentials. The FES can be derived from inverting the probability distribution functions of targeted feature variables, which are sampled from the AA trajectories,

$$U_{CG}(\boldsymbol{\chi}) = -k_B T \cdot lnP'(\boldsymbol{\chi}) + C \tag{8}$$

where $P'(\boldsymbol{\chi})$ is the conditional probability with degeneracy corrections. For example, the CG potential for a pairwise interaction is usually derived from its radial distribution function, $g(r) = \langle P(r) \rangle / (\rho \cdot 4\pi r^2)$, known as the Boltzmann Inversion (BI) method. The approach is implemented as the **CGIB** tool in OpenMSCG. Because the direct BI approach ignores the correlations between different types of interactions, its CG potentials may often be unsatisfactory.[55] However, these potentials are still useful as initial trial CG potentials for the Iterative Boltzmann Inversion (IBI) method[55] or other iterative methods, such as REM.

(2) **Force-Matching** and **Ultra-Coarse-Graining** Methods. Instead of directly matching the structural correlations, as done in BI and IBI, the FM approach minimizes the differences between forces from CG models and reference forces from all-atom trajectories. In practice, tabulated effective forces are derived by the variational principle by minimizing force residuals:

$$\chi^2[\boldsymbol{F}] = \frac{1}{3N} \langle \sum_{\boldsymbol{I}=1}^{N} \left| \boldsymbol{F}_{CG,I}[\boldsymbol{M}_{\boldsymbol{R}}^{N}(\boldsymbol{r}^n)] - \boldsymbol{F}_{AA,I}(\boldsymbol{r}^n) \right|^2 \rangle \tag{9}$$

where *N* and *n* are the number of CG sites and all-atom sites (atoms), respectively. Because the CG forces are calculated from a group of linear functions of the model parameters, minimization of χ^2 is equivalent to obtaining the solution ϕ for the least-squares regression problem:[48]

$$\boldsymbol{F}\phi = \boldsymbol{f} \tag{10}$$

where $\boldsymbol{F}$ is the coefficient matrix of force derivatives $\left[\frac{\partial f_{i,t}}{\lambda_j}\right]_{3n_A \times n_t,\ n_\lambda}$, n_A is the number of atoms, n_t is the number of trajectory frames and n_λ is the total number of parameters in the model. The coefficient matrix is highly sparse, and the computer memory requirement is linear to the size of the trajectory, which is not practical.[48] Therefore, many improved algorithms have been developed to resolve this issue, such as the Block-Average (BA) and Normal Equation (NE)[48] methods. The NE algorithm, which is the most commonly used, is implemented in OpenMSCG. The solution ϕ is obtained by solving an equivalent singular value decomposition problem as:

$$\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F}\phi = \boldsymbol{F}^{\mathrm{T}}\boldsymbol{f} \tag{11}$$

in which a square matrix $\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F}$ with only $n_\lambda \times n_\lambda$ dimensions needs to be stored and accumulated over trajectory frames. To address potential overfitting, ridge regression[56] and Bayesian[57] regularization approaches are also applied within this framework.

The UCG methodology, which has been extensively developed in recent years,[19-24] can be implemented by extending the conventional FM framework.[15] After reading a trajectory frame and constructing the features, a weighting function determines the probabilities of internal states for every CG particle on-the-fly. To efficiently introduce the internal state probabilities into the FM framework, we use an indirect sampling approach that does not explicitly involve the probability information during FM. In detail, a number of independent frames of the CG system are spawned, which all have the same coordinates but different CG types that represent different internal states and are assigned randomly in agreement with the determined weights, known as the dynamic type. If one chooses a sufficiently large number for sampling, the dynamic type FM force residual becomes the ideal UCG force residual. The construction of the coefficient matrix for spawned frames, as well as other modeling steps, are performed in the same manner as conventional FM.

Both the conventional FM and the newly developed UCG methodologies are implemented in the **CGFM** tool, and their detailed workflow is illustrated in Figure 2(a).

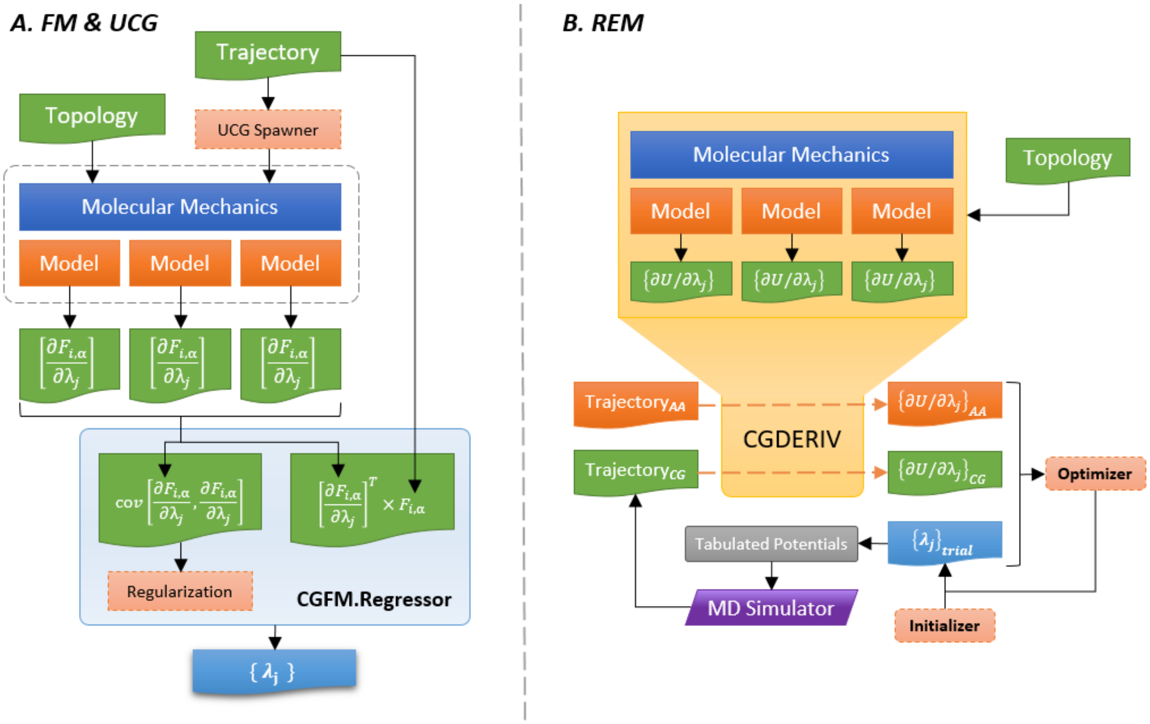

A. FM & UCG
Trajectory
Topology
UCG Spawner
Molecular Mechanics
Model
Model
Model
$\left[\frac{\partial F_{i,\alpha}}{\partial \lambda_j}\right]$
$\mathrm{cov}\left[\frac{\partial F_{i,\alpha}}{\partial \lambda_j}, \frac{\partial F_{i,\alpha}}{\partial \lambda_j}\right]$
$\left[\frac{\partial F_{i,\alpha}}{\partial \lambda_j}\right]^T \times F_{i,\alpha}$
Regularization
CGFM.Regressor
$\{\lambda_j\}$
B. REM
Molecular Mechanics
Topology
Model
Model
Model
$\{\partial U/\partial \lambda_j\}$
Trajectory_{AA}
Trajectory_{CG}
CGDERIV
$\{\partial U/\partial \lambda_j\}_{AA}$
$\{\partial U/\partial \lambda_j\}_{CG}$
Optimizer
Tabulated Potentials
$\{\lambda_j\}_{trial}$
MD Simulator
Initializer

Figure 2. Illustrations of workflows for the CGFM and CGREM tools: (A) The CGFM tool used for FM and UCG methods, in which the force-field coefficients are solved from linear regression. (B) The CGREM tool used for the REM method, in which the force-field coefficients are obtained by iteratively minimizing the relative entropy between all-atom and CG trajectories calculated by the CGDERIV tool.

