CHARMM: The Biomolecular Simulation Program

B. R. BROOKS,[1*] C. L. BROOKS III,[2,3*] A. D. MACKERELL, Jr.,[4*] L. NILSSON,[5*] R. J. PETRELLA,[6,7*] B. ROUX,[8*] Y. WON,[9*] G. ARCHONTIS, C. BARTELS, S. BORESCH, A. CAFLISCH, L. CAVES, Q. CUI, A. R. DINNER, M. FEIG, S. FISCHER, J. GAO, M. HODOSCEK, W. IM, K. KUCZERA, T. LAZARIDIS, J. MA, V. OVCHINNIKOV, E. PACI, R. W. PASTOR, C. B. POST, J. Z. PU, M. SCHAEFER, B. TIDOR, R. M. VENABLE, H. L. WOODCOCK, X. WU, W. YANG, D. M. YORK, M. KARPLUS[6,10*]

[1]*Laboratory of Computational Biology, National Heart, Lung, and Blood Institute, National Institutes of Health, Bethesda, Maryland 20892*

[2]*Department of Chemistry, University of Michigan, Ann Arbor, Michigan 48109*

[3]*Department of Biophysics, University of Michigan, Ann Arbor, Michigan 48109*

[4]*Department of Pharmaceutical Sciences, School of Pharmacy, University of Maryland, Baltimore, Maryland 21201*

[5]*Department of Biosciences and Nutrition, Karolinska Institutet, SE-141 57, Huddinge, Sweden*

[6]*Department of Chemistry and Chemical Biology, Harvard University, Cambridge, Massachusetts 02138*

[7]*Department of Medicine, Harvard Medical School, Boston, Massachusetts 02115*

[8]*Department of Biochemistry and Molecular Biology, University of Chicago, Gordon Center for Integrative Science, Chicago, Illinois 60637*

[9]*Department of Chemistry, Hanyang University, Seoul 133-792, Korea*

[10]*Laboratoire de Chimie Biophysique, ISIS, Université de Strasbourg, 67000 Strasbourg, France*

Received 12 September 2008; Revised 24 February 2009; Accepted 3 March 2009
DOI 10.1002/jcc.21287
Published online 14 May 2009 in Wiley InterScience (www.interscience.wiley.com).

Abstract: CHARMM (Chemistry at HARvard Molecular Mechanics) is a highly versatile and widely used molecular simulation program. It has been developed over the last three decades with a primary focus on molecules of biological interest, including proteins, peptides, lipids, nucleic acids, carbohydrates, and small molecule ligands, as they occur in solution, crystals, and membrane environments. For the study of such systems, the program provides a large suite of computational tools that include numerous conformational and path sampling methods, free energy estimators, molecular minimization, dynamics, and analysis techniques, and model-building capabilities. The CHARMM program is applicable to problems involving a much broader class of many-particle systems. Calculations with CHARMM can be performed using a number of different energy functions and models, from mixed quantum mechanical-molecular mechanical force fields, to all-atom classical potential energy functions with explicit solvent and various boundary conditions, to implicit solvent and membrane models. The program has been ported to numerous platforms in both serial and parallel architectures. This article provides an overview of the program as it exists today with an emphasis on developments since the publication of the original CHARMM article in 1983.

 J Comput Chem 30: 1545–1614, 2009

Key words: biomolecular simulation; CHARMM program; molecular mechanics; molecular dynamics; molecular modeling; biophysical computation; energy function

I. Introduction

Understanding how biological macromolecular systems (proteins, nucleic acids, lipid membranes, carbohydrates, and their complexes) function is a major objective of current research by computational chemists and biophysicists. The hypothesis underlying computational models of biological macromolecules is that the behavior of such systems can be described in terms of the basic physical principles governing the interactions and motions of

Additional Supporting Information may be found in the online version of this article.

Correspondence to: B. R. Brooks; e-mail: brbrooks@helix.nih.gov or C. L. Brooks III; e-mail: brookscl@umich.edu or A. D. MacKerell, Jr.; e-mail: alex@outerbanks.umaryland.edu or L. Nilsson; e-mail: Lennart.Nilsson@ki.se or R. J. Petrella; e-mail: petrella@fas.harvard.edu or B. Roux; e-mail: roux@uchicago.edu or Y. Won; e-mail: won@hanyang.ac.kr or M. Karplus; e-mail: marci@tammy.harvard.edu

Contract/grant sponsors: NSF, NIH, DOE, Accelrys, CNRS, NHLBI

their elementary atomic constituents. The models are, thus, rooted in the fundamental laws of physics and chemistry, including electrostatics, quantum mechanics and statistical mechanics. The challenge now is in the development and application of methods, based on such well-established principles, to shed light on the structure, function, and properties of often complex biomolecular systems. With the advent of computers, the scope of molecular dynamics (MD; see footnote for naming conventions)[†] and other simulation techniques has evolved from the study of simple hard-sphere models of liquids in the 1950s,[1] to that of models of more complex atomic and molecular liquids in the 1960s,[2,3] and to the study of proteins in the 1970s.[4] Biological macromolecular systems of increasing size and complexity, including nucleic acids, viruses, membrane proteins, and macromolecular assemblies, are now being investigated using these computational methods.

The power and usefulness of atomic models based on realistic microscopic interactions for investigating the properties of a wide variety of biomolecules, as well as other chemical systems, has been amply demonstrated. The methodology and applications have been described in numerous books[5–10] and reviews.[11–13] Studies of such systems have now reached a point where computational models often have an important role in the design and interpretation of experiments. Of particular interest is the possibility of employing molecular simulations to obtain information that is difficult to determine experimentally.[14,15] A dictionary definition of "simulation" is, in fact, "the examination of a problem, often not subject to direct experimentation," and it is this broad meaning that is intended here. Typical studies range from those concerned with the structures, energies, and vibrational frequencies of small molecules, through those dealing with Monte Carlo and MD simulations of pure liquids and solutions, to analyses of the conformational energies and fluctuations of large molecules in solution or in crystal environments.

As the field of biomolecular computation continues to evolve, it is essential to retain maximum flexibility and to have available a wide range of computational methods for the implementation of novel ideas in research and its applications. The need to have an integrated approach for the development and application of such computational biophysical methods has led to the introduction of a number of general-purpose programs, some of which are widely distributed in academic and commercial environments. Several[16–21] were described in a special 2005 issue of *Journal of Computational Chemistry* (*JCC*). One of the programs, CHARMM (Chemistry at HARvard Molecular Mechanics), was not included in that publication because an article was not prepared in time for the issue. CHARMM was first described in *JCC* in 1983,[22] although its earlier implementations had already been used to study biomolecules for a number of years.[23]

CHARMM is a general and flexible molecular simulation and modeling program that uses classical (empirical and semiempirical) and quantum mechanical (QM) (semiempirical or *ab initio*) energy functions for molecular systems of many different classes, sizes, and levels of heterogeneity and complexity. The original version of the program, although considerably smaller and more limited than CHARMM is at present, made it possible to build the system of interest, optimize the configuration using energy minimization techniques, perform a normal mode or MD simulation, and analyze the simulation results to determine structural, equilibrium, and dynamic properties. This version of CHARMM[22,24] was able to treat isolated molecules, molecules in solution, and molecules in crystalline solids. The information for computations on proteins, nucleic acids, prosthetic groups (e.g., heme groups), and substrates was available as part of the program. A large set of analysis facilities was provided, which included static structure and energy comparisons, time series, correlation functions and statistical properties of molecular dynamic trajectories, and interfaces to computer graphics programs. Over the years, CHARMM has been ported to many different machines and platforms, in both serial and parallel implementations of the code; and it has been made to run efficiently on many types of computer systems, from single processor PCs, Mac and Linux workstations, to machines based on vectorial or multicore processors, to distributed-memory clusters of Linux machines, and large, shared-memory supercomputer installations. Equally important, the structure of the program has provided a robust framework for incorporating new ideas and methodologies—many of which did not even exist when CHARMM was first designed and coded in the late 1970s. Some examples are implicit solvent representations, free energy perturbation methods, structure refinement based on X-ray or NMR data, transition path sampling, locally enhanced sampling with multiple copies, discretized Feynman path integral simulations, quantum mechanical/molecular mechanical (QM/MM) simulations, and the treatment of induced polarization. The ability of the basic framework of CHARMM to accommodate new methods without large-scale restructuring of the code is one of the major reasons for the continuing success of the program as a vehicle for the development of computational molecular biophysics.

The primary goal of this article is to provide an overview of CHARMM as it exists today, focusing on the developments of the program during the 25 years since the publication of the first article describing the CHARMM program in 1983.[22] In addition, the current article briefly reviews the origin of the program, its management, its distribution to a broad group of users, and future directions in its development. Some familiarity with the original CHARMM article is assumed. Although many details of CHARMM usage, such as input commands and options, are included, full documentation is available online at www.charmm.org, as well as with all distributions of the program. The present work also provides, *de facto*, a review of the current state of the art in computational molecular biophysics. Consequently, it

[†]Method abbreviations, e.g., MD for molecular dynamics and MEP for minimum energy path, and module names, e.g., PBEQ for the PB module, as well as preprocessor keywords (see Section XI.B.), are in *allcaps*. CHARMM commands, subcommands, or command options are in *italics* with the first four letters capitalized. (The parser in CHARMM uses only the first four letters of a command; however, it is case-insensitive.) The term "keyword" is reserved for preprocessor keywords, not command options. File and directory names are enclosed in quotation marks, e.g., "build" directory. The "module" designation refers to portions of CHARMM source code that form a modular functional unit, not necessarily a Fortran module.

should be of interest not only to the CHARMM user community, but also to scientists employing other programs.

II. Overview of the Program

The central motivation for creating and developing the molecular simulation program CHARMM is to provide an integrated environment that includes a wide range of tools for the theoretical investigation of complex macromolecular systems, with particular emphasis on those that are important in biology. To achieve this, the program is self-contained and has been designed to be versatile, extensible, portable, and efficient. CHARMM strikes a balance between general efficiency (the ability of the end user to easily set up, run, and analyze a project) and extensibility/versatility (the ability of the program to support new implementations and the use of many methods and approaches). This section provides an introduction to some general aspects of the CHARMM program and its use, including the essential elements of a typical CHARMM project. In what follows, detailed descriptions are given of most of the program's features.

II.A. Outline of a Generic CHARMM Project

A typical research project with CHARMM can be described in very general terms based on the information flow in the program, which is schematically illustrated in Figure 1. The user begins a project by first setting up the atomic model representing the system of interest (see also Section IX.A.). This consists of importing the "residue" topologies file (RTF) and force field parameters (PRM), generating the "protein" structure file (PSF), and assembling a complete configuration (coordinates) of all the atoms in the system; the quotes around "residue" and "protein" indicate that the same (historical) notation is used when the program is applied to molecules in general. For molecules and moieties that have been parameterized, such as proteins, nucleic acids, and lipids, standard CHARMM PRM and RTF files can be used, and the setup procedure is straightforward if most of the coordinates are known. For molecules not included in the standard libraries, CHARMM is designed to allow for the use of a virtually unlimited variety of additional molecular topologies and force field parameters. (The available force fields are discussed in Section III.) For calculations involving multiple copies of a structure, such as reaction path calculations in which the coordinates of the two end structures are derived from X-ray crystallographic data, consistency of atom labels is required across all of the copies, particularly for chemically equivalent atoms (e.g., Cδ1 and Cδ2 of Tyr). CHARMM provides a set of general tools for facilitating the setup and manipulation of the molecular system (e.g., coordinate transformations and the construction of missing coordinates; Sections IX.B. and C.) and for imposing a variety of constraints (Section V.B.) and restraints (Section III.F.) on the system, where appropriate; restraints allow changes in the property of interest with an energetic penalty, while constraints fix the property, usually to user-specified values. The user can specify a number of options for the calculation of nonbonded interactions and can choose to impose any of a number of boundary conditions on the system (Section IV). To carry out the calculations in an acceptable length of real time, the user must consider tradeoffs in accuracy/complexity *versus*

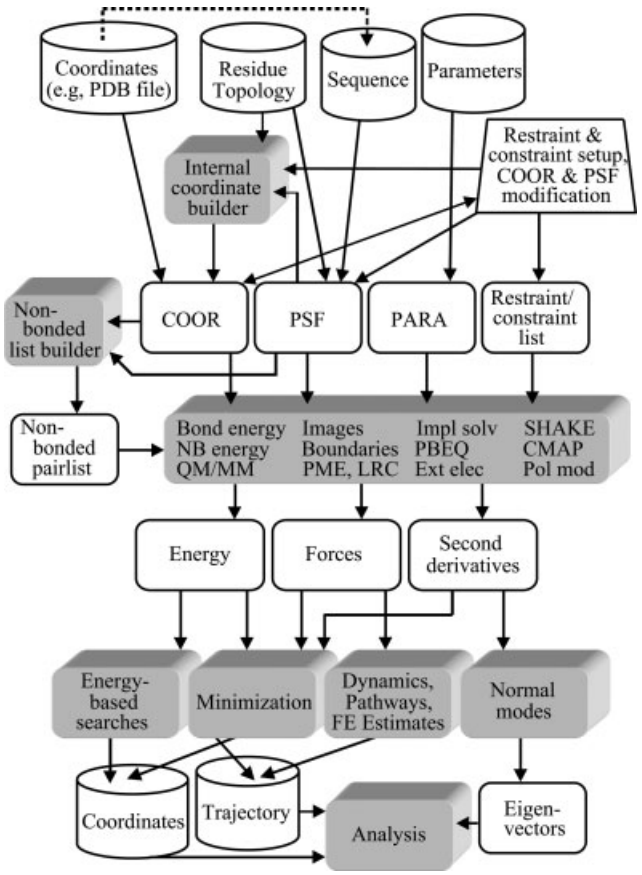

Figure 1. Diagram depicting the general scheme of the information flow in a CHARMM project. Information from data and parameter files (top row cylinders) and the input file (second row trapezoid) is first used to fill CHARMM data structures, which are then used by the energy routines and related modules (some of which are listed in the central grey box) to calculate the energy and its derivatives. This information is then used by various CHARMM modules for production calculations (second row from the bottom), which generate data in output files or internal data structures (bottom row) that are analyzed to obtain final results. Key: cylinders: data files; trapezoid: input file; white rectangles: data structures; shaded rectangles: CHARMM functionalities/modules; PDB: protein data bank; COOR, PSF, and PARA: internal CHARMM data structures for system coordinates, system topology/connectivity (PSF), and energy function parameters, respectively; NB energy: nonbonded energy; QM/MM: combined quantum mechanical/molecular mechanical methods; PME: Particle-Mesh Ewald summation method; LRC: long-range corrections for truncated van der Waals interactions; Impl solv: implicit solvation models; PBEQ: PB electrostatics module; Ext elec: Extended electrostatics; CMAP: backbone dihedral angle correction term for all-atom protein representation; Pol mod: polarizable models; Pathways: reaction pathway calculations; FE estimates: methods for estimating free energy differences.

efficiency (Section XII) when selecting the model to be employed in the calculations; in addition, he or she may need to use a parallel compilation of the code or to utilize time-saving features such as lookup tables (Section X). There are currently two Web-based interface utilities that can be used to facilitate the setup phase of a CHARMM project, CHARMM-GUI[25] and CHARMMing.[26]

The project may require a preproduction stage: e.g., for an MD simulation, the usual procedure is to minimize the system structure (often obtained from crystallographic or NMR data), to heat the system to the desired temperature, and then to equilibrate it. Once this is done, the project enters the production stage, during which the atomic conformation of the system may be refined, explored, and sampled by the application of various computational procedures. These procedures may consist, among other possibilities, of performing energy minimization, propagating MD or Langevin dynamics trajectories, sampling with Metropolis Monte Carlo or grid-based search algorithms, obtaining thermodynamic free energy differences via free energy perturbation computations, performing transition path sampling, or calculating normal modes of vibrations. With such methodologies, it is possible to simulate the time evolution of the molecular system, optimize, and generate conformations according to various statistical mechanical ensembles, characterize collective motions, and explore the energy landscape along particular reaction pathways. Some computational techniques (e.g., so-called "alchemical" free energy simulations) include the consideration of "unphysical" intermediate states to improve the calculation of physical observables, including the free energy, entropy, and enthalpy change due to a mutation or conformational transition. These algorithms and methods, which are central to many theoretical studies of biological macromolecules and other mesoscopic systems, are discussed in Sections V, VI, and VII. Although several key quantities are normally monitored during the production stage of a project, additional system properties may have to be determined by postprocessing the data—e.g., to calculate free energy changes from the coordinates or diffusion coefficients from the velocities saved during one or more MD trajectories. These derived quantities, whose calculation is described in Section VIII, may include time series, correlation functions, or other properties related to experimental observables. Finally, the advanced CHARMM user in some cases will have extended the program's functionality in the course of carrying out his project, either by creating CHARMM scripts (Section II.C.), writing external code as an adjunct, utilizing internal "hooks" to the CHARMM source code (Section IX.A.), or directly modifying one or more source code modules. After such developmental code has been made to conform to CHARMM coding standards and tested, it should be submitted to the CHARMM manager so as to be considered for inclusion in future distributions of the program (Section XI).

II.B. Functional Multiplicity of CHARMM

An important feature of CHARMM is that many specific computational tasks (e.g., the calculation of a free energy or the determination of a reaction pathway) can be accomplished in more than one way. This diversity has two major functions. First, the best method to use often depends on the specific nature of the problem being studied. Second, within a given type of problem or method, the level of approximation that achieves the best balance between accuracy requirements and computational resources often depends on the system size and complexity. A typical example arises in the class of models that are used to represent the effect of the surrounding solvent on a macromolecule. The most realistic representation treats the solvent environment by explicitly including the water molecules (as well as any counter ions, crystal neighbors, or membrane lipids, if they are present), and imposing periodic boundary conditions (PBC), which mimic an infinite system by reproducing the central cell[7,8] (see section IV.B.). Systems varying from tens to even hundreds of thousands of particles can be simulated with such all-explicit-atom models for hundreds of nanoseconds using currently available computational resources, such as large, distributed memory clusters of nodes and parallel program architectures. However, a drawback of treating solvated systems in this way is that most of the computing time (often more than 90%) is used for simulating the solvent rather than the parts of the system of primary interest. Consequently, an alternative approach is often used in which the influence of the solvent is incorporated implicitly with an effective mean-field potential (i.e., without the inclusion of actual water molecules in the calculation). This approach can greatly reduce the computational cost of a calculation for a protein relative to the use of explicit solvent, often by a 100-fold or more, and captures many of the equilibrium properties of the solvent. However, it introduces approximations, so that hydrodynamic and frictional solvent effects, as well as the role of water structure, are usually not accounted for in the implicit solvent approach. A variety of implicit solvent models, with differing accuracy and efficiency profiles, are available in CHARMM; a detailed discussion can be found in Section III.D. An intermediate approach between all-atom PBC simulations and implicit solvent models involves simulating only a small region explicitly in the presence of a reduced number of explicit solvent molecules, while applying an effective solvent boundary potential (SBP) to mimic the average influence of the surrounding solvent.[27–29] The SBP approach is often advantageous in simulations requiring an explicit, atomic representation of water in a limited region of the system—e.g., in the study of a reaction taking place in the active site of a large enzyme.[30] The choice of solvent representation for a project thus depends on several factors, including the accuracy requirements of the calculation, the type of data being sought, the system size, and the computational resources and (real) time available.