(3) Iterative **Relative Entropy Minimization**. The REM method[29-31] aims to minimize the relative entropy objective function,

$$S_{rel} = \int p_{AA}(\boldsymbol{r})\, ln \frac{p_{AA}(\boldsymbol{r})}{p_{CG}(M[\boldsymbol{r}])} d\boldsymbol{r} + \langle S_{map} \rangle \tag{12}$$

where $p_{AA}(\boldsymbol{r})$ and $p_{CG}(M[\boldsymbol{r}])$ are the probabilities of the all-atom and CG configurations, e.g., as sampled from MD simulations, and S_{map} describes the degeneracy due to mapping. In the canonical ensemble, Eq. 12 can be rewritten as,

$$S_{rel} = \beta\langle U_{CG} - U_{AA}\rangle - \beta\langle A_{CG} - A_{AA}\rangle \tag{13}$$

where $A = -k_B T\, lnZ$ is the Helmholtz free energy. This method optimizes the model parameters $\{\lambda_i\}$ by minimizing the relative entropy, S_{rel}. In practice, an iterative approach can be used to tune the parameters with a certain step length χ, described as

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} - \chi\left[\left(\frac{\partial S_{rel}}{\partial \lambda_i}\right) \Big/ \left(\frac{\partial^2 S_{rel}}{\partial {\lambda_i}^2}\right)\right] \tag{14}$$

where the derivatives can be written as

$$\frac{\partial S_{rel}}{\partial \lambda_i} = \beta\left(\langle\frac{\partial U}{\partial \lambda_i}\rangle_{AA} - \langle\frac{\partial U}{\partial \lambda_i}\rangle_{CG}\right) \tag{15}$$

and

$$\frac{\partial^2 S_{rel}}{\partial {\lambda_i}^2} = \beta\left(\langle\frac{\partial^2 U}{\partial {\lambda_i}^2}\rangle_{AA} - \langle\frac{\partial^2 U}{\partial {\lambda_i}^2}\rangle_{CG}\right) + \beta^2\left(\langle\left(\frac{\partial U}{\partial \lambda_i}\right)^2\rangle_{CG} - \langle\frac{\partial U}{\partial \lambda_i}\rangle_{CG}^2\right) \tag{16}$$

If the parameters are linear coefficients in the potential, the second-order derivatives will be zero and stepwise tuning from step *k* to step *k+1* can be written as

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} - \frac{\chi}{\beta}\left[\left(\langle\frac{\partial U}{\partial \lambda_i}\rangle_{AA} - \langle\frac{\partial U}{\partial \lambda_i}\rangle_{CG}\right) \Big/ \left(\langle\left(\frac{\partial U}{\partial \lambda_i}\right)^2\rangle_{CG} - \langle\frac{\partial U}{\partial \lambda_i}\rangle_{CG}^2\right)\right] \tag{17}$$

In Eq. 17, the potential energy derivatives are evaluated over two CG trajectories. The term $\langle \ldots \rangle_{AA}$ denotes the ensemble average from the CG trajectory mapped from the all-atom reference simulations, while the term $\langle \dots \rangle_{CG}$ denotes the average from the trajectory from a CG simulation using a trial set of parameters. The initial parameters can be obtained from BI or FM. In OpenMSCG, this method is implemented in two tools, CGDERIV and CGREM, illustrated in Figure 2(b). The tool CGDERIV calculates the ensemble averages of the energy derivatives from any given trajectories, which serves as the kernel work for CGREM. CGREM carries out the iterative workflow described as below:

a. Read in the $\langle \frac{\partial U}{\partial \lambda_i} \rangle_{AA}$ calculated by CGDERIV on the reference trajectory mapped from the all-atom MD simulation.

b. Read in the initial trial parameters $\lambda_i^{(0)}$

c. Launch external MD software to conduct a CG simulation with the trial parameters $\lambda_i^{(k)}$ and generate a new CG trajectory.

d. Call CGDERIV to calculate $\langle \frac{\partial U}{\partial \lambda_i} \rangle_{CG}$ and $\langle \left(\frac{\partial U}{\partial \lambda_i}\right)^2 \rangle_{CG}$from the new CG trajectory.

e. Calculate the step length and adjust the trial parameters to $\lambda_i^{(k+1)}$

f. Repeat step (c)-(e) until the parameters converge.

Beyond the standard stepwise scheme, CGREM is flexible enough to adopt any customized optimize schemes for $\lambda_i^{(k)} \rightarrow \lambda_i^{(k+1)}$, which use the terms $\langle \frac{\partial U}{\partial \lambda_i} \rangle$, $\langle \left(\frac{\partial U}{\partial \lambda_i}\right)^2 \rangle$, and $\langle \frac{\partial^2 U}{\partial {\lambda_i}^2} \rangle$ calculated by the CGDERIV tool.

2.3 Tools for Modeling of Complex Biomolecules

One of the most important applications of CG modeling and simulations is for studies on biological macromolecules, such as proteins and nucleic acids.[58] In most CG models, such as MARTINI,[3, 59] one or more CG sites are used to represent each functional group (e.g., an amino acid residue). To further increase the spatiotemporal scales, models with less granularity (lower CG resolution), in which a single CG site represents multiple functional groups, are needed. To this end, two systematic and quantitative approaches, the Essential Dynamics Coarse-Graining (EDCG)[60] and HeteroENM[34] methods, have been successfully applied in various CG studies of complex biomolecular systems. Both methods aim to retain and reproduce the most significant atomistic fluctuations (also known as essential dynamics) from the CG models based on analysis from all-atom simulation trajectories. These two methods are implemented in OpenMSCG and are briefly introduced below.

(1) **Essential Dynamics Coarse-Graining**. When grouping atoms into a specified number of domains, each of which represented by a single CG site, the covariance of atomistic fluctuations between the atoms in the same domain will be lost. The target of EDCG is to optimally solve the grouping rules that can yield the least loss (residual) of covariance, which is defined as

$$\chi^2 = \frac{1}{3N}\sum_{\boldsymbol{I}=1}^{N}\frac{1}{n_t}\sum_{\boldsymbol{t}=1}^{n_t}\left(\sum_{\boldsymbol{i\in I}}\sum_{\boldsymbol{j>i\in I}}\left|\Delta r_i(t)-\Delta r_j(t)\right|^2\right) \tag{18}$$

where N is the number of CG sites (or domains) and n_t is the number of trajectory frames. The covariance matrix can be usually calculated from principal component analysis of the all-atom trajectories. The minimization of the residual is equivalent to retaining the most significant low frequency motions in the CG models. A practical approach for EDCG protein models is to define

CG sites from the center of mass of a group of C_α atoms, which are assumed to be contiguous in the amino acid sequence of the protein (i.e., a sequential model). Therefore, once N is decided, the minimization can be done by vatiationally adjusting the boundaries between the N domains in the protein primary sequence. Previous approaches combined simulated annealing with the steepest descent search for optimization, which are not guaranteed to find the global minimum. In OpenMSCG, EDCG is implemented in the tool CGED, in which a new algorithm, dynamical programming, is applied to ensure global optimization. The details of the implementation are described in the Supporting Information.

(2) **Heterogeneous Elastic Network Models**. After the CG sites are defined by the EDCG approach, the structure of the biomolecules can be maintained by an elastic network model (ENM),[61, 62] in which every pair of CG sites with an average separation distance within a certain cutoff are connected by an effective harmonic spring. Additionally, the HeteroENM approach optimizes the force constants of every pairwise spring in order to reproduce its mean-squared distance fluctuations. In practice, a uniform set of force constants are assigned to the model initially and then followed by iterative updates via

$$\frac{1}{4k_{ij}^{n+1}} = \frac{1}{4k_{ij}^{n}} - \alpha\left(\Delta x_{ij,NMA}^2 - \Delta x_{ij,MD}^2\right) \tag{20}$$

where k_{ij}^n is the force constant for the spring between the CG sites i and j in the iteration n, and $\Delta x_{ij,NMA}^2$ is the fluctuation of the pairwise distance calculated by normal mode analysis on the minimized structure from the trial set of force constants $\{k_{ij}^n\}$.

2.4 Features and Implementations

OpenMSCG is an open-source package developed mainly using *Python3*, with computationally intensive tasks, i.e., the Verlet-List algorithm, developed and optimized in C++ and wrapped as extensions to the Python framework. It is a high-performance computing software that can handle the modeling of large-scale systems and is also capable of using multi-threading and parallel computing techniques on supercomputers.

The OpenMSCG software package can be used in two ways. First, the essential tools, such as CGMAP, CGFM and CGREM, are provided as command-line-interface scripts, which can be directly launched with specified input files and runtime options. Second, all components, including the modules and tools described above, are wrapped as Python packages with an API, which can be called by users to develop their own custom modeling workflows. This feature will allow users to script their full modeling workflow, publish the scripts (or Jupyter-Notebooks) to demonstrate the details of the work. Therefore, OpenMSCG will greatly improve the reliability, reproducibility, and sharing of bottom-up CG models and applications.

The software package is prepared and released in *Anaconda Cloud*, providing an easy and user-friendly way for installations and upgrades. The source code is under version control on GitLab, from which users can download the package for customized installation and development (https://software.rcc.uchicago.edu/git/MSCG/openmscg).

3 Examples and Discussion

3.1 Liquid-Vapor Interface of a Methanol Droplet

To demonstrate the capabilities of the OpenMSCG package, MS-CG and UCG (RLE) models for a methanol-vapor interface were constructed. In this work, we focus on the formation of a droplet-like methanol cluster, which can be thought of as an extension of the liquid-vapor slab structure that was reported in the previous UCG study.[22]

Atomistic Simulations: The all-atom system was built by placing 1728 methanol molecules into a 49.25Å×49.25Å×49.25Å cubic box. The simulation was performed in GROMACS[39] using OPLS/AA force-field parameters[63] in the constant NVT ensemble with a Nosé-Hoover thermostat[64, 65] at 298.15K. The Particle-Mesh-Ewald technique[66] was used for the long-range electrostatic interactions, and all covalent bonds with hydrogen atoms were treated by the LINear Constraint Solver (LINCS) algorithm[67]. The system was simulated for 10 nanoseconds (ns) until a droplet structure in the vacuum was formed and maintained over the course of time (Figure 3A). Finally, in order to obtain the trajectory for CG parameterization, the constant NVT simulation was performed for an additional 5 ns.