II.C. The CHARMM Scripting Language

Although CHARMM can be run interactively, as is often done when the CHARMM graphics facility (GRAPHX) is being used, intensive computational projects are normally executed in batch mode through the use of input files (see Fig. 2). A set of command structures, including *GOTO*, *STREam*, and *IF-ELSE-ENDIf* structures, corresponding to the respective control-flow statements in source code, provide the basis for a powerful high-level

```
* CHARMM Example input file. 10ps dynamics of BPTI in vacuum
*

! SETUP: Get topology, parameter and sequence data
READ rtf card name top_all22_prot.inp
READ param card name par_all22_prot.inp
READ sequence pdb name bpti.pdb

! generate the PSF, with 3 disulfides, and get coordinates
GENErate bpti
PATCh disu bpti  5  bpti 55
PATCh disu bpti 14  bpti 38
PATCh disu bpti 30  bpti 51
READ coor pdb name bpti.pdb

! add coordinates for hydrogen atoms
HBUIld select hydrogen end

!SIMULATE: Run dynamics with SHAKE constraints on all bonds
shake bonds
OPEN unit 11 write unformatted name bpti.trj
DYNAmics start leapfrog nstep 10000 timestep 0.001 -
firsttemp 35.0 finaltemp 285.0 teminc 50.0 ihtfrq 200 -
cutnb 14.0 ctofnb 12.0 fshift inbfrq -1 nsavc 100 iuncrd 11

! ANALYZE: Compute average coordinates and rms fluctuations
! from the coordinate trajectory file
OPEN unit 11 read unformatted name bpti.trj
COOR dyna firstunit 11 nunits 1
WRITe coor pdb name bpti-ave.pdb
```

Figure 2. CHARMM input file for an MD simulation of BPTI and a simple analysis of the resulting trajectory. This is similar in form to that used in the first MD simulation of a protein.[4] The example uses the CHARMM22 all-hydrogen force field, with topology descriptions for standard amino acids, and the interaction parameters in the text files "top_all22_prot.inp" and "par_all22_prot.inp," respectively. A PDB file is used to provide the amino acid sequence and the atomic coordinates; depending on the source of the PDB file, some manual editing may be required. Coordinates for hydrogen atoms are constructed using the HBUILD algorithm, SHAKE constraints are applied to all bonds, and the dynamics run is started at 35 K with heating in 50 K increments at 0.2 ps intervals to a final temperature of 285 K. Specifications for the calculation of nonbonded interactions are also given on the dynamics command line. Coordinates are saved every 100 steps to a binary file, which is reopened after the simulation and used to compute the average structure and RMS fluctuations. Other examples can be found at www.charmm.org.

scripting language that permits the general and flexible control of complicated simulation protocols and facilitates the prototyping of new methods. The various functionalities of CHARMM can easily be combined in almost any way using these command structures in scripts to satisfy the requirements of a particular project. In general, the order of CHARMM commands is limited only by the data required by the command. For example, the energy cannot be calculated unless the arrays holding the coordinates, parameters, and structural topology, etc., have already been filled (see Fig. 1). The command parser allows the substitution of numerous variables, which are set either internally by the program during execution (for example, the current number of atoms is accessible as "?natom"), or externally by the user (for example, a user may initially issue the command "*SET* temperature 298.15," and then substitute its value as "@temperature" on any command line in the CHARMM input script). All components of the most recent energy evaluation, as well as the results of many other calculations, are available as internal CHARMM variables (?identifier). The numerical values for the variables can then be written to an external file, further processed, or used in control statements ("*IF* ?ener.lt. −500 *THEN*..."). Arrays of these variables can also be constructed (e.g., "segid1," "segid2," ..., "segid10") and referenced (@segid@@j). The parser has a robust interpreter of arithmetic expressions (*CALC*), which can be used to evaluate algebraic functions of these variables using basic mathematical operations, including random number generation. Variable values may also be passed to the program at the start of execution. In addition, it is possible to call other CHARMM scripts as subroutines (*STREam ... RETUrn*), and to access operating system commands (*SYSTem*); depending on the operating system, CHARMM can use environment variables in filenames. In addition, the *SCALar* command facility performs arithmetic and statistical manipulations on internal CHARMM vectors (e.g., coordinates, forces, charges, masses, user-defined arrays). CHARMM variables and arrays can be read from (*GET, SCALar READ*) or written to (*ECHO, WRITe TITLe, SCALar WRITe*) external files, with or without header information, allowing, for example, easy access from external graphing programs. The extent of printing can be controlled with the *PRNLevel* and *WRNLevel* commands, which take integers in the range of −10 (print no messages or warnings) to +11 (print all). In general, values larger than 5 (default) will result in output that is not needed for production calculations but may be useful for debugging and script-checking purposes. For example, *PRNLevel* 8 will print the name of every energy-based subroutine as it is called.

Since CHARMM input files can take the form of miniprograms written in the interpretive language of CHARMM commands, common tasks can be coded in a general way at the script level. As examples, standard input scripts have been written for the addition of explicit solvent to a system, and a series of scripts has been developed that automates the setup of the initial configuration for a membrane–protein MD simulation (see Fig. 3).[31–33] It is also possible to implement complex methods and simulation protocols at the level of the input file without changing the source code. For example, the Random Expulsion method[34] has been implemented in this way in a study of ligand escape from a nuclear receptor[35] (see Fig. 4); see also Blondel et al.[36] Another example is the development and parameterization of a coarse-grained model of an amphipathic polypeptide which was used to investigate the kinetics of amyloid aggregation.[37] The flexibility of the scripting language is such that one could implement Metropolis Monte Carlo sampling in a few lines directly from the input files (though this would run less efficiently than the dedicated MC module). In addition, the scripting language is used extensively when performing the calculations required for the optimization of force field parameters (see next section).

III. Atomic Potential Energy Function

The relationship between structure and energy is an essential element of many computational studies based on detailed atomic models. The potential energy function, by custom called a force

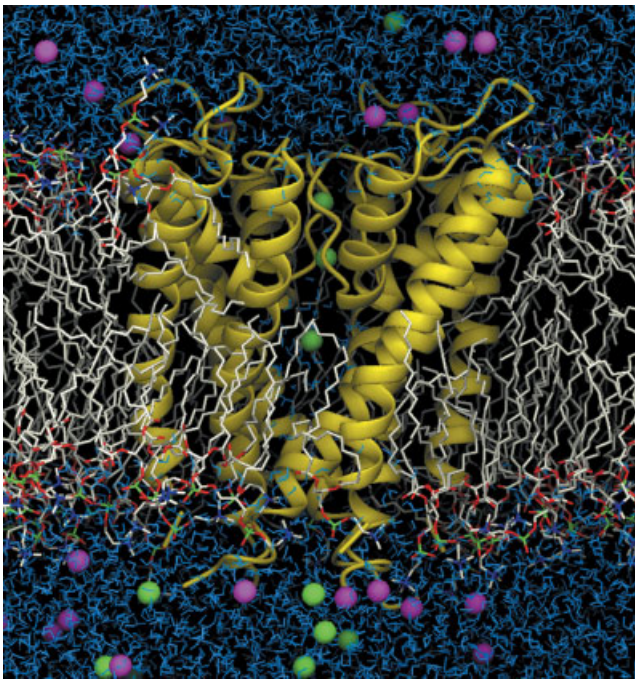

Figure 3. The KcsA K+ channel (helical ribbons) embedded in an explicit dipalmitoyl phosphatidylcholine (DPPC) phospholipid membrane (stick figures; fatty acids are white and head groups are red, green, and white) bathed by a 150 mM KCl aqueous salt solution (blue and green spheres represent potassium and chloride ions, respectively, and water molecules outside the membrane are shown in blue). The simulation system, consisting of 40,000 atoms, was used to compute a multi-ion PMF governing ion conduction[33] through the channel and to determine the sources of its ionic selectivity[723] (from Bernèche and Roux[33]).

field, is used to calculate the potential energy of the system and its derivatives from the coordinates corresponding to the structure or conformation. It has two aspects: the mathematical form and the empirical parameters. In CHARMM, the topology (RTF) and parameter (PRM) files (see Fig. 1), along with the polymer sequence, allow the potential energy function to be fully defined. First derivatives of the potential energy are used to determine the atomic forces, which are required for MD simulation and energy minimization. Second derivatives of the potential energy, which are required for the calculation of vibrational spectra and for some energy minimization algorithms, are also available. In a program like CHARMM, which is undergoing continuous development, changes in the force field and the rest of the code are often linked and developments in both made in concert.

Because force fields are approximations to the exact potential energy, they are expected to improve over time. The goals of force field development involve at least three factors; they are accuracy, breadth, and speed. Accuracy can be defined as the extent to which calculations using a force field can reproduce experimental observables. Breadth refers to the range of moieties, molecules, and systems to which a force field can be applied at the required level of accuracy. Speed is the relative efficiency of calculations using one force field over another, all else being equal; this often depends largely on the level of detail of the models, although the form of implementation can also have a role. In addition, the introduction of improvements to a *given* force field must be balanced by the need for stability of the force field (i.e. constancy of the form and parameters) over time. This is particularly true of accuracy gains: while improved accuracy in a given force field may be desired, continual change would make comparison of results from different versions of the force field problematic. In CHARMM, there have been continual force field developments over the years, many of which are discussed, including the development of force fields based on more detailed atomic representations (e.g., all atom, polarizable) and applicability to more molecular types (e.g. DNA, carbohydrates, lipids). At the same time, an effort has been made not to change validated and well-tested force fields, thereby facilitating comparison of results from studies performed at different times and in different laboratories. Notably, the only modification to the protein part of the all-atom fixed-point-charge CHARMM force field[38] since May 1993 has been the addition of a dihedral correction term (see Section III.C. later, CMAP); the nucleic acid part of this force field[39–41] has remained unchanged since 1998.

III.A. Molecular Mechanics Force Fields

The general form of the potential energy function most commonly used in CHARMM for macromolecular simulations is based on fixed point charges and is shown in eq. (1) (see also Brooks et al.[22] and Section IX.A.).

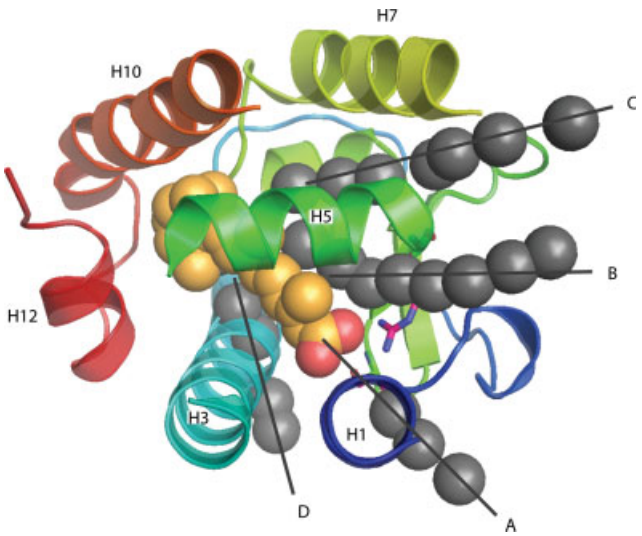

Figure 4. Four different (A–D) ligand escape pathways (shown as grey spheres along black guiding lines) identified using Random Acceleration Molecular Dynamics[35] in the ligand binding domain of the retinoic acid receptor. Helices are shown as ribbons, and the retinoic acid ligand in the bound initial state is shown as red and gold spheres (from Carlsson et al.[35]).

$$U(\vec{R}) = \sum_{\text{bonds}} K_b(b - b_0)^2 + \sum_{\text{angles}} K_\theta(\theta - \theta_0)^2 + \sum_{\text{Urey-Bradley}} K_{UB}(S - S_0)^2 + \sum_{\text{dihedrals}} K_\varphi(1 + \cos(n\varphi - \delta)) + \sum_{\text{impropers}} K_\omega(\omega - \omega_0)^2 + \sum_{\substack{\text{non-bonded}\\ \text{pairs}}} \left\{ \varepsilon_{ij}^{\min} \left[\left(\frac{R_{ij}^{\min}}{r_{ij}} \right)^{12} - 2 \left(\frac{R_{ij}^{\min}}{r_{ij}} \right)^{6} \right] + \frac{q_i q_j}{4\pi\varepsilon_0 \varepsilon r_{ij}} \right\} + \sum_{\text{residues}} U_{\text{CMAP}}(\varphi, \psi) \quad (1)$$

The potential energy, ($U(\vec{R})$), is a sum over individual terms representing the internal and nonbonded contributions as a function of the atomic coordinates. Internal terms include bond (b), valence angle (θ), Urey–Bradley (UB,S), dihedral angle (φ), improper angle (ω), and backbone torsional correction (CMAP, φ, ψ) contributions, as shown in eq. (1). The parameters K_b, K_φ, K_{UB}, K_θ, and K_ω are the respective force constants and the variables with the subscript 0 are the respective equilibrium values. All the internal terms are taken to be harmonic, except the dihedral angle term, which is a sinusoidal expression; here n is the multiplicity or periodicity of the dihedral angle and δ is the phase shift. The all-atom implementations of the CHARMM force field include all possible valence and dihedral angles for bonded atoms, and the dihedral angle term about a given bond may be expanded in a Fourier series of up to six terms. Most commonly, one dihedral angle term is used, though two or more have been introduced in some cases. In addition, for the protein main chain, a numerical correction term, called CMAP, has been implemented (see later). For three bonded atoms A—B—C, the Urey–Bradley term is a quadratic function of the distance, S, between atoms A and C. The improper dihedral angle term is used at branchpoints; that is, for atoms A, B, and D bonded to a central atom, C, the term is a quadratic function of the (pseudo)-dihedral angle defined by A—B—C—D. Both the Urey–Bradley and improper dihedral terms are used to optimize the fit to vibrational spectra and out-of-plane motions. In the polar hydrogen models (models in which CH3, CH2, and CH groups are treated as single extended atoms; see later), the improper dihedral angle term is also required to prevent inversion of chirality (e.g., about the C_α atom in proteins). Although the improper dihedral term is used very generally in the CHARMM force fields, the Urey–Bradley term tends to be used only in special cases.

Nonbonded terms include Coulombic interactions between the point charges (q_i and q_j) and the Lennard–Jones (LJ) 6–12 term, which is used for the treatment of the core-core repulsion and the attractive van der Waals dispersion interaction. Nonbonded interactions are calculated between all atom pairs within a user-specified interatomic cutoff distance, except for covalently bonded atom pairs (1,2 interactions) and atom pairs separated by two covalent bonds (1,3 interactions). The relative dielectric constant, ε, is set to one in calculations with explicit solvent, corresponding to the permittivity of vacuum, ε_0. In addition, the electrostatic term can be scaled using other values for the dielectric constant or a distance-dependent dielectric; in the latter, the electrostatic term is inversely proportional to r_{ij}^2, the distance between the interacting atoms squared. Expressions for ε used for implicit solvent model calculations are discussed in Section III.D. CHARMM also contains an explicit hydrogen bonding term, which is not used in the current generation of CHARMM force fields, but remains as a supported energy term for the purposes of facilitating model development and hydrogen bonding analysis.[42] In the LJ term, the well depth is represented by $\varepsilon_{ij}^{\min}$, where i and j are the indices of the interacting atoms, r_{ij} is the interatomic distance, and $R_{ij}^{\min}$ is the distance at which the LJ term has its minimum. Typically, $\varepsilon_{ii}^{\min}$ and $R_i^{\min}$ are obtained for individual atom types and then combined to yield $\varepsilon_{ij}^{\min}$ and $R_{ij}^{\min}$ for the interacting atoms via a standard combination rule. In the current CHARMM force fields, the $\varepsilon_{ij}^{\min}$ values are obtained via the geometric mean ($\varepsilon_{ij}^{\min} = \sqrt{\varepsilon_{ii}^{\min}\varepsilon_{jj}^{\min}}$) and $R_{ij}^{\min}$ via the arithmetic mean, $R_{ij}^{\min} = (R_i^{\min} + R_j^{\min})/2$. Other LJ combining rules are also supported, e.g., $R_{ij}^{\min} = \sqrt{R_i^{\min}R_j^{\min}}$), allowing for the use of alternative force fields in CHARMM (see later). Separate LJ parameters and a scaling factor for electrostatics can be used for the nonbonded interactions between atoms separated by three covalent bonds (1,4 interactions). The Buckingham potential[43] has recently been added as an alternative to the simple LJ for treating the core repulsion. The Morse potential,[44] often used for bond-breaking, is also implemented.

The simple form for the potential energy used in eq. (1) represents a compromise between accuracy and speed. For biomolecules at or near room temperature, the harmonic representation is generally adequate, though approximate, and the same holds true for the use of the LJ potential for the van der Waals interactions. However, alternative force fields with additional correction terms are available in CHARMM (Section III.B.) and can be used to check the results obtained with eq. (1). The earliest force field in CHARMM was based on an extended-atom (united atom) model, in which no hydrogen atoms were included explicitly. The omitted hydrogens were treated instead as part of the atom to which they were bonded.[45,46] These "extended atom" force fields typically required the explicit hydrogen bonding term mentioned earlier. A significant advance beyond the early models was based on the finding that the distance and angle dependencies of hydrogen bonds could be treated accurately by the LJ and electrostatic terms alone if the so-called polar hydrogens (OH and NH) were treated explicitly.[47] This eliminated the need for the inclusion of explicit hydrogen bonding terms and led to the creation of PARAM19,[48] called "the polar hydrogen model" for simulations of proteins. This model, which was first developed in the mid 1980s[47] is still widely used, particularly in simulations of proteins with an implicit treatment of the solvent (Section III.D.).

All-atom representations are the basis of the present generation of CHARMM force fields and were designed for simulations with explicit solvent. In these force fields, an effort was made to optimize the parameters using model compounds representative of moieties comprised by the macromolecules.[49] Testing was done against a variety of experimentally determined structural and thermodynamic properties of model compounds and macromolecules, augmented by QM calculations. A balance

of polar interactions (e.g., hydrogen bonds) between protein–protein, protein–water, and water–water interactions was maintained in the parameterization. CHARMM uses a slightly modified form of the TIP3P water model,[50] which includes LJ parameters for the hydrogens as well as the oxygen.[48,51] The properties of the model are not significantly altered,[52–54] because the hydrogens (r_{min} = 0.2245 Å) are well inside the van der Waals spheres of the oxygens (r_{min} = 1.7682 Å, O—H bond length = 0.9572 Å). The modification was introduced to avoid singularities in the use of integral equations for representing the solvent[55]; it is not important for explicit-solvent MD simulations. Currently, the all-atom models in CHARMM include the CHARMM22 force field for proteins,[56] the CHARMM27 force field for nucleic acids,[39,41] and force fields for lipids.[57–59] A limited set of parameters for carbohydrates is available,[60] with a more extensive set under development[61] (Brady, J. W.; Pastor, R.W.; MacKerell, A.D., Jr.; work in progress).

These force fields have been designed to be compatible, allowing for studies of heterogeneous systems. The nucleic acid and lipid force fields are significant improvements over earlier all-atom models produced in the 1990s[62,63]; the gains were achieved through extensive testing with macromolecular simulations and improved QM benchmarks.[59] In addition, force field parameters are available for a variety of modified protein and nucleic acid moieties and prosthetic groups.[41,64,65] Moreover, a description of the appropriate methods for extending the CHARMM all-atom force fields to new molecules or moieties has been published,[49] and tools for carrying out this type of extension are available via the CHARMM Web page at http://www.charmm.org. The all-atom CHARMM force fields, with a few improvements described later, have been applied to many different systems and shown to be adequate for quantitative studies (e.g., free energy simulations). Separately, an extended version of the CHARMM all-atom force fields for the treatment of candidate drug-like molecules is currently under development. Combined with a flexible parameter reader and automated RTF generation, this "generalized" force field will be particularly useful for screening of drug candidates (Brooks, B. R.; MacKerell, A. D., Jr.; work in progress).

III.B. Additional Supported Force Fields

Access to multiple, highly optimized, and well-tested force fields for simulations of biological macromolecules is useful for assessing the robustness of the computational results. In addition to the force fields developed specifically for CHARMM, versions of the AMBER nucleic acid, and protein force fields,[66,67] the OPLS protein force fields[68] with the TIP3P or TIP4P water models,[50,69] and the nucleic acid force field from Bristol-Myers Squibb[70] have been integrated for use with other parts of the CHARMM program. The SPC,[71] SPC/E[72], and ST2[73] water models are also available. A recent comparison of simulations with the CHARMM22, AMBER, and OPLS force fields showed that the three models give good results that are similar for the structural properties of three proteins.[69] Since that study, the CHARMM force field has been improved by adding a spline-based 2D dihedral energy correction term (CMAP) for the protein backbone (see Section III.C.).[74] For the free energy of hydration of 15 amino acid side chain analogs, the CHARMM22, AMBER, and OPLS force fields yielded comparable deviations (of about 1 kcal/mol) from the experimental values.[75,76] A simulation of the conformational dynamics of the eight principal deoxyribo and ribonucleosides using long explicit-solvent simulations showed that the CHARMM27 force field yields a description in agreement with experiment and provides an especially accurate representation of the ribose moiety.[77] This study also details a comparison of simulations using the CHARMM27 and AMBER nucleic acid force fields, performed with CHARMM. A simulation study described by Reddy et al.[78] compares the different force fields available in CHARMM for B-DNA oligomers. In addition, CHARMM has been shown to yield quantitative agreement with NMR imino proton exchange experiments on base opening.[79–81]

CHARMM also includes the Merck Molecular Force Field (MMFF)[82,83] and the Consistent Force Field (CFF).[84,85] These force fields use so-called "Class II" potential energy functions that differ from that in eq. (1) by the addition of cross terms between different internal coordinates (e.g., terms that couple the bond lengths and angles) and alternative methods for the treatment of the nonbonded interactions. The CFF force field is based on the early force field of Lifson and Warshel.[86] The MMFF force field is specifically designed to be used within the CHARMM program for the study of a wide range of organic compounds of pharmaceutical interest. CHARMM is able to read PDB, MERCK, or MOL2 formatted files, including MOL2 databases, so as to support large-scale virtual drug screening. Also, a script is available that transforms the MMFF parameterization for a given molecule so as to be consistent with the standard CHARMM force field.

III.C. Recent Extensions and Current Developments

Improved Backbone Dihedral Angle Potential

An important advance for the accurate calculation of the internal energies of biomolecules is the introduction of a multidimensional spline fitting procedure.[74,87] It allows for any target energy surface associated with two dihedral angles to be added to the potential energy function in eq. (1). The use of the spline function, referred to as CMAP, corrects certain small systematic errors in the description of the protein backbone by the all-atom CHARMM force field. The CMAP correction, which is based on *ab initio* QM calculations, as well as structure-based potentials of mean force, significantly improves the structural and dynamic results obtained with MD simulations of proteins in crystalline and solution environments.[74,88] Additional simulations have shown improved agreement with N—H order parameters as measured by NMR.[89] The spline function is expected to be generally useful for improving the representation of the internal flexibility of biopolymers when the available data indicate that corrections are required.[90]

Treatment of Induced Polarization

A refinement in the fixed charge distribution of the standard CHARMM biomolecular force field is the incorporation of the

influence of induced electronic polarization. Polarization is expected to have particularly important effects on the structure, energetics, and dynamics of systems containing charged (e.g., metal ions) or highly polar species. There is also an indication that polarization effects can be significant in accurately modeling the nonpolar hydrocarbon core of lipid membranes.[91,92] Although the physics of polarization is well understood, there are problems associated with introducing it into biomolecular simulations. They concern the choice of a suitable mathematical representation, the design of efficient computational algorithms, and the reparameterization of the force field. The three most promising representations are the fluctuating charge model introduced by Rick and Berne,[93] which is based on the charge-equalization principle,[94] the classical Drude oscillator model (also called the Shell model),[95] and the induced point dipole model.[96–98] Patel and Brooks[99] have developed and tested a polarizable CHARMM force field for proteins based on a charge-equalization scheme (CHEQ module). It is currently being used in molecular simulations to explore the role of electronic polarizability in proteins and peptides in solution,[99,100] at phase boundaries in alcohols,[101,102] and alkanes,[103] and in the conductance of ion channels.[92] MacKerell, Roux and coworkers are exploring a polarizable model based on the classical Drude oscillator methods[104] and have developed the SWM4-DP polarizable water model,[105,106] which has been used to simulate DNA in solution.[107] A recent parameterization of alkanes,[108] alcohols,[109,110] aromatics,[111] ethers,[112] amides,[113] and small ions[114] demonstrates the ability of Drude oscillator-based polarizabilities to reproduce a set of experimental observables that are incorrectly modeled by force fields with fixed charges. Examples include the dielectric constants of neat alkanes,[108] water–ethanol mixtures with concentrations that vary over the full molar fraction range,[109,113] and liquid *N*-methylacetamide, as well as the excess concentration of large, polarizable anions found at the air–water interface.[115–118] Gao and coworkers have used polarizable intermolecular potential functions, PIPFs, that model electronic polarization with an induced point dipole approach to study polarization effects in a series of organic liquids including alkanes, alcohols, and amides[96,98,119]; the results obtained with the induced-dipole model were found to be in good accord with those obtained from combined QM/MM simulations in which polarization effects were introduced with QM calculations.

In all the three induced polarization methods, the polarization is modeled as additional dynamical degrees of freedom that are propagated according to extended Lagrangian algorithms. This treatment avoids the need to introduce computationally inefficient approaches based on iterative self-consistent field (SCF) methods.[104,120] Efforts are currently underway to obtain complete sets of protein, nucleic acid, and lipid parameters for these polarizable force fields.

The polarizable models described here represent ongoing combined code and parameter developments that will be incorporated into the next generation of CHARMM force fields. Once this has been accomplished, it will be possible to carry out additional comparative studies (i.e., simulations with and without polarization) to determine the types of problems for which the use of such polarizable force fields is important.

III.D. Implicit Solvent Methods

Although MD simulations in which a large number of solvent molecules are included provide the most detailed representation of a solvated biomolecular system (see later), incorporating the influence of the solvent implicitly via an effective mean-field potential can provide a cost-efficient alternative that is sufficiently accurate for solving many problems of interest. Although implicit solvent simulations have computational requirements (CPU and memory) that can be close to those for vacuum calculations, they avoid many of the artifacts present in the latter, such as large deviations from crystal structures, excessive numbers of salt bridges, and fluctuations that are too small relative to crystallographic B factors. The reduction in computer time obtained with implicit models, relative to the use of an explicit solvent environment, can be important for problems requiring extensive conformational searching, such as simulations of peptide and protein folding[121–123] and studies of the conformational changes in large assemblies.[122,124] Implicit solvent approaches allow the estimation of solvation free energies while avoiding the statistical errors associated with averages extracted from simulations with a large number of solvent molecules. Examples of this type of approach are the MM/GBSA or MM/PBSA approaches to approximate free energies,[125] pK_a calculations for ligands in a protein environment,[126–129] and scoring protein conformations in *ab initio* folding or homology modeling studies.[130–133] An implicit solvent also permits arbitrarily large atomic displacements of the solute without solvent clashes, leading to more efficient conformational sampling in Monte Carlo and grid-based algorithms. Recently developed implicit membrane models, by analogy with implicit water (or other solvent) models, facilitate the study of proteins embedded in membranes.[134–139] Implicit solvent representations are also useful as conceptual tools for analyzing the results of simulations generated with explicit solvent molecules and for better understanding the nature of solvation phenomena.[140,141] Finally, the instantaneous solvent relaxation that is inherent in implicit solvation models is useful for the study of macromolecular conformational changes over the "simulation-accessible" nanosecond or shorter timescales, as in forced unfolding MD simulations of proteins,[142] *versus* the experimental microsecond to millisecond timescales. Treating the solvent explicitly in this type of calculation can introduce artifacts because of possible coupling between the solvent relaxation, which occurs on the nanosecond timescale, and the sped-up conformational change.

Several implicit solvent approaches are available in CHARMM, which effectively extend the number of available force fields in the program. The implicit solvent models differ both in their theoretical framework (e.g., the surface area-based empirical solvation potentials *versus* the approximate continuum models based on generalized Born theory) and in their implementation. A comparison of five of the effective (implicit solvent) free energy surfaces for three peptides known to have stable conformations in solution is presented by Steinbach.[143] Good agreement between results obtained with implicit and explicit solvent has been observed for the potential of mean force (PMF) as a function of the end-to-end distance of a 12-residue peptide[144] and as a function of the radius of gyration of a six-resi-

due peptide.[145] The implicit solvent methods currently available in CHARMM are outlined below. A comparison of the speeds of several of the methods with vacuum and explicit solvent calculations is also presented.

Solvent-Accessible Surface Area Models

One of the earliest and simplest implicit solvent models implemented in CHARMM, and currently the fastest one in the program, is based on the solvent-accessible surface area (SASA).[146] Models of this kind make the assumption that the solvation free energy of each part of a molecule is proportional to its SASA—i.e., they approximate the contribution arising from solute interactions with the first solvation shell by use of a term that is a sum of all of these individual "self-energy" contributions. In the original formulation by Eisenberg and coworkers,[147,148] the solvation free energy term was expressed as $G_H = \sum H_i f_i + C_i$, where H_i is the hydrophobicity of an individual protein residue, f_i is the fraction of the residue's surface that is available to solvent, the C_i's are constants, and the sum is over all residues in the molecule. The method was subsequently refined by the introduction of atomic solvation parameters (ASPs), which are the atomic analogues of the H_i factors, and the solvation energy term was written as a sum over individual atomic contributions (without the constant terms).[147,148] This form of the SASA model has largely replaced the Wesson and Eisenberg formulation, although the latter is still available in CHARMM (along with a derivative form for membranes). The current CHARMM implementation of the SASA model[149] uses the polar hydrogen (PARAM19) potential energy, has two ASPs, calculates the SASA analytically[150] and includes approximate solvent shielding effects for the charges. One ASP value in the CHARMM SASA model is negative, favoring the direct solvation of polar groups, and the other is positive, approximating the hydrophobic effect on nonpolar groups.[149] The two parameters were optimized to be consistent with the simplified treatment of electrostatic interactions based on the neutralization of charged groups[151] and the use of distance-dependent dielectric screening (with $\varepsilon(r) = 2r$). The charge neutralization and distance-dependent dielectric address, in an approximate way, solvent shielding of the electrostatic interactions that is not accounted for in the simpler SASA-based solvation models. However, in the present approach the shielding does not depend on the environment (i.e., given the same interatomic distance, a pair of charges in the interior of a protein feels the same screening as a pair of charges at the protein surface) so that it is most accurate for peptides and small proteins, where most of the atoms are on or near the surface. The change in the SASA, as a function of the system coordinates, can be used to obtain forces for minimization and dynamics. In part because the surface area calculation is analytic and based on interatomic distances, the SASA model is fast and has been shown to be useful in computationally demanding problems, such as the analysis of interactions in icosahedral viral capsids.[152] The two-ASP SASA model has been used for investigating the folding mechanism of structured peptides[153–156] and small proteins,[157] as well as the reversible mechanical unfolding of a helical peptide.[158] Moreover, simulations of the early steps of aggregation of amyloid-forming peptides using the SASA model have provided evidence of the importance of side chain interactions[159,160] and elucidated the role of aggregation "hot-spots" along the polypeptide sequence.[161] Because of the efficiency of the two-ASP SASA model,[149] most of the studies mentioned involved simulations of several microseconds in length, which have yielded adequate sampling of the peptide systems at equilibrium. A SASA model based on the all-atom representation is also present in CHARMM as part of the RUSH module[162] (see CHARMM documentation).

Gaussian Solvation Free Energy Model (EEF1)

A related model, referred to as EEF1,[151] combines an excluded-volume implicit solvation model with a modified version of the polar hydrogen energy function (PARAM19 atomic representation). The model is similar in spirit to SASA/ASP but does not require the calculation of the SASA. In EEF1, as in the SASA/ASP model, the solvation free energy is considered to be the sum of contributions from the system's constituent elements. The solvation free energy of each group of atoms in the EEF1 model is equal to the solvation free energy that the same group has in a reference (model) compound, minus the solvation lost due to the presence of other protein groups around it (solvent exclusion effect). A Gaussian function is used to describe the decay of the solvation free energy density with distance. Group contributions to the solvation free energy were obtained from an analysis of experimental solvation free energy data for model compounds.[163,164] In addition to the solvent-exclusion effect, the dielectric screening of electrostatic interactions by water is accounted for by the use of a distance-dependent dielectric constant and the neutralization of ionic side chains; the latter is essential for the EEF1 model, and was also adopted in the two-ASP SASA model.[149,153] MD simulations with EEF1 are about 1.7 times slower than vacuum simulations but significantly faster than most of the other solvation models in CHARMM (see later). The model has been tested extensively. It yields modest deviations from crystal structures in MD simulations at room temperature and unfolding pathways that are in satisfactory agreement with explicit solvent simulations. The model has been used to discriminate native conformations from misfolded decoys[130] and to determine the folding free energy landscape of a β-hairpin.[165,166] Other studies include the exploration of partially unfolded states of α-lactalbumin,[167] a series of studies of protein unfolding,[142,168–170] the investigation of coupled unfolding/dissociation of the p53 tetramerization domain,[171] the identification of stable building blocks in proteins,[172] an analysis of the energy landscape of polyalanine,[173] an analysis of the heat capacity change on protein denaturation,[174] the packing of secondary structural elements of proteins into the correct tertiary structural folds,[175] and calculations of the contributions to protein–ligand binding free energies.[176] EEF1 has been used by Baker and coworkers in successful protein–protein docking[177] and protein design studies.[178] An implicit membrane model based on EEF1 is available in CHARMM.[135] An updated parameterization based on PMF calculations for ionizable side chains[179] is referred to as EEF1.1.[135] EEF1 has also been adapted for use with the all-atom CHARMM 22 energy function,[180] but this formulation has not yet been extensively tested.

Screened Coulomb Potentials Implicit Solvent Model (SCPISM)

The SCPISM continuum model uses a screened Coulomb potential to describe solvent-shielded interactions, based on the Debye theory of liquids.[181,182] In the SCPISM model, the standard electrostatic component of the force field (Coulomb interaction *in vacuo*) is replaced by terms that describe both the screened electrostatic interactions and the self-energy of each atom. Hydrogen bonding modulation[183] and nonelectrostatic solvent-induced forces (e.g., hydrophobicity) are included in the recent version. The current implementation in CHARMM can be used for energy evaluations, minimization, and MD simulations. It has recently been shown that the SCPISM model preserves the main structural properties of proteins (of up to 75 amino acids) in long (>35 ns) Langevin dynamics simulations, as well as hydrogen bond patterns of residues at the protein/solvent interface.[88] For a 15,000-atom system, MD simulations with this method (using an all-atom model) are approximately five times slower than with EEF1 (which uses a polar hydrogen model representation).