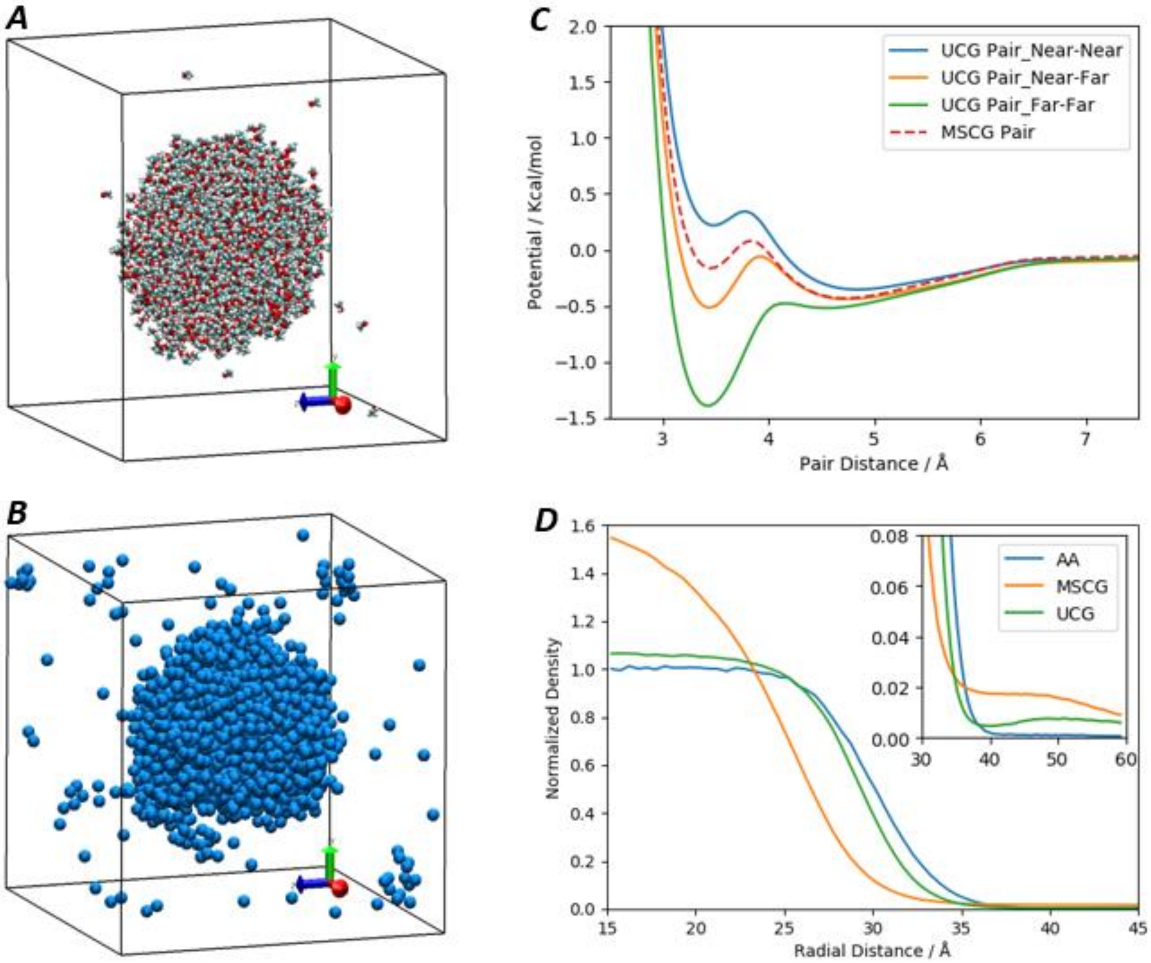


Figure 3. Example of UCG modeling for the liquid-vapor interface of a methanol droplet. (A) A snapshot from the all-atom MD simulation. (B) A snapshot from the UCG simulation demonstrates similar liquid-vapor structures. (C) The pair potentials from MS-CG (dashed line) and UCG (solid lines) methods. (D) The radial density profiles originated from the center of the droplet in simulations with all-atom, MS-CG, and UCG models.

CG Mapping and Modeling: From the generated all-atom trajectory, the CG model was parameterized from the manually mapped CG trajectory using the center-of-mass of each methanol molecule, given by the CGMAP tool in OpenMSCG. Then, using the pairwise approximation, the effective pairwise potentials between the methanol CG sites were determined. The tabulated MS-CG and UCG potentials were built using 3rd order B-splines covering the pair distance from 2.8 Å to 10.0Å with a knot spacing of 0.1Å. For UCG modeling, we adopt the rapid local equilibrium limit[22]. Based on the non-uniform nature exhibited by a droplet structure, the internal sites for methanol are distinguished via the local density. Following the earlier work on the local density-based UCG models,[21] the internal states are further defined as "denser (α)" or "less dense (β)" states with the substate probabilities assigned by:

$$p_{I,\alpha}(\rho) = \frac{1}{2}\left(1 + \tanh\left(\frac{\rho_I - \rho_{th}}{0.1\rho_{th}}\right)\right) \tag{21}$$

$$p_{I,\beta}(\rho) = 1 - p_{I,\alpha}(\rho) \tag{22}$$

in which ρ_I is the number density for the CG site I and can be obtained by a local proximity function from all neighboring methanol sites:

$$\rho_I = \sum_{J \text{ neigh } I} \frac{1}{2}\left(1 - \tanh\left(\frac{r_{IJ} - r_{th}}{0.1\mathrm{r}_{th}}\right)\right). \tag{23}$$

By design, methanol with a higher local density corresponds to the denser state, which should resemble bulk methanol, whereas the less dense state can describe the interface of the droplet. Following a prior study on density-based methanol UCG models, we employed the same UCG state parameters[22] for Eqs. 21-23: $r_{th} = 4.5$ Å and $\rho_{th} = 3.5$. After defining the UCG states, the state-wise interaction parameters were determined from the CGFM tool, which was also used for the (conventional) MS-CG model. Finally, the MS-CG and UCG simulations were performed using the LAMMPS MD package[41].

CG Model Validation: From the MS-CG and UCG simulations, we can assess the performance of CG models by calculating the normalized radial density profiles:

$$\rho(\mathrm{r}) = \frac{1}{4\pi \mathrm{r}\rho_0}\langle\sum_i \delta(|\boldsymbol{r}_i - \boldsymbol{r}_{COM}| - r)\rangle \tag{24}$$

where ρ_0 is the standard number density of methanol molecules in the bulk phase, and $\boldsymbol{r}_{COM}$ is the center-of-mass of the liquid droplet. The results for MS-CG and UCG models are shown in Figure 3D in comparison with the all-atom reference. Consistent with a finely-detailed description of the slab structure in the liquid-vapor interface for methanol,[22] the UCG model can produce an almost identical interfacial structure as the all-atom model for the droplet morphology as well. In contrast, as expected, the conventional MS-CG model yields an inaccurate density profile with a broad transition region, where molecules near the center of the droplet are overly attracted. This discrepancy is attributed to the effective CG potentials shown in Figure 3(C). In the UCG model, methanol near the interface region or gas phase is likely to be in the less dense state for which the associated pairwise CG potentials have stronger, attractive interactions to maintain the interface structure. On the other hand, molecules in the liquid phase are more likely to be in the denser state that is dominated by a pairwise potential similar to the bulk interaction. Therefore, by introducing an order parameter to the FM procedure, the molecular nature underlying the interface system can be readily captured by different statewise interactions. During the UCG simulation, the UCG site can reflect the local chemical environment and adjust its corresponding interaction, whereas the conventional MS-CG method is limited to a single potential to describe CG sites at different chemical environments, resulting in an averaged interaction of the UCG substate interactions.

3.2 Methanol-Hexane Interface

In addition to the methanol vapor-liquid droplet system, we illustrate the applicability of the MS-CG and UCG modules in OpenMSCG for a more complicated liquid-liquid two-component system. We note that the original UCG work studied the liquid-liquid interface consisting of methanol and carbon tetrachloride,[22] and we now validate the performance of OpenMSCG modules for a similar system: the liquid-liquid interface of methanol-hexane. Particularly, we investigate a methanol and hexane mixture with a 0.65-0.35 mole fraction, which has been experimentally observed to undergo phase separation and exhibit an upper critical solution temperature of 313 K[68].