Implicit Solvent with Reference Integral Site Model (RISM)

The RISM module in CHARMM implements the reference interaction site model.[184] This is based on an approximate statistical mechanical theory that involves the site–site Ornstein–Zernike integral equation and makes possible the calculation of the average solvent radial pair correlation function around a molecular solute. The calculated site–site radial distribution functions $g(r)$ and pair correlation functions $c(r)$ can then be used to determine quantities such as the PMF between two solvated molecules, and the excess chemical potential of solvation of a solute in a solvent. The method was first used to characterize the effect of solvent on the flexibility of alanine dipeptide.[55] The change in the solvent $g(r)$ on solvation can be determined, which allows for the decomposition of the excess chemical potential into the energy and entropy of solvation.[185] Further development would be required for the application of the method to larger peptides and small proteins, which is now feasible given the availability of fast computers.[186]

Poisson–Boltzmann (PB) Continuum Electrostatics

The PB equation provides the basis for the most accurate continuum models of solvation effects on electrostatic interactions. Thus PB models are used as the standards for other continuum models, but have the drawback that they are computationally intensive, though still less costly than the use of explicit solvent. The linearized PB equation for macroscopic continuum media has the form:

$$\nabla \cdot (\varepsilon(\mathbf{r})(\nabla\phi(\mathbf{r}))) - \kappa(\mathbf{r})^2\phi(\mathbf{r}) = -4\pi\rho(\mathbf{r}) \tag{2}$$

where ϕ is the electrostatic potential and ε, κ and ρ are the spatially varying dielectric constant, ionic screening, and atomic charge density, respectively. This formulation is based on the assumption that, at a given position in space, the polarization density of the solvent and the local cationic and anionic densities are linearly proportional to the local electric field and local electrostatic potential, respectively. At physiologic ionic strength and lower charge densities, the linear and nonlinear forms of the PB equation give equivalent results[187]; use of the nonlinear form, which is more computationally costly, is recommended in cases where the charge density is too high for the linear approximation to hold. This can be true at low ionic strength for nucleic acid systems. In the CHARMM program (PBEQ module), the PB equation is solved numerically using an iterative finite-difference relaxation algorithm[188,189] by mapping the system (i.e., ε, κ, and ρ) onto a discrete spatial grid. The PBEQ module can handle the linear and nonlinear forms of the PB equation, as well as a partially linearized form inspired by the 3D-PLHNC closure of Kovalenko and Hirata.[190] For the linear PB model, the electrostatic solvation free energy is calculated as

$$\Delta G_{\text{elec}} = \frac{1}{2}\sum_i q_i \phi_{\text{rf}}(i), \tag{3}$$

where q_i is the charge on particle i and $\phi_{\text{rf}}(i)$ is the reaction field at the position of particle i (usually obtained by subtracting the electrostatic potential in vacuum from that calculated with the dielectric solvent environment). This can also be expressed as[191]

$$\Delta G_{\text{elec}} = \frac{1}{2}\sum_{i,j} q_i M_{\text{rf}}(i,j) q_j, \tag{4}$$

where $M_{\text{rf}}(i,j)$ is the reaction field Green function matrix. The PBEQ module in CHARMM[191,192] computes the electrostatic potential and the solvation free energy using this approach. The accuracy of continuum electrostatic models is sensitive to the choice of the atomic radii used for setting the dielectric boundary between the solute and the solvent. For accurate PB calculations with the PBEQ module, optimized sets of atomic protein and nucleic acid Born-like radii have been determined using MD simulations and free energy perturbation calculations with explicit water molecules.[192,193] Continuum electrostatic calculations with the optimized atomic radii provide an implicit solvent approach that is generally useful; examples are the studies of nucleic acids and their complexes with proteins[194,195] and of MM/PBSA calculations on kinase inhibitor affinities.[196] The PBEQ module also has a number of features that can be used in electrostatic calculations related to biological membranes.[32,197] In particular, it can be employed to calculate the transmembrane potential profile and the induced capacitive surface charge corresponding to a given transmembrane potential difference, which is essential for examining conformational changes driven by an electrostatic voltage difference across the membrane.[197,198]

In addition to the standard Dirichlet boundary conditions (fixed potential on the edge of the grid), a number of options for imposing alternative boundary conditions on the edge of the finite grid are available; they include conducting boundary conditions (zero electrostatic potential), periodic boundary conditions in three dimensions, and planar periodic boundary conditions in two dimensions. The latter are useful for calculations involving planar membranes. The average electrostatic potential over user-specified parts of the system can also be calculated (*PBAVerage* subcommand); this is used, for example, in charge-scaling procedures. It is also possible to use the result from a coarse grid to set

up the boundary conditions of a finer grid, focusing on a small region of interest. The PBEQ module is not limited to the most common applications of the finite-difference PB equation, which involve determining the effective solvation of a solute in a given conformation. An accurate method for calculating the analytic first derivative of the finite-difference PB solvation free energy with respect to the atomic coordinates of the solute (electrostatic solvation forces) has also been implemented.[191] It allows the PBEQ module to be used in combination with several of the other tools available in CHARMM for investigating the properties of biological macromolecules (i.e., energy minimization, MD, reaction path optimization, normal modes, etc.). Since the PB calculation treats the effect of solvent only on the electrostatic interactions, it is often combined with methods for estimating the hydrophobic contribution. The simplest one approximates the term as proportional to the SASA, but in recent years more sophisticated approaches have been developed. For example, AGBNP in the Impact program[199] and PBSA in Amber[200] account for both cavity and solute–solvent dispersion interactions.

Smooth "Conductor-Like Screening Model" (COSMO) Solvation Model

Solvation boundary element methods based on the COSMO[201] model have proved to be stable and efficient. This model relies on an electrostatic variational principle that is exact for a conductor, and with certain corrections, provides useful, approximate results for many solvents over a broad range of dielectric constants.[202–204]

For such a model, the solvent reaction field potential can be represented as the potential arising from a surface charge distribution that lies at the dielectric boundary. This allows study of a two-dimensional surface problem instead of a three-dimensional volume problem. An advantage is that it is often easier to refine the discretization of the two-dimensional boundary element surface than to increase the resolution of a three-dimensional grid in a finite-difference PB calculation. In the COSMO approach, the numerical solution of the variational problem involves the discretization of the cavity surface into tesserae that are used to expand the solvent polarization density from which the reaction field potential is derived. A difficulty that can arise in the surface discretization used in these methods involves ensuring continuity of the solvation energy and its derivatives with respect to the atomic coordinates, which is critical for stable molecular mechanics optimization procedures and dynamics simulations. The smooth COSMO method developed by York and Karplus[205] addresses this problem and provides a stable and efficient boundary element method solvation model that can be used in a variety of applications. The method utilizes Gaussian surface elements to avoid singularities in the surface element interaction matrix, and a switching function that allows surface elements to smoothly appear or disappear as atoms become exposed or buried. The energy surface in this formulation has been demonstrated to have smooth analytic derivatives, and the method has been recently integrated into the semiempirical MNDO97[206] program interfaced with CHARMM.[207,208]

The smooth COSMO method, like the COSMO method, has some computational advantages (in both speed and memory requirements) over the PB method that arise from the discretization procedure. The convergence of the numerical solution in all three of the methods depends on the resolution of the grids, and in the case of the COSMO methods, the lower dimensionality of the grid used to discretize the numerical problem leads generally to increased computational efficiency and lower demands on computer memory. However, the COSMO methods are less general than the PB method in that the latter can treat spatially varying dielectric constants and effects of ion concentration in a more straightforward manner.

Generalized Born Electrostatics

Implicit solvent models based on the generalized Born (GB) formalism share the same underlying dielectric continuum model for the solvent as the Poisson or PB methods. However, GB theories replace the time-consuming iterative solution for obtaining the electrostatic potential required in finite-difference PB calculations in eq. (2) by the solvent-induced reaction field energy as approximated by a pairwise sum over interacting charges, q_i,[209–213]

$$\Delta G^{\text{elec}}_{\varepsilon_p->\varepsilon_w} = -\frac{1}{2}\left(\frac{1}{\varepsilon_p}-\frac{1}{\varepsilon_w}\right)\sum_{i,j}\frac{q_i q_j}{\sqrt{r_{ij}^2+\alpha_i\alpha_j\exp(-r_{ij}^2/F\alpha_i\alpha_j)}}. \quad (5)$$

In this expression ε_p, ε_w are the interior and exterior dielectric constants, r_{ij} is the distance between atoms i and j, and α_i is the effective Born radius of atom i, which is chosen to match the self-energy of charge i at its position in the system (i.e., α varies with the position of the atoms). The empirical factor F modulates the length-scale of the Gaussian term and typically ranges from 2 to 10, with 4 being the most commonly used value.[209] Equation (5) assumes that the shielded electrostatic interactions arising in the dielectric environment can be expressed as a superposition of pairwise terms. This is the so-called "pairwise shielding approximation". The efficiency of the GB approach lies in the possibility of estimating the effective atomic Born radii using a computationally inexpensive scheme. For example, the Coulomb field approximation assumes that the dielectric displacement for a set of charges embedded in a low dielectric cavity behaves like the Coulomb field of these charges in vacuum,[213,214] leading to the following expression for α_i

$$\frac{1}{\alpha_i}=\frac{1}{R_i}-\frac{1}{4\pi}\int_{\text{solute},r>R_i}\frac{1}{r^4}dV \quad (6)$$

where R_i is usually the atomic van der Waals radius of atom i. Many generalized Born theories approximate the volume integral, carried out over the entire solute cavity, by a discrete sum of overlapping spheres[211,212] or Gaussians.[213] Alternative methods have also been devised to carry out the integration, with moderate computational cost, either by reformulating the volume integral into a surface integral[215] or by directly using analytical integration techniques borrowed from density functional theory.[134,216,217]

Several implicit solvent schemes based on the pairwise shielding approximation exist in CHARMM. The first to be implemented in CHARMM was the Analytic Continuum Elec-

trostatics (ACE) model developed by Schaefer and Karplus.[213] This model is based on the Coulomb field approximation and the pairwise summation utilizing Gaussian functions as described earlier.[213] Applications of the model include MD simulations and studies of the folding of proteins and peptides.[121,218] An improved version of ACE, called ACE2, is now available and should be used in most applications with the PARAM19 polar hydrogen force field. Also implemented in CHARMM is a "standard" GB model following the formulation of Qiu et al.[211] This approach utilizes a pairwise sum over atoms to provide estimates of the atomic Born radii (solution to eq. 6 earlier).[219] It is optimized for use with the PARAM 19 polar hydrogen force field described earlier, with which it yields mean-absolute errors of 1–2% in the calculated solvation energies when compared with Poisson solutions using the same dielectric boundary. This model, accessed in CHARMM via the *GBORn* command (GENBORN preprocessor keyword), has been integrated with a number of other methods, such as free energy perturbation calculations and replicas. It has proven useful in folding studies of peptides and proteins,[220] the investigation of helix to coil transitions,[221] and binding free energy calculations.[222]

The description of the solvent boundary at the molecular surface in the ACE and standard GB methods can lead to problems that arise from the presence of microscopic, solvent-inaccessible voids of high dielectric in the interior of larger biomolecules. One approach used in PB calculations is to fill the voids with neutral spheres of low dielectric constant.[223] In an alternative approach, the integral formulation described by eq. (6) can be evaluated numerically with methods drawn from density functional theory.[216] This method can be extended with analytical approximations for the molecular volume or a van der Waals-based surface with a smooth switching function similar to that used by Im et al. in the context of the PB equation.[191] The molecular volume approximation is implemented in the Generalized Born/Molecular Volume (GBMV) model,[217] the smoothed van der Waals surface in the GBSW model.[134] These approaches provide results that are comparable to "exact" continuum Poisson theory.[224] However, they are considerably more time-consuming than the simpler models. The GBSW model is approximately five times as expensive as corresponding vacuum simulations, and the GBMV model is 6–10 times as expensive (see also next subsection). The GBMV and GBSW models have been applied to protein–ligand interactions,[225] protein–protein and protein–DNA interactions,[141] pH-coupled MD[127,129] and protein folding/scoring in structure prediction.[132] Key in improving the accuracy of these models have been extensions beyond the Coulomb field approximation described in eq. (6) earlier,[216,217] which is exact only for a single charge at the center of a spherical cavity.[226] The FACTS model (fast analytical continuum treatment of solvation) is a recently developed GB method in which the effective Born radius of each atom is estimated efficiently by using empirical formulas for approximating the volume and spatial symmetry of the solvent that is displaced by its neighboring atoms.[227] Apart from the factor F in eq. (5), the GB implementations in CHARMM involve empirical volume parameters for the calculation of the Born radii in eq. (6). The ACE model uses type-dependent atomic volumes derived by averaging over high-resolution structures in the PDB,[228] and a single adjustable (smoothing) parameter. The value normally chosen for this parameter (1.3) gives the best agreement between the solute volume description underlying ACE--the superposition of Gaussians-- and the solute cavity model that is used in the standard finite difference PB methods.

Currently, the focus in GB developments has begun to shift away from matching PB results and toward reproducing explicit solvent simulations and experimental data through reparameterization of the models.[138,229] Recent examples demonstrate that the resulting class of implicit solvent force fields can reproduce folding equilibria for both helical and β-hairpin peptides, as illustrated in Figure 5a for the folding of Trp-zip, a small helical peptide.

Speed Comparison of Implicit Solvent Models

Since reducing the required computer time is one of the primary reasons for the use of implicit solvent models, approximate timings obtained for small- to medium-sized systems are given in Table 1. The fourth column lists the computational cost for each model relative to a corresponding vacuum calculation using the same system, cutoff distances, atomic representation, and conditions. By this "intrinsic cost" measure, which gives an indication of the speed of the implicit solvent term calculation, *per se*, the implicit models are all in the range of 1.7 to 10 times slower than vacuum. As expected, the cost of the explicit water calculations (using periodic boundary conditions and particle mesh Ewald summations; see Section IV.B.) is much greater than that of the implicit models; i.e., explicit solvent calculations are approximately 20–200 times slower than the corresponding vacuum calculations, depending on the size of the system, the number of water molecules used, and the atomic representation used for the solute. Column 5 of the table lists the computational cost for each model, using its recommended cutoff distances and atomic representation, relative to a vacuum calculation on the same system using an 8 Å cutoff and a polar hydrogen representation. By this "actual cost" measure, which relates the speeds of the models when they are used as recommended (default parameters), the implicit models vary in speed by a factor of 50 or more. These differences arise primarily from the fact that the models employ different atomic representations (all-hydrogen *vs.* polar hydrogen) and nonbonded cutoff distances (8 Å in SASA *vs.* up to 20 Å in the others), in addition to having different intrinsic speeds or costs. The polar-hydrogen model has approximately two times fewer atoms than the all-hydrogen model for proteins, so that there are approximately four times fewer pairwise interactions in models 1 and 2 than in models 3–6. The longer nonbonded cutoff distances for models 4–6 mean that larger numbers of pairwise intramolecular protein interactions are taken into account. The actual cost, rather than the intrinsic cost, must be used to estimate the relative computer times that will be required for calculations with the given models. For example, MD simulations with the SASA model are up to 100–200 times faster than explicit water simulations.

Implicit Membrane Models

In the same spirit as the implicit solvent (water) potentials, implicit membrane representations reduce the required computer time by modeling the membrane environment about a solute

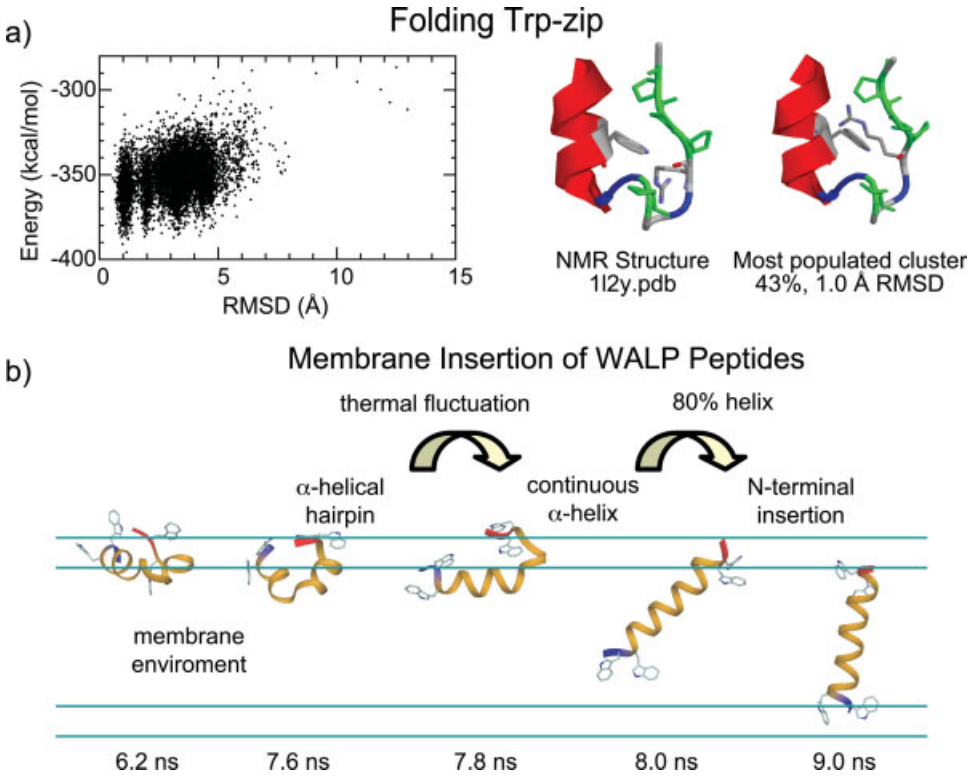

Figure 5. Combining replica-exchange molecular dynamics with implicit solvent. (a) Folding of the Trp-zip peptide.[229] A consistent parameterization of the CHARMM all-hydrogen force field and the GBSW implicit solvent model was used, with 16 replicas in a temperature range of 270 to 550 K. The left panel shows the distribution of potential energy values from the 270 K window. The right panel provides a comparison of the most populated cluster from the simulations and the NMR-derived structure; the backbone RMSD between the two structures is 1 Å. (b) Implicit membrane/implicit solvent replica-exchange molecular dynamics simulations[233] of a designed 19-residue peptide, WALP-19. The peptide inserts into the membrane via a mechanism involving the following steps: (1) migration to the membrane-water interface as a partially unstructured peptide; (2) formation of helical structure via D-hairpin conformations; (3) helical elongation through thermal fluctuations to ~80% helical; and (4) N-terminal insertion across the membrane.

(often an embedded protein or peptide) as one or more continuous distributions. Formulations based upon either PB theory (GB-like models)[230] or Gaussian solvation energy density distributions (an EEF1-type model)[135] have been developed. The first GB/IM model was developed as an extension of the simple two-dielectric form of the GB theory[219] by splitting the integral in eq. (6) into intramembrane and extramembrane parts.[136] This model has been shown to reproduce the positions of helices within a biological membrane. The introduction of a smooth switching function to describe the solute–solvent boundary[134] and the reformulation of the integration schemes for eq. (6)[216,217] have led to the introduction of a GB model that permits arbitrarily shaped low-dielectric volumes to be "embedded" in the high-dielectric solvent.[231] This model has been developed in the GBSW and GBMV modules, and it has been applied to the simulation and folding of integral membrane peptides and proteins[232] with direct comparisons to measured properties from solid-state NMR experiments[137]; it has also been used in studies of the insertion of peptides into membranes[233] and peptide association and oligomerization in membrane environments.[234] Studies of the mechanism by which insertion of designed peptides into membrane bilayers proceeds, as illustrated in Figure 5b, demonstrate the utility of implicit models in the exploration of membrane-mediated phenomena.

An EEF1-type model for implicit solvent and membrane studies (IMM1)[135] has been implemented in CHARMM. Like EEF1,[151] the method utilizes Gaussian functions to describe the extent of burial of atoms in different regions (i.e., the aqueous solvent versus the bilayer membrane). IMM1 has been extended so as to account for the surface potential due to anionic lipids,[139] the transmembrane potential,[235] and the treatment of membrane proteins with an aqueous pore.[236] It has been used to obtain insights into the forces that drive transmembrane helix association,[180,237] calculate pH-dependent absolute membrane binding free energies,[238] and determine the voltage-dependent energetics of alamethicin monomers.[235]

Table 1. Approximate Relative Computational Costs of MD Calculations Using Various Solvation Models in CHARMM (Version c34b1) for Proteins in the Approximate Range of 50 to 500 Residues in Size (750 to 7500 Atoms in the All-H Representation).

			Cost relative to:	
	Atomic representation	Outer NB cutoff (Å)	Vacuum w/ the solvation model-specific cutoff and atomic representation ("intrinsic cost")	Vacuum w/ an 8 Å cutoff and a polar H atomic representation ("actual cost")
1) SASA	polar H	8	1.5–1.9	1.5–1.9
2) EEF1	polar H	10	1.6–1.7	2–3
3) SCPISM	all H	14	1.7	10–16
4) ACE	all H	20	3.5–4.5	60–80
5) GBSW	all H	20	4.5–6	70–100
6) GBMV	all H	20	6–10	100–175
7) TIP3P	all H (solute)	16	20–60	200–500+
8) TIP3P	polar H (solute)	16	50–200	200–500+

The "atomic representation" column indicates whether the solvation model is based on a polar hydrogen (PARAM19) or an all-hydrogen (PARAM22) atomic model. (In the TIP3P calculations, this applies only to the protein, since the water model is unchanged). The "outer NB cutoff" column gives the outer cutoff distance for non-bonded interactions recommended for the model. The relative costs, or speeds, of the various solvent models show a much greater variability when they are all compared to a single vacuum calculation on a given system (last column, "actual cost") than they do when each model is compared to a vacuum calculation that uses the same atomic representation and cutoff distance (fourth column, "intrinsic cost"). See text. The TIP3P results (7,8) are for calculations using 30–60 times as many explicit water molecules as protein residues. The TIP3P calculations have a higher computational cost relative to vacuum when the simpler and faster polar H model is used for the protein. All benchmarking was performed on an Intel Pentium 4 3.20 GHz CPU with an ifort (9.0) CHARMM compilation and repeated on a 1.6 GHz AMD Opteron CPU with a gnu (gcc-4.2) compilation, using a non-bonded list update frequency of 10 steps/update.

Determination of Ionization States

Accurately simulating the electrostatic properties of a protein depends upon the correct determination of the charged state of all ionizable residues. The ionization state of a residue is determined by the free energy difference between its protonated and unprotonated forms at a given pH. This can be expressed in terms of the change in pK_a (ΔpK_a) of the amino acid in a protein relative to the intrinsic pK_a of the amino acid in solution. Correspondingly, the free energy of transfer of the charged amino acid from the solvent to the protein environment is equal to the reversible work required to ionize the side chain in the protein minus the work needed to ionize it in an isolated peptide in bulk water.[239] Although ΔpK_a can also be calculated using free energy perturbation with explicit solvent molecules (see Section VI), a PB or GB treatment representing the solvent as a dielectric continuum usually offers a convenient and reasonably accurate approximation, because the change in pK_a tends to be dominated by electrostatic contributions to the solvation free energy. The calculation of pK_a shifts can be done with the finite-difference PBEQ module.[191,192,240] Estimates of the pK_a based on the PB equation can be improved by introducing conformational sampling; e.g., calculated pK_a shifts obtained by averaging over the coordinates from an MD simulation (see Section VIII) are usually more accurate than what is calculated with a single structure.[240–243] In some cases, there is a strong coupling between the ionization states of the residues and the predominant conformation of a protein. To address this issue, a methodology has been implemented that combines the calculation of pK_a with the generalized Born methods described earlier and MD. This approach, called pH-MD,[127,129] provides a means of coupling changes in protein and peptide conformations with changes in the proton occupancy of titratable residues. The methodology utilizes an extended Lagrangian to dynamically propagate the proton occupancy variables, which evolve in the electrostatic field of the protein/solvent environment through the GBMV[216] or GBSW[134] models. The pH-MD method, which has been successfully applied to a number of protein systems,[127–129] extends the range of techniques that are available for accurately representing electrostatic interactions in solvated biological systems.

III.E. Quantum Mechanical/Molecular Mechanical Methods

Because the QM treatment of an entire biological macromolecule requires very large amounts of computer time, combined QM/MM potentials are commonly used to study chemical and biological processes involving bond cleavage and formation, such as enzymatic reactions. In this approach, a small region (the QM region) of the system, whose electronic structural changes are of interest, is treated quantum mechanically and the remainder of the system (the MM region) is represented by a classical MM force field. Typically, the former is a solute or the active site of an enzyme, while the latter includes the parts of the protein and the solvent environment that are not involved in the reaction. QM/MM methods were first used for studying poly-

ene electronic excitations in 1972[244] and carbonium ion stabilization in the active site of lysozyme in 1976.[245] Energy calculations based on the QM/MM methodology were carried out for reactions in solution and in enzymes several years later.[246]

In the QM/MM approach, electrostatic effects as well as steric contributions from the environment are incorporated directly into the electronic structure calculations of the reactive region, affecting its charge polarization and chemical reactivity.[247] A QM/MM potential employing semiempirical QM models (QUANTUM module) was first implemented in CHARMM in 1987,[248,249] through the incorporation of parts of the MOPAC program.[250] It was used for the first MD free energy simulation of an S_N2 reaction in aqueous solution[248]; numerous applications to enzymatic reactions have since been published (see, for example Refs. 251–256). Because of its ability to treat bond-forming and bond-breaking processes, to describe both the electronic ground state and excited states,[257] and to reduce the required computer time dramatically relative to full QM calculations, the QM/MM approach has become the method of choice for studying chemical reactions in condensed phases and in macromolecular systems such as enzymes and ribozymes.[258,259] In addition to the MOPAC-based QUANTUM module and its derivative SQUANTM, the semiempirical, self-consistent charge density functional tight-binding (SCC-DFTB) methods have been implemented directly in CHARMM.[260] Also, a number of external electronic structure programs have been interfaced with CHARMM and its MM force fields for use in the QM part of QM/MM calculations. In this subsection, the key features of the QM/MM module in CHARMM are summarized. Details of the theory and applications can be found in Refs. 247, 249, 256 and 261.

Treatment of Boundary Atoms

In a combined QM/MM method, the most difficult part of the system to model is the covalent boundary between the QM and MM regions[249,262]; this problem is avoided if the boundary is between molecules (e.g., between a "QM" ligand and an "MM" solvated protein). For the general case, there are three main criteria that the boundary between the QM and MM regions should satisfy.[263] First, the charge polarization at the boundary should closely approximate that obtained from QM calculations for the entire system. The effective electronegativity of a boundary atom in the MM region should be the same as that of a real QM atom. Second, the geometry at the boundary must be correct. Finally, the torsional potential energy surface at the boundary should be consistent with the surfaces arising from both QM and MM calculations.

Three approaches for treating the QM/MM boundary have been implemented in CHARMM. They are:

- Hydrogen link atom.[246,249,264] In this most commonly used approach, the valency of the QM fragment is saturated by a hydrogen atom that is introduced into the system along the covalent bond between the QM and MM regions. Although the link-atom approach has been used in numerous studies, it introduces additional degrees of freedom into the system; in addition, partial charges on the MM atoms that are closest to the link-atom must be removed to avoid convergence difficulties. The latter problem has been solved by the use of a double link-atom method[265] that incorporates a balanced bond saturation of both the QM and MM fragments.
- Delocalized Gaussian MM (DGMM) charges.[266] This method incorporates the delocalized character of charge densities on MM atoms using Gaussian functions, and it has been successfully combined with the double link atom approach. The method greatly simplifies the rules governing QM/MM electrostatic interactions.
- Generalized Hybrid Orbital (GHO) method.[263] This method partitions the system at an sp^3 atom. The boundary atom is included in both the QM calculation, with a fully optimized hybrid orbital and three auxiliary orbitals, and also the MM force field, through the retention of the classical partial charge. The method is an extension of the frozen, localized orbital approach,[267] and it neither introduces nor eliminates degrees of freedom. The GHO method has been implemented in CHARMM for semiempirical,[263] SCC-DFTB,[268] *ab initio* Hartree-Fock,[269] and DFT[270] quantum chemical models, the latter two through the GAMESS-US interface.

QM/MM Interactions

The interactions between the QM and MM regions are separated into an electrostatic term, arising from the electric field of the MM atoms, and a van der Waals component, accounting for dispersion interactions and Pauli repulsions. Although the electrostatic interaction Hamiltonian employs standard partial atomic charges of the force field, the van der Waals term includes empirical parameters for the QM atoms. Thus, like DFT itself, the QM/MM methods yield semiempirical potentials, which can be optimized by comparing interaction energies obtained from QM/MM calculations to those from fully quantum-mechanical optimizations for a database of biomolecular complexes.[249,271–276] The QM van der Waals parameters depend on the QM model and the basis set; they have been the subject of extensive validation studies.[249,271–276]

The use of combined QM/MM potentials also provides the opportunity to examine the contribution from specific energy components, including electrostatic and polarization energies. A detailed analysis of the polarization energies can be useful for developing empirical polarizable force fields,[271,277] as well as for studying the polarization energy contributions to ligand-protein binding interactions.[278] The energy decomposition method implemented in CHARMM has been used to study inhibitor-protein complexes[278] and the differential polarization energy contribution to the reactant and transition state in enzyme reactions.[279] Because the adequate treatment of long-range electrostatic effects has a large influence on the accuracy of combined QM/MM energies, an efficient linear-scaling Ewald method has been implemented in QM/MM methods.[280] In addition, an approach using the generalized SBP method[29] (GSBP; see Section IV.B.) for the treatment of electrostatics in QM/MM calculations is also available in CHARMM.[281]

Program Source for QM/MM Implementations

As mentioned, for the self-consistent-charge DFTB Hamiltonian (SCC-DFTB) methods,[282,283] and the MOPAC-derived semiem-

pirical methods (QUANTUM[249] and SQUANTM) (Nam, K., Walker, R. C., Crowley, M., York, D. M., Case, D. A., Brooks, C. L., III, Gao, J., in preparation.), the QM/MM potentials are distributed as part of the CHARMM program. In 2005, an updated version of the QUANTUM module, called SQUANTM, was developed. It features a more efficient (i.e., faster) implementation of the QM/MM potential[284] and is now the preferred module for MOPAC-type QM/MM calculations in CHARMM. In addition, there is a CHARMM interface to the MNDO97 program[206]; see also Section III.D. Interface routines have also been created for *ab initio* molecular orbital and DFT packages, including GAMESS-UK,[266,285] GAMESS-US,[286,287] CADPAC,[288] and Q-Chem.[289] Interfaces to NWChem (5.0),[290,291] Gaussian (03),[292] and MOLPRO (2006.1)[293] programs have been implemented through the recently developed MSCALE functionality in CHARMM, which is a general facility for combining potential energy functions and models. The external QM programs to which CHARMM has been interfaced have to be obtained from their authors. With the exception of Q-Chem, all of the CHARMM/QM interfaces (either internal or external) are modular in form and can be linked together with other functionalities in the CHARMM executable to carry out energy minimization and MD simulations. By contrast, Q-Chem[294,295] is interfaced to CHARMM through the exchange of external files, so that CHARMM and Q-Chem are separate executables; this facilitates the initial setup but slows down execution. Analytical first derivatives have been implemented for all of the quantum chemical models. In addition, numerical second derivatives can be calculated with the *VIBRan* subcommand *DIAGonalize FINIte*. Furthermore, numerical second derivatives for any of the CHARMM QM/MM potentials can also be computed through the POLYRATE interface (see Section VII.F.).

In all QM/MM calculations in CHARMM, each time an energy or force evaluation is required, an SCF calculation is performed. The electrostatic energy, which includes both QM and QM/MM contributions, is added to the MM energy to yield the total energy for the system. During an MD simulation or energy minimization, the density matrix from the previous step is used as the initial guess for the next SCF calculation. In evaluating QM/MM interactions, the *ab initio* molecular orbital and DFT methods include the contribution from all MM partial charges of the system, i.e., without cutoff, whereas the semiempirical modules have the option of using a cutoff list as well as the particle mesh Ewald method for periodic systems.

III.F. Restraining Potential Functions

In addition to the "physical" terms in the potential energy function, a number of different restraint terms can be applied to the system with CHARMM. These restraints are useful for the study of many problems; they can be used to restrain the system to a given conformation during various stages of a computation (e.g., energy minimization, equilibration), to introduce a biasing potential for the performance of umbrella sampling in PMF calculations (see later),[296] or, more generally, to drive the system toward a known end state in any kind of sampling procedure. The simplest type of restraint is the spatial harmonic positional restraint, in which a selected set of atoms is subjected to a quadratic potential relative to a given reference position in Cartesian space. A harmonic restraint that is a function of the "best-fit" root-mean-square deviation (RMSBFD) relative to a reference structure can also be applied to selected atoms with arbitrary weights. This restraint transiently reorients the structure relative to a reference structure with a rigid best-fit coordinate transformation, based on the selected atoms and weights, prior to the application of the distance restraints. It is analytically differentiable.[297] Internal coordinate and dihedral angle restraints can also be applied. The Miscellaneous Mean Field Potential (MMFP) module is a general facility that is used to apply spherical, cylindrical, and planar restraining potentials to a selected group of atoms or their center of mass. The module can also be used to impose a distance restraint (on two sets of atoms), a pseudo-angle restraint (three sets) or a dihedral angle restraint (four sets). Additionally, restraints on the radius of gyration as well as on contact maps can be imposed in CHARMM.[298–300] Restraints can be applied that correspond to user-specified molecular shapes (*SHAPe*) or combinations of distances (*CONStrain DISTance*). For NMR-based structural determination[90,301,302] special-case distance restraints corresponding to the Nuclear Overhauser Effect (NOE) can be imposed, as well as flat-bottomed dihedral restraints based on dihedral angle data from scalar coupling constant measurements.[303] The NOE facility also supports time-averaged distance restraints,[304] which only require restraints to be satisfied on average. The analytical forces introduced by all restraints in CHARMM are consistent with the first derivative of the energy, which is particularly important for the RMSBFD restraint.[297]

IV. Nonbonded Interactions and Boundary Methods

To complete the description of the Hamiltonian for the system, the CHARMM user needs to specify the option with which the nonbonded energy terms will be computed. In molecular mechanics calculations all atoms, in principle, can interact via the LJ and electrostatic interaction terms with all other atoms. However, the computational time for all-pair calculations scales as N^2, where N is the number of atoms; this scaling behavior leads to an excessive computational cost for large systems. For all but the smallest systems, to save time, explicit calculation of the nonbonded pairwise interaction terms is usually limited to atom pairs whose interparticle separation is less than a user-specified cutoff distance; these pairs are stored in a list, which in many applications (such as MD simulations) is not recalculated at every step. In CHARMM, this "nonbonded pair list" or "nonbonded list" may be atom- or group-based and is typically used in conjunction with various methods to treat the long-range interactions, such as extended electrostatics and long-range LJ corrections, in addition to various truncation schemes. The nonbonded lists in CHARMM can be constructed using several types of algorithms based on spatial grids or clustering methods that speed up neighbor identification significantly for large systems.

The treatment of nonbonded interactions at and beyond the boundary of the model system is also important in biomolecular

calculations, because the part of the system that is being modeled explicitly is often much smaller than the real system. In a typical example, a single protein molecule surrounded by several thousand water molecules in a 1000 nm^3 volume is used to represent about 10^{12} protein molecules and 10^{19} water molecules in a 1 μl volume of a 1 μM protein solution. Early MD simulations (e.g., the classic study of argon[2]) showed that a very small system (e.g., 256 Argon atoms) possessed many of the properties of the macroscopic liquid. Nevertheless, the limited size of the simulated system can introduce artifacts into the results. This can be due to the relatively small number of particles that interact; i.e., the protein feels the influence of far fewer water molecules in the model than it does in the real system. There are also possible surface effects, since the small simulated system has a much larger surface area/volume ratio than the real system; in the earlier example, this ratio is 10,000 times larger in the model system. The magnitude of such size-related effects can be reduced by adding an energy term that mimics the properties of the neglected surroundings, such as an SBP, or by imposing periodic boundary conditions (PBC) on the system. In PBC, all of the molecules in the central cell are surrounded by other molecules, as if there were no explicit boundaries. (Nonetheless, there can still be finite-size effects if the size of the central cell is chosen to be smaller than some intrinsic correlation length of the molecular system).[305] Also, some studies have indicated that spherical cutoff methods may introduce some artificial long-range ordering of water at water/vapor and water/lipid interfaces, an effect that is typically absent when lattice sum methods, which require PBC, are used for the calculation of electrostatic interactions[306] (see Section IV.B).

The various methods in CHARMM for the treatment of boundaries and nonbonded interactions are briefly described in this section. The reader is referred to the CHARMM documentation for further details. The optimal methods to use in a given problem are, as is often the case, a compromise between efficiency and accuracy. The user may have to test the system using two or more of the available methods for accuracy, via appropriate comparisons to experiment, and computational efficiency. Currently, for MD simulations with the fixed-point-charge force fields, the best (most accurate) approach is considered to be use of PBC systems with a nonbonded cutoff of at least 12–14 Å, the force-shifting or force-switching nonbonded options, the particle mesh Ewald treatment for long-range electrostatics, and LJ corrections for long-range van der Waals interactions. However, if the system of interest is very large, or if extended simulation times or many simulations are required, a less time-consuming SBP method may need to be employed. With the SBP methods, it is desirable to include all nonbonded interactions, possibly via extended electrostatics, or to perform electrostatic scaling,[307] in addition to applying the appropriate reaction field method for contributions beyond the boundary.