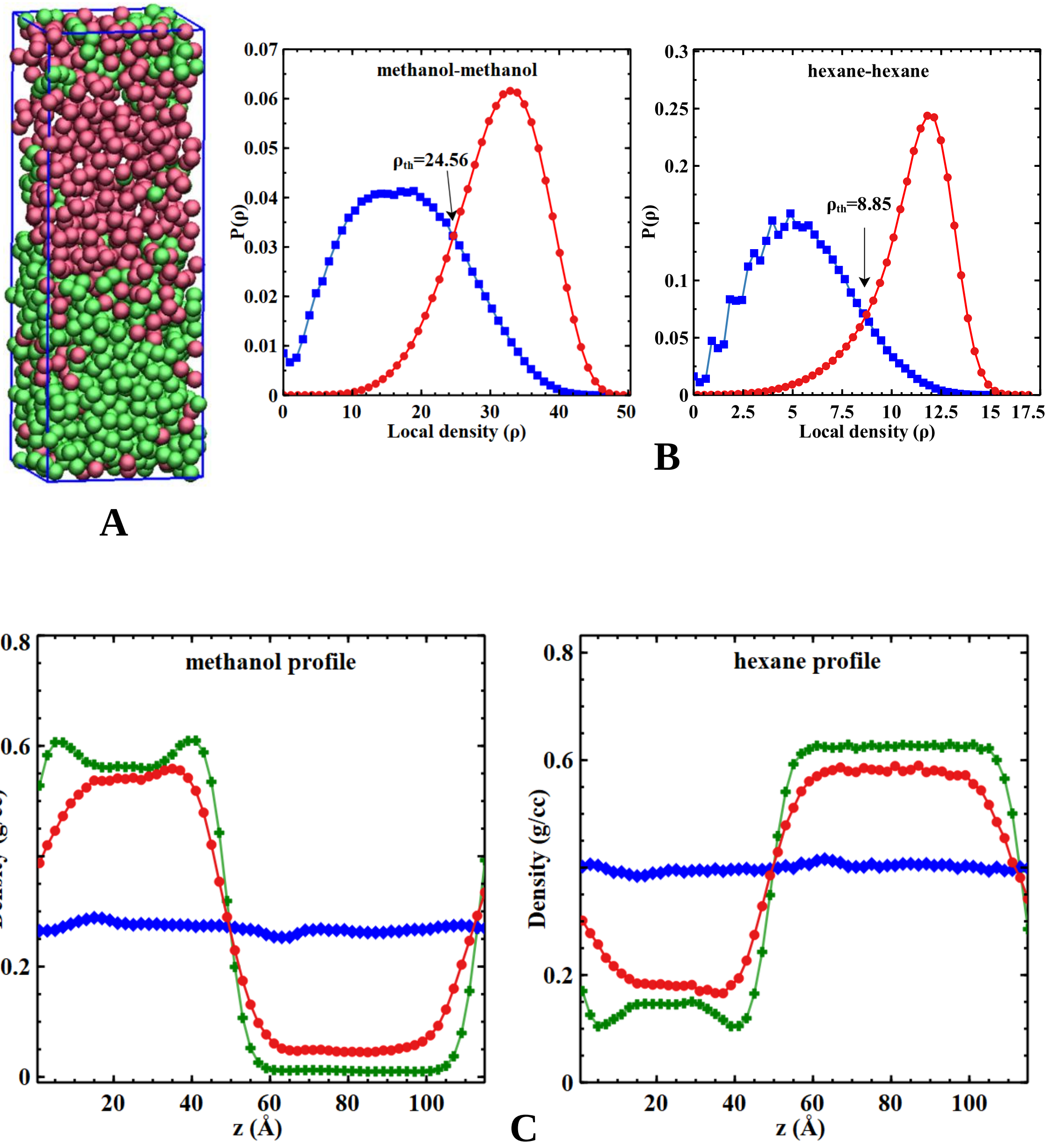


Figure 4. Example use of OpenMSCG for methanol-hexane interface. (A) A snapshot of the methanol (green) and hexane (red) interfaces at the CG level. (B) Local density distributions of methanol-methanol (left) and hexane-hexane (right) in the high density (red circles) and low density (blue squares) regions. (C) Comparison of density profiles of methanol (left) and hexane (right) from the MS-CG (blue diamonds) and UCG (green plus) models with the reference all-atom structures (red circles).

Atomistic Simulations: The all-atom slab system consists of 1000 molecules of methanol and 550 molecules of hexane arranged in a slab-like geometry in the z-direction (Figure 4A) under the periodic boundary condition (PBC). The slab model was constructed from equilibrated and relaxed bulk systems of methanol (containing 1000 molecules) and hexane (containing 550 molecules) separately. Specifically, these bulk systems were equilibrated in the constant NPT ensemble for a period of 4 ns with subsequent constant NVT relaxation for a period of 5 ns. Following the relaxations, the x- and y- dimensions of the bulk methanol and hexane were adjusted to have the same values, and the z- dimension was simultaneously adjusted to maintain their respective densities. Then, the bulk liquids were combined along the z-direction to form the layered slab (Figure 4A)**.** The layered model was then relaxed for 6 ns in the constant NPT ensemble, followed by production runs for 5 ns in the constant NVT ensemble. Intermediate snapshots during the constant NVT simulation were collected every 1 ps during the production runs, which were subsequently used to construct the CG models for parameterization.

All the atomistic simulations described above were performed at 295 K and 1 atm using the LAMMPS[41] MD package with a timestep of 1 fs. The General AMBER Force Field (GAFF)[69] was used as an atomistic force field, where the partial atomic charges for calculating coulombic forces were determined using the AM1-BCC method[70, 71]. A force cut-off distance of 10 Å was used for both the pairwise Lennard-Jones (LJ) and the coulombic interactions. The LJ interactions were shifted such that they smoothly decay to zero beyond the cut-off distance. For coulombic interactions, particle-particle particle-mesh (PPPM)[72, 73] solvers were employed to account for their contributions beyond the cut-off distance. A Nosé-Hoover thermostat with a damping constant of 0.1 ps and Nosé-Hoover barostat with a damping constant of 1 ps were used to maintain the reference temperature and pressure of the systems, respectively.

CG Mapping: From the all-atom trajectory, the CG trajectory for parameterization was constructed using the CGMAP command of the OpenMSCG package. Both methanol and hexane molecules were mapped onto their center-of-mass at a resolution of one site per molecule. From the mapped CG trajectory, the MS-CG and UCG potentials were determined via FM, supported by the CGFM command of the OpenMSCG package.

MS-CG and UCG Potentials for the Liquid-Liquid Interface: For the MS-CG potentials, a linear combination of B-splines with a resolution of 0.2 Å was used to represent all pairwise interactions (methanol-methanol, hexane-hexane, and methanol-hexane) with inner and outer cut-offs of 4 Å and 12 Å, respectively.

Analogous to the MS-CG models, the UCG interactions were expressed using the same linear combination of B-splines for the substate interactions. The inner and outer cut-off of 3 Å and 8 Å was used for methanol-methanol, 4 Å and 8 Å for hexane-hexane, and 3.5 Å and 8 Å for methanol-hexane interactions. As described earlier, the main purpose of the UCG methodology is to enhance the expressivity of CG models by coupling the CG interactions with a relevant order parameter. Like the methanol droplet the local self-density, which is the number of the same molecular entities within a given radius, can serve as an order parameter to distinguish the phase-separated nature in interface. To note, the cross-density was employed for the UCG models of methanol-carbon tetrachloride,[22] and a general discussion on the definition of local density is given by Vanya et al.[74] Since the methanol-hexane interface exhibits locally high and low density regions of each molecule, as seen from the density profile (Figure 4C, red curve), the use of self-density, given by Eq. 23, as an order parameter is expected to modulate the interactions of methanol or hexane based on their local neighbors. The calculated local self-density was subsequently used to estimate the probabilities of the UCG substates (denser and less dense states) using Eqs. 21-22. To properly distinguish the denser and less dense states, the variables required for defining the UCG states (R_{th} and ρ_{th}) were determined based on the order parameter histogram constructed from the all-atom simulations. To minimize the overlap between substates, the density cut-off parameter was chosen as the density value where two substate histograms intersect (Figure 4B). Table 1 lists the final state parameters for the UCG model of the methanol-hexane liquid-liquid interface.

Table 1. The distance and local self-density cut-off (UCG state parameters) for the methanol-hexane interface.

Molecules	$\boldsymbol{R_{th}}$ **(Å)**	$\boldsymbol{\rho_{th}}$	$\boldsymbol{\rho}$ **type**
Methanol	9.00	24.56	Self-density
Hexane		8.85	

Having determined the UCG internal states, the effective UCG state-wise potentials were obtained via the dynamic type method described in Sec. II. To accurately embed the substate probability information into the FM framework, the mapped CG trajectory was replicated 50 times, and the CGFM command was employed to infer the UCG state-wise potentials.

CG Model Validation: We then calculated the structural characteristics of MS-CG and UCG models for methanol-hexane interface. As seen from the density profiles in Figure 4C, the MS-CG model fails to distinguish the different phases of each molecule and instead produces a single-phase system. This discrepancy is likely due to the inability of the MS-CG model to capture the heterogeneous nature of the system in different phases. Nevertheless, the UCG model overcomes this limitation by readily distinguishing the different phases using the self-density order parameter, resulting in an enhanced density profile compared to that from the MS-CG simulations (Figure

4C). Notably, the UCG model shows the phase-separated behavior and maintains the slab-like geometry of the system, while there are slight deviations from the all-atom reference at the edge of interface, as seen in Figure 4C**.** This is consistent with other CG models for interfaces, for which a limited number of internal states are used to describe the phase boundary.[24] To summarize, in this section, we utilize the OpenMSCG package to construct the UCG models for the methanol droplet and methanol-hexane interface. As expected from the previous success with local density-based UCG models, we demonstrated that the UCG models can greatly enhance the structural correlations by distinguishing the local molecular environment described by the order parameter, whereas conventional MS-CG models fail to do so. These results show the capabilities of OpenMSCG package that one can make use of to build highly robust and predictive CG and UCG models of complex systems with a choice of appropriate order parameters. Hence, future work would focus on how to determine the optimal order parameter from complex, arbitrary atomistic systems.