IV.A. Nonbonded Interactions

Spherical Cut-Off Methods

Calculation of the nonbonded pairwise atomic interactions, i.e., interactions between atoms not directly bonded to one another, is typically the most computationally demanding aspect of energy and energy-derivative calculations. As the number of possible pairwise interactions in a system of N atoms grows as N^2, the explicit calculation of all Coulombic and LJ terms is usually impractical for large systems. It is, therefore, necessary in systems of greater than a few thousand atoms to truncate the nonbonded interactions at a user-specified cutoff distance. The use of this approximation, which is referred to as a spherical cutoff approach, means that only atom pairs within the cutoff distance need to be included, greatly speeding up the calculation. However, it may introduce artifacts. Most notably, a simple truncation of the potential energy creates artificial forces at the cutoff distance (because of the discontinuity in the energy), which can give rise to artifacts in dynamics or structure.[308] Such artificial forces have been shown, for example, to significantly inhibit protein motion.[309] For this reason, proper truncation schemes for nonbonded interactions are an essential part of the spherical cutoff approach; this is especially true for the electrostatic interactions, which have a longer range than the van der Waals interactions. The simplest treatments consist of truncating the Coulomb interaction at the cutoff distance, while using a numerical procedure to decrease the unwanted influence of the truncation.[308] CHARMM provides a variety of truncation methods that act to smooth the transition in the energy and force at the cutoff distance, thereby reducing the errors in that region. These methods, which can be applied to both the electrostatic (Coulombic) and LJ interactions, include energy shifting and switching,[22] as well as force shifting and switching approaches.[308,310] The force shift/switch methods insure that, as the interatomic separation approaches the truncation distance, the forces go to zero in a smooth, continuous manner. These methods are, thus, particularly useful in MD simulations, where the forces determine the trajectories of the atoms, and they are the currently recommended approaches for most cases when a spherical cutoff is used. MD trajectories of even highly charged biomolecules like DNA have been shown to be stable if the appropriate smoothing functions and cutoff distances (usually at least 12 Å) are used (see later).[40,311]

Generating the Nonbonded Pair List

As stated earlier, the purpose of using finite cutoffs in energy calculations is to reduce the number of nonbonded interaction terms. However, the calculation to determine which atom pairs fall within the cutoff distance can, itself, be time-consuming. Verlet first introduced the idea of reducing the required frequency of this calculation by extending the spherical cutoff region about each atom with an additional volume shell,[312] which is referred to as a buffer region. In this technique, all of the atom pairs that are within the outer cutoff distance are determined and stored in the nonbonded list, while only the pairs that are within the inner cutoff are used in the energy (and force) calculation. This approach reduces the computer time in two ways: (1) for a fixed cutoff distance, the time for calculating energies, and forces from a nonbonded list grows linearly (rather than quadratically) with the system size; and (2) in many calculations, the list does not have to be recalculated at every step. In MD or energy minimizations, the atomic positions generally do not vary greatly from one step to the next, so that the non-

bonded list compiled with the buffer shell contains all the atom pairs that will be required in the energy calculations for the next several steps. The same list can, in principle, be used until a pair of atoms in the system moves from beyond the outer cutoff to within the inner cutoff; at the very least, one interparticle distance in the system must have decreased by the width of the buffer shell before the list needs to be recalculated. Accordingly, the "heuristic" nonbonded option in CHARMM allows the list to be automatically updated (recalculated) whenever one or more atoms have moved a distance greater than half the width of the buffer shell. The user can alternatively specify a fixed update frequency, typically from 10 to 50 steps/update; for cases in which the system configuration is changing rapidly (e.g., protein folding simulations), more frequent updates may be required. The larger the buffer shell, the less frequently the nonbonded list needs to be recalculated (but the longer it takes to calculate the list, itself). A typical buffer width used in MD simulations is 1–2 Å, although for large systems and the slow listbuilder option (see below), it is often advantageous to use a buffer width of 4 Å or more.

The use of a list and a buffer region does substantially reduce the overall CPU time for many calculations, relative to the corresponding non-list-based calculations. However for large systems, the fraction of time that is spent compiling the nonbonded list can still be significant. This is especially true if the list is calculated in a brute-force way, by distance-testing all the $N(N - 1)/2$ atom pairs. The BYGROUPS algorithm in CHARMM speeds up list generation by using standard CHARMM atomic groupings and compiling a group–group pair list (which is much faster than compiling the atom–atom list), and then calculating the atom–atom list from this shorter list. It is currently the default listbuilder in CHARMM and supports nearly all the features and options in the program (e.g., periodic boundary conditions and all free energy methods). However, since the algorithm tests all possible group–group pairs, it has $O(N^2)$ time complexity and is slow for large systems. Yip and Elber[313] developed a listbuilder algorithm that partitions the system into cubical spatial regions whose side length is equal to the outer nonbonded cutoff distance (which includes the buffer thickness) and then performs distance testing only between atoms in the same or directly adjacent cubes. This method, which was implemented in CHARMM as the BYCUBES method by Tom Ngo, has $O(N)$ (linear) time complexity and is faster than BYGROUPS for large systems. The "By-Cluster-In-Cubes" or BYCC algorithm[314] uses both the grouping and spatial partitioning techniques and, therefore, it has $O(N)$ time complexity and is faster than the other two algorithms. BYCC is approximately 2.2–2.8 times faster than BYCUBES across all system sizes and cutoff distances, and across a variety of platforms. The speed advantage of BYCC relative to BYGROUPS increases with system size and decreases with cutoff distance; for protein/water systems and a 12 Å cutoff distance, the relative speed advantage across various platforms is approximately $1 + 2 \times 10^{-4}N$ (where N is the number of atoms in the system). Hence for a 1000-atom system, the relative speed advantage is ~1.2, but for a 100,000-atom system it is ~20. For the latter system, MD simulations can be significantly faster using any of the cubical listbuilder algorithms (BYCC, BYCUBES, or BYCBIM), particularly for calculations using a thin buffer shell and high update frequencies. The memory requirements of BYCC are marginally higher than those of BYGROUPS and substantially lower than those of the other algorithms. In conjunction with the *NBACtive* command, BYCC can also calculate the list for user-specified "active" parts of the system without the need for modifying the PSF. This partial-system list feature is fundamental to a general conformational search and structure prediction module that is currently being developed in CHARMM (the Z Module, ZEROM keyword). In addition, BYCC is the basis of the domain decomposition parallel scheme being implemented in CHARMM (see Section X.B.). For a given set of atomic coordinates and cut-off distances, all three algorithms (BYCUBES, BYGROUPS, and BYCC) generate the same nonbonded list. All are also capable of generating a group–group pair list (as opposed to an atom–atom pair list), which is required by some CHARMM models (e.g., EEF1). In the group-based lists, a pair of groups are included if the separation between group centers is less than the cutoff distance. Such lists are sometimes used because they prevent the splitting of neutral groups into partially charged subgroups in the regions around the cutoff distance, which may lead to small errors in the electrostatic term. However, the use of a group list means that some atom pairs included in the energy calculations have interparticle separations greater than the cutoff distance. The BYCBIM algorithm extends the BYCUBES method to systems with images or periodic boundaries, and it (like BYGROUPS and BYCC) works for parallel simulations. It is currently the most efficient listbuilder in CHARMM for calculations involving image atoms.

Extended Electrostatics

The Extended Electrostatics model approximates the full electrostatic interactions of a finite set of particles by partitioning the electric potential and the resulting forces acting on a particle i located at r_i into a "near" and an "extended" contribution.[315] The near contribution arises from the charged particles which are spatially close to r_i (within a cutoff distance), while the extended contribution arises from the particles which are spatially distant from r_i. The total electrostatic potential can be written as a sum of the two. Interactions between particles within the cutoff distance are calculated by a conventional pairwise additive scheme, whereas interactions between particles separated by a distance greater than the cutoff are evaluated using a time-saving multipole (dipole and quadrupole) approximation. The energy and forces are calculated by explicitly evaluating pairs in the near-neighbor list and using the stored potentials, fields, and gradients to approximate the distant pairs. The electric potential and its first and second derivatives are calculated only when the nonbonded list is updated and stored. This simple approximation is based on the assumption that for distant pairs the atomic displacements are sufficiently small between updates and that the changes in their electrostatic interactions can be accurately calculated using local expansions. The approach is particularly useful for efficiently including electrostatic interactions at all distances in the treatment of a finite system, which is simulated using SBPs such as stochastic boundary

potential (SBOU),[27] spherical SBP (SSBP),[28] and GSBP[29] (see Section IV.B.). Examples are given in free energy difference calculations.[316] The method has been extended to include higher order multipoles in a CHARMM implementation of the fast multipole method[317] (FMA module). An alternative method for the rapid calculation of the long-range electrostatic energies and forces in a system is Linear Time Complexity Reduction (LTCR). In this method, the $1/r_{ij}$ dependence of the electrostatic term is approximated as a polynomial in the squared distance, so that the double sum over pairwise electrostatic interactions can be rewritten as a functional of single sums over single-particle terms.[318]

Long-Range LJ Corrections

Correction schemes for the LJ energy and virial beyond the atom truncation distance have been implemented in CHARMM. One method (invoked with the *LRC* option of the *NBONd* command) determines the number density of each atom type in the system, and applies an isotropic correction to the LJ energy and virial acting on each atom in the system.[8] A second method is script-based, makes no isotropic assumptions, and calculates the correction to the virial explicitly, resulting in a more accurate pressure and surface tension. The latter method does not correct for the energy changes associated with truncation[319] and it is significantly more costly than an LRC calculation; however, because the virial correction does not need to be updated at every step in MD simulations (instead, e.g., every 100 or 1000 steps), the overall cost of the anisotropic correction can be reduced. Lastly, the long-range LJ interactions can be calculated using the Isotropic Periodic Sum (IPS) method described later. The IPS method calculates long-range interactions using the so-called isotropic and periodic images of a local region around each particle. It corrects not only energies, but also the forces and the virial. Because IPS assumes that the distant environment around an atom is similar to (and as heterogeneous as) the local environment, it preserves the density of the system, and the incorporation of contributions from the long-range interactions into the short-range potential gives more accurate results than those obtained with an isotropic long-range correction.

IV.B. Boundary Conditions

Solvent Boundary Potentials

One approach for simulating a small part of a large system (e.g., the enzyme active site region of a large protein) uses an SBP. In SBP simulations, the macromolecular system is separated into an inner and an outer region. In the outer region, part of the macromolecule may be included explicitly in a fixed configuration, while the solvent is represented implicitly as a continuous medium. In the inner region, the solvent molecules and all or part of the macromolecule are included explicitly and are allowed to move using molecular or stochastic dynamics. The SBP aims to "mimic" the average influence of the surroundings, which are not included explicitly in the simulation.[27,28] There are several implementations of the SBP method in CHARMM. The earliest implementation, called the SBOU, uses a soft nonpolar restraining potential to help maintain a constant solvent density in the inner or "simulation" region while the molecules in a shell or buffer region are propagated using Langevin dynamics.[27] By virtue of its simplicity, this treatment remains attractive and it is sufficient for many applications.[320,321] To improve the treatment of systems with irregular boundaries in which part of the protein is in the outer region, a refinement of the method has been developed that first scales the exposed charges to account for solvent shielding and then corrects for the scaling by postprocessing.[307]

The SSBP, which is part of the MMFP module (see Section III.F.), is designed to simulate a molecular solute completely surrounded by an isotropic bulk aqueous phase with a spherical boundary.[28] In SSBP, the radius of the spherical region is allowed to fluctuate dynamically and the influence of long-range electrostatic interactions is incorporated by including the dielectric reaction field response of the solvent.[28,29] This approach has been used to study several systems.[322–325] Because SSBP incorporates the long-range electrostatic reaction field contribution, the method is particularly useful in free energy calculations that involve introducing charges.[322–325]

Like the SBOU charge-scaling method,[307] the GSBP is designed for irregular boundaries when part of the protein is outside the simulation region.[29] However, unlike SBOU, GSBP includes long-range electrostatic effects and reaction fields. In the GSBP approach, the influence of the outer region is represented in terms of a solvent-shielded static field and a reaction field expressed in terms of a basis set expansion of the charge density in the inner region, with the basis set coefficients corresponding to generalized electrostatic multipoles.[29,326] The solvent-shielded static field from the outer macromolecular atoms and the reaction field matrix representing the coupling between the generalized multipoles are both invariant with respect to the configuration of the explicit atoms in the inner region. They are calculated only once (with the assumption that the size and shape of inner region does not change during the simulation) using the finite-difference PB equation of the PBEQ module. This formulation is an accurate and computationally efficient hybrid MD/continuum method for simulating a small region of a large macromolecular system,[326] and is also used in QM/MM approaches.[281,327]

Periodic Boundary Conditions and Lattice Sum Methods

CHARMM has a general image support facility that allows the simulation of symmetric or periodic boundary systems. All crystal forms are supported, as well as planar, linear, and finite point groups (such as dimers, tetramers, etc.). Figure 6 depicts the simulation of a virus capsid where icosahedral symmetry has been imposed so that it is necessary to represent explicitly only 1/60th of the entire capsid.[328] It is also possible to build a unit cell related to its neighbors with any space group symmetry, to optimize its lattice parameters and molecular coordinates, and to carry out a vibrational phonon analysis using the crystal module (CRYSTAL),[329] which is an extension of the original image facility.[22,330] Simulations allowing lipids in opposing membrane leaflets to exchange can be carried out using $P2_1$ boundary conditions.[330] The image facility achieves its generality by treating image atoms (coordinates and forces) explicitly, thus avoiding the size and transformation limitations inherent in the more

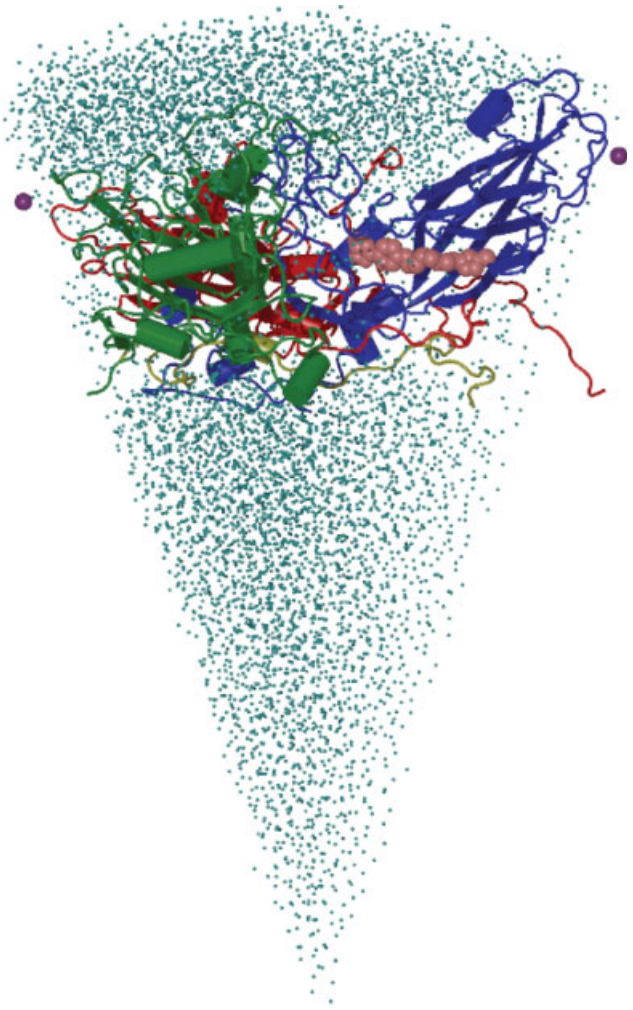

Figure 6. The protomeric unit of HRV14 (ribbon) capsid comprising VP1 (blue), VP2 (green), VP3 (red), and VP4 (yellow) peptide chains and two calcium ions (purple spheres). The protomer is solvated on the inside and outside with water molecules shown as small cyan spheres (which fill the interior of the capsid space). The primary unit has 12,432 protein atoms and 19,953 water atoms. Symmetry conditions, imposed through the use of the general image facility in CHARMM, model the entire virus capsid of ~750,000 atoms.[328] This illustrates the use of molecular symmetry in the CHARMM program to reduce the size of a calculation in large systems.

commonly used minimum-image convention. This also allows the virial to be computed with a single-sum method for a rapid evaluation of the pressure.[8] Bond linkages (with additional energy terms including bond angle, dihedral angle, and improper dihedral angle terms) can be introduced between the primary atoms and image atoms to allow the simulation of "infinite" polymers, such as DNA, without end effects. For infinite systems, the simulation can be restricted to the asymmetric unit because arbitrary rotations, translations, and reflections can be applied to generate the coordinates for larger versions of the system (see also Fig. 6). To ensure better numerical stability in the volume and shape fluctuations of the unit cell during constant-pressure Nosé–Hoover–Andersen–Klein[331] dynamics, the symmetry operations on the central cell are handled internally by keeping the atomic coordinates in a symmetric projection of the unit cell vectors. The latter condition is imposed to prevent unwanted torque on the system due to box shape changes (e.g., in the triclinic case).

If periodic boundary conditions are imposed on the system, the electrostatic energy can be expressed as a lattice sum over all pair interactions and over all lattice vectors. Namely,

$$U(\vec{r}^N) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{\vec{m}}{}' \frac{q_i q_j}{4\pi\varepsilon_0\left|\vec{r}_j - \vec{r}_i + \vec{m}\right|}, \tag{7}$$

where r_i is the position vector and q_i is the charge of particle i, N is the number of atoms in the unit cell, $\vec{m}$ is the lattice vector of the (real space) periodic array of unit cells, and the prime on the sum indicates that $j \neq i$ when $\vec{m} = 0$. This sum converges conditionally—i.e., it depends on the order of the summation over unit cells—and slowly.

The method developed by Ewald[332] transforms the summation to two more complicated but absolutely and rapidly convergent sums, plus a "self-energy" term and a "dipole" term. The dipole term, which captures the conditional convergence of the original sum and includes the external reaction field conditions, can be made to vanish (see below). The total electrostatic energy, $U(\vec{r}^N)$, then equals

$$U(\vec{r}^N) = \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{2\pi V}\sum_{\vec{k}\neq 0}\frac{\exp(-\pi^2 k^2/\kappa^2)}{k^2}\exp\left(2\pi I\vec{k}(\vec{r}_j - \vec{r}_i)\right) + \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\vec{m}=0}^{\infty}\frac{q_i q_j erfc\left(\kappa\left|\vec{r}_j - \vec{r}_i + \vec{m}\right|\right)}{\left|\vec{r}_j - \vec{r}_i + \vec{m}\right|} - (\kappa/\sqrt{\pi})\sum_i {q_i}^2, \tag{8}$$

where *erfc* is the complementary error function, κ is a constant, k is the reciprocal space lattice vector, V is the volume of the unit cell, and I is the imaginary unit. The first term is a reciprocal space sum over all pairwise interactions (both short- and long-range) in the infinite lattice, in which the charge distributions about each particle are spherical Gaussians. The second term is a direct sum over all short-range pairs and consists of two components: A) the point-charge interactions between the short-range pairs and B) a term that cancels the contributions of these pairs in the first term (reciprocal space sum); i.e., the latter component subtracts the interactions between the Gaussian charge distributions for all short-range pairs. The third term, which is the self-energy term, provides the same type of cancellation for each Gaussian charge distribution in the unit cell interacting with itself. The parameter κ does not affect the total energy and forces, but rather adjusts the relative rates of convergence of the real and reciprocal space sums; it is usually chosen so as to optimize the balance of accuracy and efficiency of the calculations. If κ is chosen to be large enough, only the $\vec{m} = 0$ elements contribute to the second (short-range) term, and it

reduces to the minimum-image convention sum. The triple sum of the first term can be rewritten as a double sum over $\vec{k}$ and i. The dipole term[333,334] can be added to account for the effects of the total dipole moment of the unit cell, the shape of the macroscopic lattice, and the dielectric constant of the surrounding medium. However, this term vanishes in the limit that the net dipole moment of the unit cell, which is origin-dependent and affected discontinuously by image wrapping, vanishes, or the external dielectric constant goes to infinity (so-called "tin foil" boundary conditions). In CHARMM, because interactions between 1,2 and 1,3 bonded atom pairs are excluded from the point-charge part of the direct sum and, hence, do not appear in the second term of eq. (8), their contributions to the reciprocal sum are corrected for in a separate calculation (EWEX term).

Recent variants of the Ewald method, which employ pairwise cutoff lists for the direct sum, charges on grids, and fast Fourier transforms, greatly enhance computational performance. One of these, the particle mesh Ewald (PME) method,[335,336] has been incorporated into CHARMM. Although convergence of the Ewald summation requires neutrality of the unit cell, the Ewald and PME methods can be used for a system carrying a net charge by the effective superposition of a structureless neutralizing background onto the unit cell. CHARMM optionally computes both the energy and the virial correction terms for the net charge case,[337] which may be included with a user-specified scale factor that is optimally determined by the dielectric.[338,339] The treatment of the long-range electrostatics based on PME, and the constant pressure and constant surface tension simulation algorithms[340] are implemented for the crystal symmetries as defined in the CRYSTAL facility. Consequently, the CRYSTAL facility must be used for such calculations in CHARMM.

Although the Ewald and PME methods are formally applicable to periodic systems, it is also possible to use them to calculate the electrostatic energy and forces within a finite isolated cluster without cutoff effects. The method relies on truncation of the $1/r$ Coulomb potential at a finite range R. To remove all interactions between charges belonging to neighboring unit cells while keeping those within a finite cluster of diameter s, it is sufficient to sum over all lattice vectors using a filter function-modified Coulomb potential[341] with finite range R, such that $s < R < L - s$, where L is the center-to-center distance between neighboring cells. With this modification, the PME methods can be used to rapidly compute energies and forces with no interference from periodicity and with nearly linear scaling.[342]

PBC with the minimum-image convention can also be used in CHARMM through the PBOUND module, but the facility does not currently support constant-pressure MD and an Ewald description of the electrostatics.

IPS for Long-Range Interactions

The IPS method[343] is a general method for calculating long-range interactions, that, unlike Ewald-based methods, does not sum contributions over lattice images. Instead, so-called "isotropic" periodic images are assumed to represent remote structures. The isotropic and periodic character of the images simplifies the summation of long-range interactions relative to a summation over lattice images. The IPS method reduces the calculation of particle interactions to the calculation of short-range interactions within a defined region (a cutoff distance) plus long-range interactions given by IPSs. Because of the periodicity of the image regions, the total forces acting on one atom from a second atom and all of its images goes smoothly to zero at the boundary of the local region about the first atom, so that no truncation is needed. Simulation results have shown that for a LJ fluid, the energy, density, and transport coefficients are nearly independent of the cutoff distance for all but the shortest cutoff distances (less than $\sim$8 Å).[343,344]

Analytic solutions of IPS have been derived for electrostatic and LJ potentials, but it can be applied to potentials of any functional form, and to fully and partially homogeneous systems, as well as to nonperiodic systems. Customized formulations of the method have been developed for use in systems with 1- or 2-dimensional homogeneity (1D or 2D IPS); for example, 2D IPS can be used for membrane systems. For liquid/vapor interfaces, 2D IPS is exact when the interface is homogeneous in the interfacial plane. Because 2D IPS assumes a finite thickness of an interfacial system, it is not suitable for liquid-liquid interfacial systems where the thickness is infinite. For liquid/liquid interfaces, such as lipid bilayers in water, PME/IPS (PME for electrostatics and three-dimensional [3D] IPS for van der Waals interactions) appears to provide the most realistic conditions. The PME/IPS method is in excellent agreement with large cutoffs for interfacial densities and dipole potentials and only slightly underpredicts the surface tensions,[345] though the method is not exact for the long-range interactions in these inhomogeneous systems. For true lattice systems where long-range structure can be accurately described by periodic boundary conditions, IPS is less accurate than lattice sum-based methods like PME. Recent advances in the IPS method to include a second longer cutoff (Wu, X. and Brooks, B.R., submitted for publication) have eliminated many of the aforementioned problems.

The IPS method is computationally efficient and is readily parallelized, in part because, unlike PME, it does not require the calculation of Fourier transforms. The communication scheme is similar to that for other cutoff-based methods.

V. Minimization, Dynamics, Normal Modes, and Monte Carlo Methods

An essential element of CHARMM functionality is the calculation of the energy and its derivatives, because this makes possible the study of many properties by energy minimization, Monte Carlo sampling, normal mode analysis, and MD. CHARMM provides a number of minimization methods and several approaches to the propagation of trajectories that allow for the sampling of a variety of ensembles.

V.A. Energy Minimization

CHARMM supports a number of minimization methods (*MINImize* command) that rely on either the first derivatives or the first and second derivatives of the energy function [eq. (1)]. Multiple methods are included in the program because each one

has its advantages. They include the simplest method, Steepest Descent (SD), and other first-derivative methods such as a variant of the Fletcher–Powell algorithm and a conjugate gradient technique (CONJ). The latter two methods obtain better convergence than SD by including information on the derivatives from prior points of the minimization. The second-derivative methods operate in either the full space of the Hessian (Newton–Raphson, NRAP) or in a subspace of the full Hessian (Adopted-Basis set Newton–Raphson, ABNR). The NRAP algorithm has additional features that can force it off a saddle point; these are useful, for example, when the initial structure has unwanted symmetry. A minimization method that is intermediate between the first-derivative and full Hessian methods, the truncated-Newton (TN) minimizer (TNPACK), has also been implemented in CHARMM.[346] This approach is comparable to ABNR with respect to computational efficiency, though its convergence is better, particularly for systems with less than 400 atoms. In general, the first-derivative methods are more robust in the initial stages of energy minimization calculations, while the NRAP and ABNR or TNPACK techniques provide better convergence to the local minimum when there are no large gradient components. Typically, initial minimizations are performed using the first derivative methods, usually beginning with SD, especially in cases where there are bad contacts causing a large initial gradient. This is followed by the NRAP method for small systems (<300 atoms), or ABNR or TNPACK when NRAP matrices become too large. Methods such as SD and CONJ are also more robust than second derivative methods when faced with energy and force discontinuities that occur with some energy terms and options (e.g., electrostatic truncation).

In addition to potential energy minimization, local saddle points may be identified in CHARMM by minimizing the norm of the potential energy gradient (*GRAD* option of *MINImize* command). Depending on the initial conditions, the search will either be terminated at a minimum or a saddle point on the potential energy surface. This feature is primarily used for determining first-order saddle points. As the second-derivative matrix is employed to calculate first derivatives of the target function in this method, it is much slower than ABNR and NRAP and, therefore, is not recommended for more standard energy minimizations. Alternatively, saddle points can be located using the *SADDle* option associated with NRAP. This option identifies the most negative eigenvalue(s) and maximizes along the corresponding eigenvector(s) while minimizing in all other directions. Another approach to finding accurate saddle points is implemented as part of the TREK module (Section VII.A.).

V.B. Molecular Dynamics

Classical MD simulations are used for evaluating the structural, thermodynamic, and dynamic properties of biomolecular systems.[4] Such simulations require integration of Newton's equations of motion, which determine the coordinates of the system as a function of time. The principal assumption in the use of MD is that classical dynamics is adequate and that quantum corrections to the atomic dynamics are negligible. This assumption is valid for most problems of interest in macromolecular biological systems; i.e., above ~50 K, for a given biomolecular potential energy surface, the classical and QM descriptions of the dynamical properties of interest effectively coincide.[24,347,348] Notable exceptions arise in chemical reactions (proton tunneling; see Section III.E.). Also, for the estimation of the absolute entropy (and free energy), higher temperatures are required to reach the classical limit; however, for entropy and free energy difference calculations the classical treatment often provides a good approximation even at room temperature because the low-frequency modes make the dominant contribution.[349] This, of course, provides the theoretical basis for the widely used classical free energy simulation methods (Section VI.A.).

MD trajectories in CHARMM are controlled by the general and multi-optioned *DYNAmics* command. A single call to *DYNAmics* can initiate, propagate, and terminate a trajectory, as well as specify options for the dynamics integration scheme, nonbonded interactions, the image atom list, thermostats, heating schedules, initial assignment and rescaling of velocities, statistical ensembles, system recentering, the generation of binary trajectory and velocity files, the output of formatted files containing coordinates, forces, and velocities, the writing of energy statistics to standard output, and the reading and writing of restart files. The algorithms by which the atomic positions of the system are propagated after the computation of the forces are called dynamics integrators. There are currently five supported integrators within CHARMM: ORIG, LEAP, VVER, VER4, and VV2. Each integrator is unique and has its own strengths and limitations. The standard integrator, LEAP, is based on the Verlet leap-frog algorithm. It is the most general and most widely used of CHARMM's integrators and has the largest number of supported features. The leap-frog algorithm was selected to be the standard because, in its simplest form, it is an efficient, high-precision integrator with the fewest numerical operations.[8] The newest integrator, VV2, which is based on a velocity Verlet scheme with improved temperature and pressure control,[350] has been implemented to support polarizable models based on the classical Drude oscillators.[104] The oldest integrator, ORIG,[22] is based on the lower-precision Verlet three-step method. This is the most limited of the CHARMM integrators, but it is retained for historical reasons and testing of other integrators. The original velocity Verlet integrator, VVER, is also a high-precision integrator that supports a multiple-time-step method (MTS),[351] but it is otherwise limited (e.g., no pressure calculation). The leap-frog integrator has been extended to a theoretical 4th spatial dimension in the development of the VER4 integrator[352] for the purpose of enhanced conformational sampling in 4-dimensional MD (Section VI.E.); the integrator is usable only for this function.

The standard Verlet MD integration scheme or one of its variants is often used to perform simulations in the microcanonical ensemble (*NVE*), in which the total energy and volume are constant. The *NVE, NVT* (canonical), and *NPT* (isothermal-isobaric) ensembles are the "workhorses" of contemporary MD simulations. *NPT* is often useful during equilibration for achieving the desired water density in a system with explicit solvent; once the system is stable, a change to the *NVE* or *NVT* ensemble may be appropriate. For testing and evaluating new simulation methods, the *NVE* ensemble has the advantage that energy conservation

can be used as a necessary (though not sufficient) diagnostic for the validity of the calculations. The leap-frog integrator also calculates a high-frequency corrected total energy[353] which eliminates the time-step dependence of the total energy. Since the Verlet integration methods are symplectic, in the absence of constraints like SHAKE,[354] this corresponds to monitoring energy drift with a shadow Hamiltonian.[355] Moreover, constrained dynamics with Verlet and SHAKE is symplectic if the constraints are introduced with sufficient accuracy.[356]

Using this approach, the fluctuation in the total energy has been typically observed to decrease by one order of magnitude or more. By eliminating high frequency noise, small changes in the total energy become more readily observable. A similar approach is also used for the piston degrees of freedom (see later) to allow an accurate estimate of the transfer of heat into a constant temperature and pressure system. Both velocity reassignment and velocity scaling can be performed with the Verlet-type integrators to couple atoms in the simulation volume to a heatbath; velocity scaling is often used to gradually heat or cool a system targeting a desired temperature.

All the integrators are consistent with the use of SHAKE-type methods[354] for the imposition of holonomic constraints. These constraints can be employed, for example, to fix the length of covalent bonds involving hydrogen atoms when these motions are not of specific interest, as is the case in most applications of MD simulations not involving vibrational spectrum analysis or proton NMR. SHAKE-type constraints are used for fixing the relative positions of charges that are not localized on atoms, as in the early ST2 water model,[73] the TIP4P model,[50] and other more elaborate water models. Eight types of holonomic constraints are available in CHARMM. When more than one type of constraint is applied, an iterative, self-consistent approach is used to satisfy all constraints. The supported constraints include: *SHAKe* (simple distance constraints), *LONEpair* (general massless particle constraint facility; preprocessor keyword LONEPAIR), *CONStrain FIX* (atomic positional constraints), *ST2* (required restraints for the ST2 water model that are activated on PSF *GENEration* when ST2 is the residue type), *FIX* (a *TSM* subcommand used for fixing internal coordinates), *RIGId* (a *SHAPes* option that creates a rigid body object), SHAKA4 (a *SHAKe* subcommand of *FOUR* for constraints in the 4th spatial dimension), and *PATH* (path constraints to keep the structures on a particular hyperplane, used with the RXNCOR facility; see Section VII.C.). *SHAKe* allows the use of a longer timestep, typically 2 fs, when integrating Newton's equations of motion.[351,354,357] The lonepair facility is a general constraint code for all "massless" particles in CHARMM, with the exception of those in the ST2 water model. On each iteration, massless particle positions are determined relative to atomic positions, and the forces calculated on massless particles are transferred to atoms in such a manner as to preserve the net torque and force. The use of the *CONStrain FIX* command can significantly improve speed, since it results in the removal of constrained atomic pairs or groups from the nonbonded lists required for the calculation of the energy and forces. All of these constraints include a pressure correction term, which arises from the fictitious forces on the system that maintains the constraints.

Ensembles for Dynamics

Several constant temperature (*NVT*, canonical ensemble) and pressure (*NPT*) methods can be used with the equations-of-motion integrators. Constant temperature and pressure simulations can be performed with CHARMM using methods that are based on the ideas of extended Lagrangian dynamics.[331,358] This approach ensures that well-defined statistical ensembles are achieved. Also, multitemperature controls are available, through which the temperatures of different parts of the system are coupled to different thermostats. This can aid in equilibrating the system or in keeping the system at the desired temperature when its components (e.g., protein and its water environment) have significantly different properties; an interesting application of such multiple thermostats involved keeping a protein and its solvent shell at different temperatures.[359] The Nosé–Hoover heat bath methods work with the leap-frog Verlet and velocity-Verlet integrators in CHARMM. For *NPT* simulations, the Hoover heat bath method can be used in conjunction with a pressure coupling algorithm designated as the Langevin Piston.[360] This is a robust method in which Langevin-type random and frictional forces are applied to piston degrees of freedom (e.g., during MD equilibration) to obtain a valid thermodynamic ensemble. Methods for other ensembles as variants of this approach are available in CHARMM, as described in the work by Zhang et al.[340] A corresponding method is used in simulations of lipid bilayers and other interfacial systems in which a constant surface tension is maintained.

A modified velocity-Verlet algorithm is available to simulate systems in which induced polarizability is represented with classical Drude oscillators that are treated as auxiliary dynamical degrees of freedom.[104] The familiar SCF regime is simulated if the auxiliary Drude particles are reset to their local energy-minimum positions after every timestep of the physical atoms, but this procedure is computationally inefficient. The SCF regime can be approximated efficiently with two separate Nosé–Hoover thermostats acting on the polarizable atoms and their auxiliary Drude particles. The first thermostat, coupled to the center-of-mass of the atom-Drude pair, keeps the true physical degrees of freedom at any desired temperature. The second low-temperature thermostat ($\sim$1 K), acts on the relative atom-Drude motion within the reference frame of the center-of-mass of each pair to control the amplitude of the classical oscillators relative to their local energy minima. In its CHARMM implementation, the double-thermostat velocity-Verlet algorithm allows efficient SCF-like constant-pressure, constant-temperature MD simulations of systems of polarizable molecules with a timestep of $\sim$1 fs.

In addition, a modified Berendsen method[361] has been implemented that allows for both constant temperature and constant pressure simulations. Although the Berendsen approach works well for small systems and for very weak coupling constants, and has been widely used, it may lead to differential heating of heterogeneous systems, most notably interfacial systems.[360] Furthermore, the resulting MD trajectory does not correspond to

any thermodynamic ensemble. Thus, the methods for *NVT* and *NPT* simulations described earlier are recommended over this method, despite its advantage of ease of use.

Non-Verlet Integrators

Langevin dynamics (LD) simulations, which propagate the system coordinates with the Langevin equation,[362] rather than Newton's equations, include random and frictional forces that mimic the effects of the environment on the dynamics of the simulated system.[363,364] Coupling a fully solvated system to a Langevin heatbath is an effective way of maintaining a constant-temperature ensemble. This type of Langevin heatbath coupling can be used as a complement to the implicit solvation methods (Section III.D.), which treat the effect of solvent on the solute energy but do not include the frictional and dissipative properties of solvent. LD is also used in stochastic boundary simulations. It is suitable for studying long-time-scale events that occur in macromolecules, such as protein folding. LD is also useful for small systems, such as small molecules in the gas phase, where the temperature based on the atomic velocities is poorly defined and the free energy transfer between modes can be very slow.

V.C. Normal Mode Methods and Harmonic Dynamics

CHARMM has a comprehensive utility for molecular vibrational analysis, called VIBRAN. The VIBRAN module includes basic tools for calculating normal modes of vibration, either with the full atomic basis or with a reduced basis in which some degrees of freedom are constrained. An example of the latter is the calculation of normal modes using only the dihedral angle degrees of freedom. The module also has the capacity to generate quasi-harmonic modes of vibration from MD simulations with either the full or reduced basis. Quasi-harmonic modes of vibration are the normal modes of vibration of a harmonic potential energy surface that would generate the same fluctuation matrix, when every mode is populated with k_BT* of energy, as that calculated from an MD simulation. There is also an extensive set of analysis tools that facilitate the analysis of normal modes. The VIBRAN facility was summarized in the original CHARMM article[22] and later described in considerable detail.[365–367] This section will primarily focus on developments that have occurred since the latter publications.