3.3 HIV CA/SP1 Interactions with REM

To demonstrate the REM capabilities of the OpenMSCG package, a CG model of the HIV-1 capsid (CA) and spacer peptide 1 (SP1) polyprotein, which assembles into the protein shell of the immature virus[75], was built and validated by comparison between reference and model RDFs. The CG model was systematically derived from all-atom MD simulations as described further below.

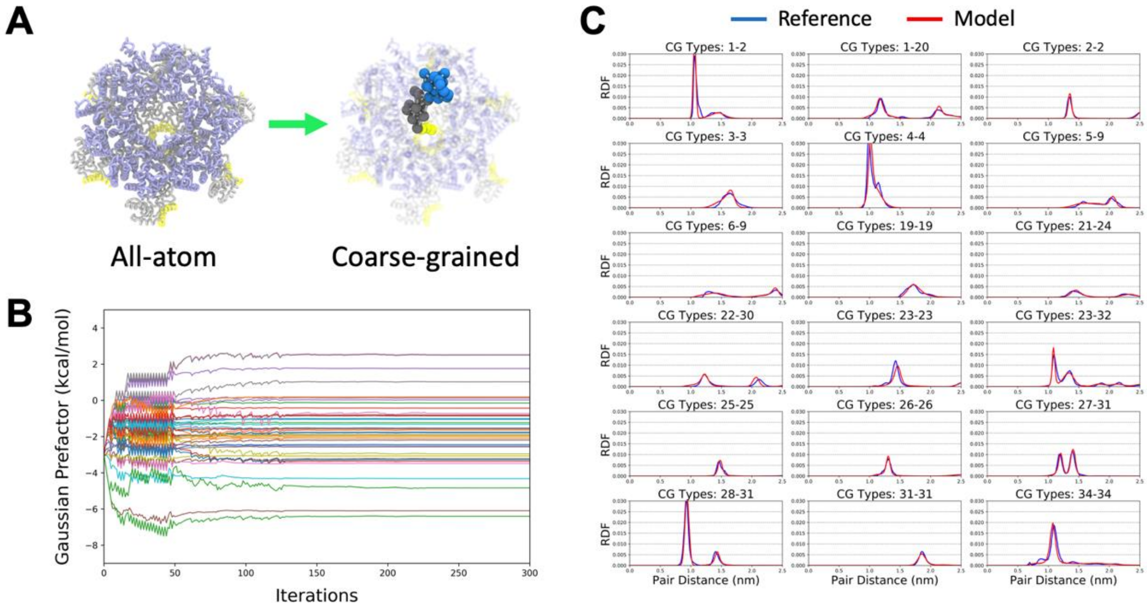


Figure 5. Example use of OpenMSCG to optimize HIV-1 CA/SP1 interactions using REM. (A) Schematic of the all-atom HIV-1 CA/SP1 oligomer that is used as reference to generate the CG model. (B) Profiles depicting parameter changes across iterations during REM. Each color depicts a distinct Gaussian prefactor. (C) Comparison of RDFs for select CG pairs computed from the all-atom reference (blue) and CG model (red).

Atomistic Simulations: The atomistic simulation was prepared using the atomic model from PDB 5L93,[76] an 18-mer CA/SP1 system (see Fig. 5A) and described using the CHARMM36m force-field.[77] The system was solvated using TIP3P water[78] and 150 mM NaCl that extended beyond the surface of the protein by 2 nm. The simulation was integrated using GROMACS[39] with a timestep of 2 fs and PBC in all directions. The system was equilibrated for 50 ns in the constant NPT ensemble using a Nosé-Hoover thermostat[64, 65] at 300 K with a 1 ps damping constant and a Parrinello-Rahman barostat[79] at 1 atm with a 5 ps damping constant. The simulation continued for 1200 ns in the constant NVT ensemble at 300 K with a 2 ps damping constant. The final 1000 ns was used as reference statistics with configurations saved every 40 ps.

CG Model Generation: The all-atom trajectory was mapped to CG space using linear EDCG where $N = 35$, i.e., each CA/SP1 monomer was mapped to a 35-site CG model. Intra-protein interactions were described by HeteroENM using a radial cut-off of 3 nm. Inter-protein interactions were described by a combination of an excluded volume potential (E_{excl}), screened Coulombic potential (E_{coul}), and a Gaussian potential (E_{gauss}) for close contacts. For E_{excl}, a soft cosine potential, $A\left[1 + cos\left(\frac{\pi r}{r_c}\right)\right]$, was used, where $A = 25$ kcal/mol and $r_c\ =\ 10$ Å. For E_{coul}, a Yukawa potential,[80] $\frac{q_i q_j}{4\pi\epsilon_r\epsilon_0 r_{ij}} e\left(-\kappa r_{ij}\right)$, was used, where q_i is the aggregate charge of CG site $i, \kappa = 0.1274$ Å^{-1} is the inverse Debye length for 150 mM NaCl, and ϵ_r is the effective dielectric constant of the protein environment , approximated as 17.5.[81] For E_{gauss}:

$$E_{gauss}(r) = \frac{H}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(r - r_0)^2}{2\sigma^2}\right) \tag{25}$$

where H , σ , and r_0 are the Gaussian prefactor, variance, and minimum energy distance, respectively. All CG pairwise contacts with an average distance less than 1.75 nm were considered close contacts. For each pairwise interaction, σ and r_0 were computed using peak fitting functions from the *Scipy* software package.[54] While these parameters were fixed, each H was optimized using REM; the initial H was uniformly initialized to -3 kcal/mol. The CG statistics for each iteration were generated using LAMMPS and the current CG force field, which was run for 21×10^6 steps using a 50 fs timestep and a Langevin thermostat (10 ps damping time) at 300 K. Statistics were gathered every 800 steps over the final 20×10^6 steps. We used the iterative Newton-Raphson method to update each H shown in Eq. 14. To aid convergence during training, a learning rate schedule for χ was implemented:

1. $\chi = 0.5$ during the first 50 iterations

2. $\chi = 0.1$ during the next 50 iterations
3. $\chi = 0.01$ during the next 200 iterations

A total of 300 iterations were used to generate the CG model. As seen in Fig. 5(b), the iterative procedure was stopped as the change to each parameter was effectively zero.

CG Model Validation: After CG parameter optimization, we tested the fidelity of the generated model by comparing pair correlation statistics against the all-atom reference. In Figure 5C, we show a selection of RDFs between pairs that are both close contacts and non-close (e.g. pairs 5-9, 19-19, and 31-31) contacts to demonstrate that pair correlations beyond those imposed by the attractive Gaussian interaction are preserved. It is evident from Figure 5C that the CG model recapitulates the peak positions and variance of the reference pair correlations. Due to the simplicity of the Gaussian functional form, which was chosen to simplify model parametrization, some secondary peak features of the reference RDFs, as seen in the 4-4 and 34-34 pair correlations, were not captured by the CG model. A user may choose to increase the complexity of the functional form, which may be accomplished by using multiple Gaussians or B-splines, in order to increase fidelity to the reference RDFs. Both functional forms have been implemented in OpenMSCG, with additional functional forms to be implemented in the future.

3.4 Comparison with Other Software Packages

Other available CG model software packages that have similar functionalities to OpenMSCG include MagiC,[82, 83] VOTCA,[84, 85] and BOCS.[86] MagiC was developed by Lyubartsev's group in 2013,[82] with a newer 3.0 version released in 2018.[83] MagiC provides a set of Fortran-based tools for CG mapping and effective potential calculations with Inverse Monte Carlo (IMC) and IBI methods, as well as several Python scripts for data post-processing. VOTCA was developed by Ruhle et al. in 2009,[85] and contains the functionalities of IMC, IBI as well as FM using cubic splines. In 2016, the REM method was also added to this software by de Oliveira et al.[84] Beyond the conventional MS-CG method, the BOCS software, which was released in 2017 by the Noid group,[86] also provides another tool for potential parameterization called the generalized Yvon-Born-Green method[87, 88] and pressure-matching method based on the Das-Andersen Hamiltonian.[14] All of these software packages are focused on providing full workflows using a bottom-up multiscale strategy to construct CG effective potentials from all-atom simulation data.

OpenMSCG has been developed to support a large group of multiscale modeling methodologies, including popular potential parameterization methods such as BI, FM, and REM, as well as the recently introduced UCG method. These parameterization tools utilize, but are not limited to, B-splines as a functional form for the CG effective potentials. As OpenMSCG was developed as an open framework, it is possible for users to create a new functional form in Python to be imported in either FM or REM workflows. Since OpenMSCG is highly modularized, users can invoke and customize any existing modules within it to develop new approaches for MS-CG-based modeling. For example, in the FM module, users can develop a customized Python module for the regularization procedure to be imported by the software. Another example is that the REM module also allows a plug-in module customized by users for updating/tuning the CG potential parameters in the iterative procedure. Combining both modules in a hybrid manner, one can also possibly utilize the advantages of FM and REM methodologies to construct new forms of high-fidelity CG models.[89]

4 Conclusions

In bottom-up multiscale coarse-grained modeling, there is no single unified approach that can encompass different conditions and resolutions of complex systems due to its data-driven nature. Even within the same molecular systems, CG models are generally not transferable between different thermodynamic conditions, e.g., temperature or density, known as the transferability problem.[2, 4, 24, 35-38] However, the workflows and protocols of MS-CG modeling are transferable and sharable. In this work, we developed the software package, OpenMSCG, for the purpose of enabling highly systematic and reproducible work from MS-CG modeling workflows. OpenMSCG integrates a comprehensive toolset to build up CG models and parameterize CG effective potentials from all-atom MD simulations. The structure of the software is well organized and documented and is easy to install (via Anaconda Cloud) and extend (via Python programming). Additionally, most of the tools are implemented with multithreading or distributed parallelization features to handle large MD data for complex systems. The benchmark systems presented in this paper demonstrate that OpenMSCG can produce reliable CG and UCG models that accurately reproduce the structural properties from all-atom models. In particular, we extended the conventional FM technique to the UCG theory based on the original algorithm which can extend the range and physical accuracy of CG models for more complex heterogeneous systems.[19] The vision of OpenMSCG is to not only enhance the performance and reproducibility of MS-CG models but also attract more researchers and contributors into the bottom-up CG modeling community. We also hope that the OpenMSCG software will be continuously updated to implement the new advances in bottom-up CG methodologies.

ACKNOWLEDGEMENTS

Research was supported by the U.S. Department of Energy (DOE), Office of Science, Basic Energy Sciences (BES), under Award DE-SC0023318 (latter phase software development from methods) and by the National Science Foundation (NSF) under Awards CHE-2102677 and OAC-

1740211 (first phase software development effort from methods). Simulations were performed using computing resources provided by the University of Chicago Research Computing Center (RCC).

REFERENCES

1. Saunders, M. G.; Voth, G. A., Coarse-graining methods for computational biology. *Annu. Rev. Biophys.* **2013**, *42*, 73-93.
2. Noid, W. G., Perspective: Coarse-grained models for biomolecular systems. *J. Chem. Phys.* **2013**, *139* (9).
3. Marrink, S. J.; Risselada, H. J.; Yefimov, S.; Tieleman, D. P.; de Vries, A. H., The MARTINI force field: Coarse grained model for biomolecular simulations. *J. Phys. Chem. B* **2007**, *111* (27), 7812-7824.
4. Jin, J.; Pak, A. J.; Durumeric, A. E. P.; Loose, T. D.; Voth, G. A., Bottom-up Coarse-Graining: Principles and Perspectives. *J. Chem. Theory Comput.* **2022**, *18* (10), 5759-5791.
5. Friedel, M.; Sheeler, D. J.; Shea, J. E., Effects of confinement and crowding on the thermodynamics and kinetics of folding of a minimalist beta-barrel protein. *J. Chem. Phys.* **2003**, *118* (17), 8106-8113.
6. Sorensen, J. M.; Head-Gordon, T., Toward minimalist models of larger proteins: A ubiquitin-like protein. *Proteins* **2002**, *46* (4), 368-379.
7. Izvekov, S.; Voth, G. A., A multiscale coarse-graining method for biomolecular systems. *J. Phys. Chem. B* **2005**, *109* (7), 2469-2473.
8. Izvekov, S.; Voth, G. A., Multiscale coarse graining of liquid-state systems. *J. Chem. Phys.* **2005**, *123* (13).
9. Noid, W. G.; Chu, J.-W.; Ayton, G. S.; Voth, G. A., Multiscale Coarse-Graining and Structural Correlations: Connections to Liquid-State Theory. *The J. Phys. Chem. B* **2007**, *111* (16), 4116-4127.
10. Noid, W. G.; Chu, J. W.; Ayton, G. S.; Krishna, V.; Izvekov, S.; Voth, G. A.; Das, A.; Andersen, H. C., The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models. *J. Chem. Phys.* **2008**, *128* (24).
11. Noid, W. G.; Liu, P.; Wang, Y.; Chu, J. W.; Ayton, G. S.; Izvekov, S.; Andersen, H. C.; Voth, G. A., The multiscale coarse-graining method. II. Numerical implementation for coarse-grained molecular models. *J. Chem. Phys.* **2008**, *128* (24).
12. Ercolessi, F.; Adams, J. B., Interatomic Potentials from First-Principles Calculations: The Force-Matching Method. *Europhys. Lett.* **1994**, *26* (8), 583-588.
13. Izvekov, S.; Parrinello, M.; Burnham, C. J.; Voth, G. A., Effective force fields for condensed phase systems from ab initio molecular dynamics simulation: A new method for force-matching. *The Journal of Chemical Physics* **2004**, *120* (23), 10896-10913.
14. Larini, L.; Lu, L. Y.; Voth, G. A., The multiscale coarse-graining method. VI. Implementation of three-body coarse-grained potentials. *J. Chem. Phys.* **2010**, *132* (16), 10.
15. Das, A.; Andersen, H. C., The multiscale coarse-graining method. IX. A general method for construction of three body coarse-grained force fields. *J. Chem. Phys.* **2012**, *136* (19).

16. Jin, J.; Han, Y. N.; Voth, G. A., Coarse-graining involving virtual sites: Centers of symmetry coarse-graining. *J. Chem. Phys.* **2019**, *150* (15), 15.
17. Cao, Z.; Voth, G. A., The multiscale coarse-graining method. XI. Accurate interactions based on the centers of charge of coarse-grained sites. *J. Chem. Phys.* **2015**, *143* (24).
18. Han, Y.; Jin, J.; Wagner, J. W.; Voth, G. A., Quantum theory of multiscale coarse-graining. *J. Chem. Phys.* **2018**, *148* (10), 102335.
19. Dama, J. F.; Sinitskiy, A. V.; McCullagh, M.; Weare, J.; Roux, B.; Dinner, A. R.; Voth, G. A., The theory of ultra-coarse-graining. 1. General principles. *J. Chem. Theory Comput.* **2013**, *9* (5), 2466-2480.
20. Davtyan, A.; Dama, J. F.; Sinitskiy, A. V.; Voth, G. A., The theory of ultra-coarse-graining. 2. Numerical implementation. *J. Chem. Theory Comput.* **2014**, *10* (12), 5265-5275.
21. Dama, J. F.; Jin, J.; Voth, G. A., The Theory of Ultra-Coarse-Graining. 3. Coarse-Grained Sites with Rapid Local Equilibrium of Internal States *J. Chem. Theory Comput.* **2018**, *13* (3), 1010-1022.
22. Jin, J.; Voth, G. A., Ultra-coarse-grained models allow for an accurate and transferable treatment of interfacial systems. *J. Chem. Theory Comput.* **2018**, *14* (4), 2180-2197.
23. Jin, J.; Han, Y. N.; Voth, G. A., Ultra-coarse-grained liquid state models with implicit hydrogen bonding. *J. Chem. Theory Comput.* **2018**, *14* (12), 6159-6174.
24. Jin, J.; Yu, A.; Voth, G. A., Temperature and phase transferable bottom-up coarse-grained Models. *J. Chem. Theory Comput.* **2020**, *16* (11), 6823-6842.
25. Katkar, H. H.; Davtyan, A.; Durumeric, A. E. P.; Hocky, G. M.; Schramm, A. C.; De La Cruz, E. M.; Voth, G. A., Insights into the Cooperative Nature of ATP Hydrolysis in Actin Filaments. *Biophys. J.* **2018**, *115* (8), 1589-1602.
26. Mani, S.; Katkar, H. H.; Voth, G. A., Compressive and Tensile Deformations Alter ATP Hydrolysis and Phosphate Release Rates in Actin Filaments. *J Chem Theory Comput* **2021**, *17* (3), 1900-1913.
27. Grime, J. M. A.; Dama, J. F.; Ganser-Pornillos, B. K.; Woodward, C. L.; Jensen, G. J.; Yeager, M.; Voth, G. A., Coarse-grained simulation reveals key features of HIV-1 capsid self-assembly. *Nat Commun* **2016**, *7*, 11568.
28. Gupta, M.; Pak, A. J.; Voth, G. A., Critical mechanistic features of HIV-1 viral capsid assembly. *Sci Adv* **2023**, *9* (1), eadd7434.
29. Shell, M. S., The relative entropy is fundamental to multiscale and inverse thermodynamic problems. *J. Chem. Phys.* **2008**, *129* (14).
30. Chaimovich, A.; Shell, M. S., Relative entropy as a universal metric for multiscale errors. *Phys. Rev. E* **2010**, *81* (6), 060104.
31. Chaimovich, A.; Shell, M. S., Coarse-graining errors and numerical optimization using a relative entropy framework. *J. Chem. Phys.* **2011**, *134* (9).
32. Shell, M. S., Coarse-Graining with the Relative Entropy. In *Advances in Chemical Physics*, 2016; pp 395-441.
33. Tschop, W.; Kremer, K.; Batoulis, J.; Burger, T.; Hahn, O., Simulation of polymer melts. I. Coarse-graining procedure for polycarbonates. *Acta Polymerica* **1998**, *49* (2-3), 61-74.
34. Lyman, E.; Pfaendtner, J.; Voth, G. A., Systematic multiscale parameterization of heterogeneous elastic network models of proteins. *Biophys. J.* **2008**, *95* (9), 4183-4192.
35. Dunn, N. J. H.; Foley, T. T.; Noid, W. G., Van der waals perspective on coarse-graining: Progress toward solving representability and transferability problems. *Acc. Chem. Res.* **2016**, *49* (12), 2832-2840.

36. Wagner, J. W.; Dama, J. F.; Durumeric, A. E. P.; Voth, G. A., On the representability problem and the physical meaning of coarse-grained models. *J. Chem. Phys.* **2016**, *145* (4), 044108.
37. Jin, J.; Pak, A. J.; Voth, G. A., Understanding Missing Entropy in Coarse-Grained Systems: Addressing Issues of Representability and Transferability. *J. Phys. Chem. Lett.* **2019**, *10* (16), 4549-4557.
38. Potter, T. D.; Tasche, J.; Wilson, M. R., Assessing the transferability of common top-down and bottom-up coarse-grained molecular models for molecular mixtures. *Phys. Chem. Chem. Phys.* **2019**, *21* (4), 1912-1927.
39. Pronk, S.; Pall, S.; Schulz, R.; Larsson, P.; Bjelkmar, P.; Apostolov, R.; Shirts, M. R.; Smith, J. C.; Kasson, P. M.; van der Spoel, D.; Hess, B.; Lindahl, E., GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics* **2013**, *29* (7), 845-854.
40. Phillips, J. C.; Hardy, D. J.; Maia, J. D. C.; Stone, J. E.; Ribeiro, J. V.; Bernardi, R. C.; Buch, R.; Fiorin, G.; Henin, J.; Jiang, W.; McGreevy, R.; Melo, M. C. R.; Radak, B. K.; Skeel, R. D.; Singharoy, A.; Wang, Y.; Roux, B.; Aksimentiev, A.; Luthey-Schulten, Z.; Kale, L. V.; Schulten, K.; Chipot, C.; Tajkhorshid, E., Scalable molecular dynamics on CPU and GPU architectures with NAMD. *J. Chem. Phys.* **2020**, *153* (4).
41. Plimpton, S., Fast parallel algorithms for short-grange molecular-dynamics. *J. Comp. Phys.* **1995**, *117* (1), 1-19.
42. MacKerell, A. D.; Bashford, D.; Bellott, M.; Dunbrack, R. L.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; Joseph-McCarthy, D.; Kuchnir, L.; Kuczera, K.; Lau, F. T. K.; Mattos, C.; Michnick, S.; Ngo, T.; Nguyen, D. T.; Prodhom, B.; Reiher, W. E.; Roux, B.; Schlenkrich, M.; Smith, J. C.; Stote, R.; Straub, J.; Watanabe, M.; Wiorkiewicz-Kuczera, J.; Yin, D.; Karplus, M., All-atom empirical potential for molecular modeling and dynamics studies of proteins. *J. Phys. Chem. B* **1998**, *102* (18), 3586-3616.
43. Ponder, J. W.; Case, D. A., Force fields for protein simulations. In *Protein Simulations*, Daggett, V., Ed. Elsevier Academic Press Inc: San Diego, 2003; Vol. 66, pp 27-+.
44. Stillinger, F. H.; Weber, T. A., Computer-simulation of local order in condensed phases of silicon. *Phy. Rev. B* **1985**, *31* (8), 5262-5271.
45. Jin, J.; Pak, A. J.; Han, Y. N.; Voth, G. A., A new one-site coarse-grained model for water: Bottom-up many-body projected water (BUMPer). II. Temperature transferability and structural properties at low temperature. *J. Chem. Phys.* **2021**, *154* (4).
46. Jin, J.; Han, Y. N.; Pak, A. J.; Voth, G. A., A new one-site coarse-grained model for water: Bottom-up many-body projected water (BUMPer). I. General theory and model. *J. Chem. Phys.* **2021**, *154* (4).
47. Lu, L. Y.; Voth, G. A., Systematic coarse-graining of a multicomponent lipid bilayer. *J. Phys. Chem. B* **2009**, *113* (5), 1501-1510.
48. Lu, L.; Izvekov, S.; Das, A.; Andersen, H. C.; Voth, G. A., Efficient, regularized, and scalable algorithms for multiscale coarse-graining. *J. Chem. Theory Comput.* **2010**, *6* (3), 954-965.
49. Wang, J.; Olsson, S.; Wehmeyer, C.; Perez, A.; Charron, N. E.; de Fabritiis, G.; Noe, F.; Clementi, C., Machine Learning of Coarse-Grained Molecular Dynamics Force Fields. *ACS Cent Sci* **2019**, *5* (5), 755-767.

50. Wang, H.; Zhang, L.; Han, J.; E, W., DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics. *Computer Physics Communications* **2018**, *228*, 178-184.
51. Husic, B. E.; Charron, N. E.; Lemm, D.; Wang, J.; Perez, A.; Majewski, M.; Kramer, A.; Chen, Y.; Olsson, S.; de Fabritiis, G.; Noe, F.; Clementi, C., Coarse graining molecular dynamics with graph neural networks. *J Chem Phys* **2020**, *153* (19), 194101.
52. Kramer, A.; Durumeric, A. E. P.; Charron, N. E.; Chen, Y.; Clementi, C.; Noe, F., Statistically Optimal Force Aggregation for Coarse-Graining Molecular Dynamics. *J Phys Chem Lett* **2023**, 3970-3979.
53. Verlet, L., Computer experiments on classical fluids .I. Thermodynamical properties of lennard-jones molecules. *Phys. Rev.* **1967**, *159* (1), 98-103.
54. Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S. J.; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C. J.; Polat, I.; Feng, Y.; Moore, E. W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. N. H.; Pedregosa, F.; van Mulbregt, P.; SciPy, C., SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **2020**, *17* (3), 261-272.
55. Moore, T. C.; Iacovella, C. R.; McCabe, C., Derivation of coarse-grained potentials via multistate iterative Boltzmann inversion. *J. Chem. Phys.* **2014**, *140* (22).
56. Hoerl, A. E.; Kennard, R. W., Ridge Regression - Biased Estimation For Nonorthogonal Problems. *Technometrics* **1970**, *12* (1), 55-&.
57. Liu, P.; Shi, Q.; Daume, H.; Voth, G. A., A Bayesian statistics approach to multiscale coarse graining. *J. Chem. Phys.* **2008**, *129* (21).
58. Pak, A. J.; Voth, G. A., Advances in coarse-grained modeling of macromolecular complexes. *Curr. Opin. Struct. Biol.* **2018**, *52,* 119-126.
59. Souza, P. C. T.; Alessandri, R.; Barnoud, J.; Thallmair, S.; Faustino, I.; Grunewald, F.; Patmanidis, I.; Abdizadeh, H.; Bruininks, B. M. H.; Wassenaar, T. A.; Kroon, P. C.; Melcr, J.; Nieto, V.; Corradi, V.; Khan, H. M.; Domanski, J.; Javanainen, M.; Martinez-Seara, H.; Reuter, N.; Best, R. B.; Vattulainen, I.; Monticelli, L.; Periole, X.; Tieleman, D. P.; de Vries, A. H.; Marrink, S. J., Martini 3: a general purpose force field for coarse-grained molecular dynamics. *Nat. Methods* **2021**, *18* (4), 382-+.
60. Zhang, Z. Y.; Lu, L. Y.; Noid, W. G.; Krishna, V.; Pfaendtner, J.; Voth, G. A., A systematic methodology for defining coarse-grained sites in large biomolecules. *Biophys. J.* **2008**, *95* (11), 5073-5083.
61. Haliloglu, T.; Bahar, I.; Erman, B., Gaussian dynamics of folded proteins. *Phys. Rev. Lett.* **1997**, *79* (16), 3090-3093.
62. Ma, J. P., Usefulness and limitations of normal mode analysis in modeling dynamics of biomolecular complexes. *Structure* **2005**, *13* (3), 373-380.
63. Jorgensen, W. L.; Maxwell, D. S.; TiradoRives, J., Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *Journal of the American Chemical Society* **1996**, *118* (45), 11225-11236.
64. Nose, S., A Unified Formulation Of The Constant Temperature Molecular-Dynamics Methods. *J. Chem. Phys.* **1984**, *81* (1), 511-519.
65. Hoover, W. G., Canoncial Dynamics - Equilibrium Phase-Space Distributions. *Phys. Rev. A* **1985**, *31* (3), 1695-1697.

66. Darden, T.; York, D.; Pedersen, L., Particle Mesh Ewald - An n.log(n) Method For Ewald Sums In Large Systems. *J. Chem. Phys.* **1993**, *98* (12), 10089-10092.
67. Hess, B.; Bekker, H.; Berendsen, H. J. C.; Fraaije, J., LINCS: A linear constraint solver for molecular simulations. *J. Comput. Chem.* **1997**, *18* (12), 1463-1472.
68. Abbas, S.; Satherley, J.; Penfold, R., The liquid-liquid coexistence curve and the interfacial tension of the methanol-n-hexane system. *J. Chem. Soc.-Faraday Trans.* **1997**, *93* (11), 2083-2089.
69. Wang, J. M.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A., Development and testing of a general amber force field (vol 25, pg 1157, 2004). *J. Comput. Chem.* **2005**, *26* (1), 114-114.
70. Jakalian, A.; Bush, B. L.; Jack, D. B.; Bayly, C. I., Fast, efficient generation of high-quality atomic Charges. AM1-BCC model: I. Method. *J. Comput. Chem.* **2000**, *21* (2), 132-146.
71. Jakalian, A.; Jack, D. B.; Bayly, C. I., Fast, efficient generation of high-quality atomic charges. AM1-BCC model: II. Parameterization and validation. *J. Comput. Chem.* **2002**, *23* (16), 1623-1641.
72. Deserno, M.; Holm, C., How to mesh up Ewald sums. II. An accurate error estimate for the particle-particle-particle-mesh algorithm. *J. Chem. Phys.* **1998**, *109* (18), 7694-7701.
73. Deserno, M.; Holm, C., How to mesh up Ewald sums. I. A theoretical and numerical comparison of various particle mesh routines. *J. Chem. Phys.* **1998**, *109* (18), 7678-7693.
74. Vanya, P.; Elliott, J. A., Definitions of local density in density-dependent potentials for mixtures. *Phys. Rev. E* **2020**, *102* (1), 6.
75. Pak, A. J.; Grime, J. M. A.; Sengupta, P.; Chen, A. K.; Durumeric, A. E. P.; Srivastava, A.; Yeager, M.; Briggs, J. A. G.; Lippincott-Schwartz, J.; Voth, G. A., Immature HIV-1 lattice assembly dynamics are regulated by scaffolding from nucleic acid and the plasma membrane. *Proc Natl Acad Sci U S A* **2017**, *114* (47), E10056-E10065.
76. Schur, F. K. M.; Obr, M.; Hagen, W. J. H.; Wan, W.; Jakobi, A. J.; Kirkpatrick, J. M.; Sachse, C.; Krausslich, H. G.; Briggs, J. A. G., An atomic model of HIV-1 capsid-SP1 reveals structures regulating assembly and maturation. *Science* **2016**, *353* (6298), 506-508.
77. Huang, J.; Rauscher, S.; Nawrocki, G.; Ran, T.; Feig, M.; de Groot, B. L.; Grubmuller, H.; MacKerell, A. D., CHARMM36m: an improved force field for folded and intrinsically disordered proteins. *Nat. Methods* **2017**, *14* (1), 71-73.
78. Neria, E.; Fischer, S.; Karplus, M., Simulation of activation free energies in molecular systems. *J. Chem. Phys.* **1996**, *105* (5), 1902-1921.
79. Parrinello, M.; Rahman, A., Polymorphic Transitions in Single Crystals - A New Molecular-Dynamics Method. *J. Appl. Phys.* **1981**, *52* (12), 7182-7190.
80. Yukawa, H., On the Interaction of Elementary Particles. I. *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series* **1935**, *17*, 48-57.
81. Li, L.; Li, C.; Zhang, Z.; Alexov, E., On the Dielectric "Constant" of Proteins: Smooth Dielectric Function for Macromolecular Modeling and Its Implementation in DelPhi. *J Chem Theory Comput* **2013**, *9* (4), 2126-2136.
82. Mirzoev, A.; Lyubartsev, A. P., MagiC: Software package for multiscale modeling. *J. Chem. Theory Comput.* **2013**, *9* (3), 1512-1520.
83. Mirzoev, A.; Nordenskiold, L.; Lyubartsev, A., Magic v.3: An integrated software package for systematic structure-based coarse-graining. *Comput. Phys. Commun.* **2019**, *237*, 263-273.

84. de Oliveira, T. E.; Netz, P. A.; Kremer, K.; Junghans, C.; Mukherji, D., C-IBI: Targeting cumulative coordination within an iterative protocol to derive coarse-grained models of (multi-component) complex fluids. *J. Chem. Phys.* **2016**, *144* (17).

85. Ruhle, V.; Junghans, C.; Lukyanov, A.; Kremer, K.; Andrienko, D., Versatile object-oriented toolkit for coarse-graining applications. *J. Chem. Theory Comput.* **2009**, *5* (12), 3211-3223.

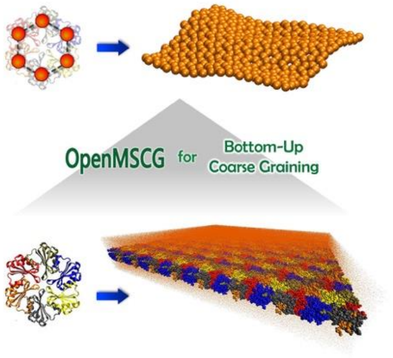
86. Dunn, N. J. H.; Lebold, K. M.; DeLyser, M. R.; Rudzinski, J. F.; Noid, W. G., BOCS: Bottom-up open-source coarse-graining software. *J. Phys. Chem. B* **2018**, *122* (13), 3363-3377.

87. Mullinax, J. W.; Noid, W. G., Generalized Yvon-Born-Green Theory for Molecular Systems. *Phys. Rev. Lett.* **2009**, *103* (19), 4.

88. Mullinax, J. W.; Noid, W. G., A Generalized-Yvon-Born-Green Theory for Determining Coarse-Grained Interaction Potentials. *J. Phys. Chem. C* **2010**, *114* (12), 5661-5674.

89. Pak, A. J.; Dannenhoffer-Lafage, T.; Madsen, J. J.; Voth, G. A., Systematic Coarse-Grained Lipid Force Fields with Semiexplicit Solvation via Virtual Sites. *J. Chem. Theory Comput.* **2019**, *15* (3), 2087-2100.

Table of Contents Graphic



SUPPORTING INFORMATION

for

OpenMSCG: A Software Tool for Bottom-up Coarse-graining

Yuxing Peng,[1] Alexander J. Pak,[2] Aleksander E. P. Durumeric,[3] Patrick G. Sahrmann,[4] Sriramvignesh Mani,[4] Jaehyeok Jin,[4] Timothy D. Loose,[4] Jeriann R. Beiter,[4] and Gregory A. Voth[4, a)]

[1] NVIDIA Corporation, 2788 San Tomas Expressway, Santa Clara, CA 95051, USA

[2]Department of Chemical and Biological Engineering, Colorado School of Mines, Golden, Colorado 80401, USA

[3]Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany

[4]Department of Chemistry, Chicago Center for Theoretical Chemistry, James Franck Institute, and Institute for Biophysical Dynamics, The University of Chicago, Chicago, Illinois 60637, USA

[a)] Author to whom correspondence should be addressed: gavoth@uchicago.edu

Dynamic Programming Algorithm for Essential Dynamics Coarse-Graining (EDCG)

The goal of EDCG is to minimize the residual from mapping ***n*** C_α atoms of a protein into ***N*** CG sites, which is defined as

$$\chi = \frac{1}{3N}\sum_{I=1}^{N}\mathbf{C}_I$$

where

$$\mathbf{C}_I = \sum_{i\in I}\sum_{j\geq i\in I}\langle\left|\Delta\boldsymbol{r}_i^{ED} - \Delta\boldsymbol{r}_j^{ED}\right|^2\rangle$$

is the loss of total covariance from mapping a group of C_α atoms to the CG site ***I***. In a sequential/linear model, C_α atoms associated with each CG site are assumed contiguous in the protein primary sequence, and a loss function for mapping a group of consecutive C_α atoms $\{a, a+1, a+2 \ldots b\}$ into a CG site can be rewritten as

$$\mathbf{C}(a,b) = \sum_{i=a}^{b-1}\sum_{j=i+1}^{b}\langle\left|\Delta\boldsymbol{r}_i^{ED} - \Delta\boldsymbol{r}_j^{ED}\right|^2\rangle$$

Therefore, to calculate the loss function for all possible values of $1 \leq a < b \leq n$, the time complexity is $O(N^4)$. To gain a higher computational efficiency, the loss function can be calculated from the recurrence equation:

$$\mathbf{C}(a,b) = \begin{cases} \mathbf{C}(a,b-1) + \mathbf{C}(a+1,b) - \mathbf{C}(a+1,b-1) + \langle |\Delta \boldsymbol{r}_a^{ED} - \Delta \boldsymbol{r}_b^{ED}|^2 \rangle, & a < b \\ 0, & a = b \end{cases}$$

The pseudo code with a time complexity of $O(N^2)$ can be designed as

```
loop L from 1 to n # number of Cα atoms to be mapped
    loop a from 1 to n-L # starting Cα atom
        b = a + L – 1 # ending Cα atom
        Calculate C(a,b) # calculate the loss
```

After obtaining all values of the loss functions, we can then define a sub-residual, $\chi(k,m)$, which is the minimum of total loss by mapping the first $\boldsymbol{k}$ C$_\alpha$ atoms into $\boldsymbol{m}$ CG sites, where $1 \leq k \leq n$, and $1 \leq m \leq N$ and $m \leq k$. A recurrence equation can then be defined as

$$\chi(k,m) = \begin{cases} \min\limits_{\mathbf{1 \leq \boldsymbol{i} \leq \boldsymbol{k}-\boldsymbol{m}+1}} \{\chi(k-i,m-1) + C(k-i+1,k)\}, & m > 1 \\ C(1,k), & m = 1 \end{cases}$$

The pseudo code can be designed as

```
Chi(k,1) = C(1,k)
loop m from 1 to N
    loop k from 1 to n
        Chi(k,m) = min{Chi(k-i,m-1)+C(k-i+1,k)}
```

The time complexity is $O(N^3)$ for calculating all sub-loss functions and the final residual $\chi(n,N)$ is the global minimum. This algorithm, known as dynamic programming, can be used to obtain the globally optimized solution.