The VIBRAN module provides the means for calculating thermodynamic properties of a system from the vibrational analysis in terms of normal or quasi-harmonic modes. An example is the calculation of the configurational entropy from normal modes obtained via quasi-harmonic analysis. These results can be combined with the overall rotational and translational contribution to the entropy and with other energetic information (i.e., vibrational enthalpies, free energies of solvation from continuum electrostatic methods) to obtain the free energies of ligand–protein,[368] protein–protein,[369] or protein–DNA interactions.[141]

There has been considerable effort in developing efficient methods for the harmonic analysis of very large biomolecules when only a few lowest-frequency modes are of interest. A number of studies[370–374] have shown that low-frequency modes, which reflect the natural flexibility of the system, often provide important functional information about biomolecules that undergo significant conformational transitions. One approach involves an iterative diagonalization in a mixed basis (DIMB),[375,376] which requires considerably less computer memory than the full basis calculation, yet converges to the same result. The method involves repetitive reduced-basis diagonalizations, where the reduced bases are constructed partially from the approximate eigenvectors and from the Cartesian coordinates. Another approach breaks the system into rigid blocks, typically one residue each, or larger. Because of their collective nature, the low-frequency modes of the system can be computed rather accurately with such a block normal mode (BNM; rotational-translation-block) approach.[377,378] In this approach, the atomic Hessian is projected into a subspace spanned by the rotational and translational motions of the blocks. The projection dramatically reduces the size of the matrix to be diagonalized and thus the cost of computation. The current implementation in CHARMM also has the option of using an iterative diagonalization procedure for sparse matrices, which makes it possible to obtain low-frequency modes of large biomolecular assemblies such as the 30S and 50S ribosome.[379] Compared with even more simplified approaches such as the elastic network model[380,381] (which is also available in CHARMM; see Section VII.E.), the BNM method has the advantage of using the full physical potential energy function [eq. (1)], which makes it possible to obtain detailed information for many kinds of biomolecules[382,383] and permits the inclusion of co-factors and ligands in a straightforward way. A comparison of CHARMM BNM[383] with a series of elastic models demonstrated the superiority of the former for calculating anisotropic B factors.

Normal mode calculations can also be carried out with QM/MM potential functions.[384] This capability is especially useful for spectroscopic characterization of the active sites of metalloenzymes,[379] characterization of stationary points along reaction pathways in enzymes, and estimates of the vibrational contributions to the activation free energy for reactions in complex systems.[254,385] With careful parameterization, QM/MM vibrational analysis can also be used to compute nonlinear infrared spectra,[386] which contain valuable information regarding the fast time-scale dynamics of condensed-phase systems. The standard implementation in the release versions of CHARMM (Section XI.) computes the second derivative matrix using finite differences of the analytical first derivatives for many of the QM methods, including AM1, PM3, and SCC-DFTB, which are included in CHARMM (Section III.E.), and other *ab initio* or density functional methods that are available in separate QM packages. QM/MM analytic second derivative support has been implemented for the Q-Chem/CHARMM interface.[387] Also, analytical computation of QM/MM second derivatives[384] are currently available in a specialized version of CHARMM in conjunction with the GAMESS-US package.[286,287]

Quasi-Harmonic Analysis

Quasi-harmonic normal modes can be extracted from a trajectory by diagonalizing the mass-weighted covariance matrix of

*k_B is the Boltzmann constant; T is the absolute temperature.

the atomic displacements from their average positions.[365] These modes are similar to the normal modes obtained from diagonalization of the Hessian, but contain anharmonic contributions as well. Once the covariance matrix has been obtained, the diagonalization can be performed on the submatrix corresponding to any subset of atoms, effectively allowing the analysis to be applied to individual residues, or just to the backbone or side chains. The modes, harmonic or quasi-harmonic, can be saved to disk for visualization, or their character can be further analyzed in terms of the contributions of individual atoms. The eigenvalues, which are related to the frequencies of the motions, can be inserted into the $3n$-dimensional harmonic oscillator expressions for the entropy, enthalpy or heat capacity[388] of the (sub)system, where n is the number of atoms. The calculation of converged quasi-harmonic entropies often requires lengthy trajectories.[389] In addition to the configurational (vibrational) entropy, the rigid-body translational/rotational contribution to the entropy can also be computed from a trajectory. For this, the (quasi)harmonic interpretation is not required and, in the absence of mass weighting, the method is identical to the standard multivariate statistical method of principal component analysis (PCA),[390] with the computed frequencies inversely proportional to the variances of the atomic displacements of the trajectory along the eigenvectors. PCA has been used to extract dominant motions in proteins in, for example, "essential dynamics."[391]

V.D. Monte Carlo Methods

In Monte Carlo (MC) simulations, random changes (moves) made to the configuration of a system are accepted or rejected in such a way as to obtain a chain of states that samples a well-defined probability distribution.[392] MC need not follow a realistic path for ensemble averages to converge, which makes it useful for simulating relaxation processes that occur on timescales that are much longer than the fastest motions of the system (typically bond stretches in biomolecular systems). Despite this advantage, there are far fewer MC than MD studies to date because initial comparisons between the two methods suggested that MC samples protein configurations inefficiently.[393] However, improved move sets now allow much faster decorrelation of observables, making MC the method of choice in many cases requiring the search of a large conformational space.[394,395] Certain features and applications of the MC module in CHARMM are summarized here; for more details, see Hu et al.[395]

Background

The sampling of a system with a series of (pseudo)randomly generated states is a Monte Carlo process. From these states, an estimate of the thermal average of quantity B over all states x_i in a system at temperature T is given by

$$\langle B \rangle \approx \frac{\sum_{i=1}^{n} \frac{B(x_i)\exp(-E(x_i)/k_BT)}{P(x_i)}}{\sum_{i=1}^{n} \frac{\exp(-E(x_i)/k_BT)}{P(x_i)}}, \quad (9)$$

where n is the number of sampled states, $E(x_i)$ is the energy of x_i and $P(x_i)$ is the probability of x_i appearing in the sampled population. Metropolis et al. (1953)[392] first noted that an efficient choice of $P(x_i)$ is the Boltzmann probability itself—i.e., $P(x_i) \propto \exp(-E(x_i)/k_BT)$. In this case, eq. (9) reduces to a simple arithmetic average: $\langle B \rangle \approx \sum_{i=1}^{n} B(x_i)/n$. One of the aims of Monte Carlo calculations is to sample the system according to the canonical probability distribution; many other importance sampling methods are based on a similar approach. In the Metropolis method, this weighting of sampled states can be achieved by accepting or rejecting a series of changes from a predefined set of possible ones (a move set) according to the acceptance probability $P_{acc,i} \approx \min(1,\exp(-\Delta E_i/k_BT))$, where ΔE_i is the change in energy between the ith state (conformation) and the previously accepted one. The series of accepted states so generated is referred to as a Markov chain. The Metropolis method satisfies the condition of detailed balance, which implies that, at equilibrium, the average number of moves between two arbitrary states is the same in either direction; this is sufficient (though not necessary) for sampling in the canonical ensemble.

Ensembles

MC in CHARMM can sample from the canonical (NVT),[392] isothermal-isobaric (NPT),[396] and grand canonical (μVT)[397] ensembles. Because the grand canonical MC algorithm allows particles to be inserted into and deleted from the system as though exchanging with a bulk solvent reservoir of known excess chemical potential (μ), it is very useful for solvating macromolecules, especially ones with restricted access to cavities.[398] Woo et al.[397] describe the grand canonical MC implementation in CHARMM, which includes cavity-bias[399] and grid-based[400] algorithms for selecting the sites of insertion; Hu et al.[401] calibrate the method to determine the value of μ required to reproduce bulk water densities with the TIP3 model[48,50] and standard nonbonded cutoffs in a periodic system.

In addition to the physically meaningful ensembles described earlier, MC in CHARMM can sample with a number of additional weighting schemes. These include the Tsallis or "generalized" ensemble[402,403] and the multicanonical or constant-entropy ensemble.[404,405] These methods accelerate the exploration of rough energy landscapes by allowing some population of high-energy configurations but still predominantly sample low-energy states, in contrast to simulations at elevated temperatures. In both cases, it is straightforward to reweight the states sampled to recover canonical averages. Multicanonical MC was used by Dinner et al.[165] to interpret fluorescence T-jump experiments for peptide folding; the Wang-Landau generalization of the method,[406,407] which is conceptually similar to adaptive umbrella sampling,[408] is also now available in the MC module of CHARMM.[409]

Move Sets

An MC simulation in CHARMM consists of two phases: the choice of a move set and its subsequent use to generate a trajectory. To optimize flexibility and speed, these two phases are handled separately. Only a small number of commands and atom selections are required to construct a move set because

several predefined types of moves, which can be combined, are provided. Certain types of moves can be used with any of the ensembles: rigid-body translations and rotations of selected sets of atoms and rotations of dihedral angles individually or in concert.[410–413] Some moves (e.g., rigid body translations and rotations) can be linked and applied together.[395] Changes to the system volume[396] and particle number[397] are included for the constant pressure and constant chemical potential methods described earlier. Also, MC can call the leap-frog integrator in CHARMM to generate trial configurations of the system (hybrid MC[414,415]) in simulations that sample states with Boltzmann or Tsallis statistics. A self-guided form of hybrid MC is available[416] (see Section VII.B.).

For each type of move, it is necessary to specify the maximum extent the system can change in one step and the relative frequency of application. The allowed step sizes can be adjusted for individual moves automatically using the acceptance ratio and dynamically optimized MC methods[417] (see Hu et al.[395] for a discussion of their impact on detailed balance). Hu et al.[395] determined the target acceptance rates that yielded the most rapid exploration of configuration space for different types of moves for peptides and found that they ranged from 20 to 95%, in contrast to the conventional belief that 50% yields the most efficient sampling. These authors went on to adjust the frequencies of applying different types of moves with a heuristic MC procedure to obtain peptide move sets that outperformed MD. Comparison of these move sets makes clear that the optimal values of move set parameters differ from one system to another. Hopefully, exploration of MC move sets for other systems at a similar level of detail will lead to "rules of thumb" for different classes of biomolecules.

Monte Carlo Minimization

With the exception of hybrid MC moves, any of the moves described earlier can be followed by minimization prior to application of the acceptance criterion.[418] Although this approach does not satisfy detailed balance, it is useful for applications like structure prediction and ligand design. Either the steepest descents or the conjugate gradient minimization algorithms can be employed. The former is preferable in most circumstances since it is much faster and the primary function of the minimization is to eliminate steric clashes. An alternative implementation that exploits the dihedral angle biasing method of Abagyan and Totrov[419] and allows simulated annealing prior to applying the acceptance criterion is also available in CHARMM (the Monte Carlo Minimization/Annealing or MCMA method.[143,418])

V.E. Grid-Based Searches

As an alternative to the Monte Carlo approach, energy-based searches of conformational space can be carried out in a systematic and/or deterministic manner. Such an approach has proven useful for energy mapping of protein side chain rotational angles and side chain structure prediction,[45,46,420–422] as well as tertiary structure prediction of proteins, given the known secondary structural elements[175] (Petrella, R.J.; in preparation). The Z Module in CHARMM (keyword ZEROM) generalizes this type of approach to facilitate various types of grid-based calculations by partitioning the conformational space into subspaces and systematizing the search. It allows for build-up procedures in which large parts of the system are generated from low-energy conformers of smaller parts, and for the inclusion of statistical information (i.e., rotamer libraries). The Z module has recently been used in molecular docking and loop prediction calculations to predict the structure of the CMV UL44 processivity factor complexed with a DNA oligomer.[423]

VI. Biased Sampling and Free Energy Methods

Thermodynamic and kinetic properties of a system such as free energy differences, reaction paths, and conformational free energy surfaces can be calculated, in principle, from sufficiently long and detailed MD simulations in an appropriate ensemble. In practice, more elaborate schemes, many of which involve nonphysical states of the system, often can be used to reduce the required computational time. Some of the approaches have been used in CHARMM since its inception, while others have been introduced more recently. One important example appears in the methods for calculating free energy differences between different thermodynamic states of a system by simulating nonphysical "alchemical" transformations.[125,424–427] The methods used to perform computational alchemy have a rigorous basis in statistical mechanics, and they represent extremely powerful tools for exploring quantities that correspond to experimental observables, while avoiding the need for prohibitively costly computations. A number of techniques are summarized here; they include free energy simulation methods, simulations in 4D space, multiple copy simulations, and discretized Feynman path integral methods. Umbrella sampling, as used to speed up convergence of estimates and to determine potentials of mean force, and computational methods specifically designed to treat conformational transitions and reaction pathways are described in Section VII.

VI.A. Free Energy Methods

The core of any free energy simulation methodology is a hybrid potential energy function $U(\mathbf{r},\lambda)$, which depends on the so-called coupling parameter, λ. In the simplest case of a linear dependence on λ,

$$U(\mathbf{r},\lambda) = U_0(\mathbf{r}) + (1-\lambda)\, U_{\mathrm{i}}(\mathbf{r}) + \lambda\, U_{\mathrm{f}}(\mathbf{r}) \qquad (10)$$

where $U_0(\mathbf{r})$ is the part of the potential energy that does not change, $U_{\mathrm{i}}(\mathbf{r})$ contains the energy terms unique to the initial state i, and $U_{\mathrm{f}}(\mathbf{r})$ contains the energy terms unique to the final state f. For values of the coupling parameter $0 \leq \lambda \leq 1$, eq. (10) can describe the initial ($\lambda = 0$), final ($\lambda = 1$) and unphysical (alchemical) intermediate states of the system. Because the convergence of the free energy depends on the size of the change between two states, it is generally necessary to proceed in a step-wise fashion from the initial to final systems, by utilizing alchemical intermediate states.

Three different modules, BLOCK,[428] TSM,[429,430] and PERT, which were all introduced *circa* 1986, are available within

CHARMM for performing free energy computations. They make it possible to calculate the free energy difference between two systems having different potential energy functions, U_i and U_f, such as two inhibitors bound to an enzyme active site.[125,424,426,427,431–436] With any of the three methods, free energy differences can be computed by both *thermodynamic integration* (TI)[437] and the *exponential formula*, often also referred to as thermodynamic perturbation (TP).[438] For TI, the (Helmholtz) free energy difference, ΔA, between the initial (i) and final (f) states is given by:

$$\Delta A = A_f - A_i = \int_0^1 d\lambda \left\langle \frac{\partial U(\mathbf{r},\lambda)}{\partial \lambda} \right\rangle_\lambda, \qquad (11)$$

where the $\langle\ \rangle_\lambda$ symbol denotes the ensemble average over the canonical distribution corresponding to λ. For thermodynamic perturbation (i.e., the exponential formula),

$$\Delta A = A_f - A_i = \sum_{i=0}^{n-1} -k_B T \ln \left\langle \exp(-\Delta U_{\lambda_i}/k_B T) \right\rangle_{\lambda_i g} \qquad (12)$$

where $\Delta U(\lambda_i) = U(\lambda_{i+1}) - U(\lambda_i)$ is the energy difference between the perturbed (λ_{i+1}) and unperturbed (λ_i) system at the ith value of λ, n is the total number of sampling windows, $\lambda_0 = 0$, $\lambda_n = 1$, and $\langle\ \rangle_{\lambda_i}$ denotes the ensemble average over the canonical distribution at λ_i. The two approaches are formally equivalent.[7]

TI can be carried out by windowing, i.e., by performing discrete simulations with specified values of λ. The ensemble averages are then calculated for each window and the integration is done numerically, e.g. using the trapezoidal rule. Alternatively, TI can be performed by slow-growth (SG), in which λ is varied gradually over the course of a single simulation.[439] Although the use of SG has been discouraged because of the "Hamiltonian lag" problem,[436] SG-type calculations can be utilized to carry out so-called "fast-growth" simulations in combination with the Jarzynski equality;[440,441] see also later. In both the TI or TP methods, the coupling does not need to be linear. Any smooth functional form in λ can be used, provided λ is varied slowly enough. Nonlinear coupling has been used to overcome the endpoint singularity problem (van der Waals endpoint problem; see later).[442–444]

The entropy and energy contributions to a free energy change can also be determined. One way is to calculate the free energy at several temperatures and evaluate the temperature derivative by finite differences, as in a laboratory experiment.[445,446] An alternative, but related, method is to perform a direct evaluation of the derivatives of the partition function by finite differences in a single simulation.[447] In CHARMM, this is implemented in the TSM module.

Detailed analysis based on statistical mechanics shows that several choices for $U(\mathbf{r},\lambda)$ can be used to compute the free energy difference, leading to a number of different computational schemes for performing free energy simulations.[424,436,448] Although all three free energy modules in CHARMM are based on eq. (10), at least in basic mode of operation, the only formal requirement for the functional form of $U(\mathbf{r},\lambda)$ is that it obey the boundary conditions $U(\mathbf{r},\lambda = 0) = U_i$ and $U(\mathbf{r},\lambda = 1) = U_f$. The different realizations of $U(\mathbf{r},\lambda)$ give rise to the primary differences among the three modules; in particular BLOCK and TSM use a so-called dual-topology approach, and PERT uses a single-topology approach.[448–450]

The BLOCK Module

The BLOCK module[428] provides a general method for scaling energies and forces between selected groups of atoms. Although originally designed to facilitate the computation and analysis of free energy simulations, the same framework can be used in other applications for which systematic manipulation of relative strengths of interactions is required, for example in conjunction with the general REPLICA module (Section VI.C.). It also provides the basis for *λ-dynamics* (see later) and *chaperoned alchemical free energy simulations*.[451]

Since, as mentioned, BLOCK adopts the dual-topology approach, the parts of the system which are not the same in the initial and final state have to be defined simultaneously. The hybrid potential energy function in BLOCK can be written as

$$\begin{aligned} U(\mathbf{r},\lambda) &= U(\mathbf{r_0},\mathbf{r_i},\mathbf{r_f},\lambda) \\ &= U_0(\mathbf{r_0}) + (1-\lambda)\, U_i(\mathbf{r_0},\mathbf{r_i}) + \lambda\, U_f(\mathbf{r_0},\mathbf{r_f}) \end{aligned} \qquad (13)$$

The coordinates $\mathbf{r}_0$, $\mathbf{r}_i$, and $\mathbf{r}_f$, respectively, are associated with the atoms that do not change, those that are present only in the initial state, and those that are present only in the final state. When setting up a free energy simulation using BLOCK, the user first has to assign the atoms in the system into "blocks," according to these three categories. For example, in the simulation of the mutation of a single protein side chain, atoms common to the wild type and mutant might be assigned to block 1 [atom coordinates $\mathbf{r}_0$ in eq. (13)], atoms unique to the wild type to block 2 [atom coordinates $\mathbf{r}_i$ in eq. (13)], and atoms unique to the mutant to block 3 [atom coordinates $\mathbf{r}_f$ in eq. (13)]. Next, the user has to define interaction coefficients to describe the interactions within each block and between each pair of blocks. Through the combination of atom assignments into blocks and the setting of the interaction coefficients, the user realizes the hybrid potential energy function [eq. (13)]. Optionally, specific energy terms can be omitted from this partitioning or scaled differently. This capability is important, for example, in the correct treatment of bonded interactions in alchemical dual-topology free energy simulations.[449,450,452] These scaled interactions (energies and forces, but not second derivatives) are used for subsequent operations, such as energy evaluation, minimization, and MD simulation. In practice, the user carries out a series of simulations at a set of λ values. The trajectories saved during the MD simulations can then be analyzed using special tools provided within the BLOCK facility to extract and average the quantities of interest, e.g., [cf. eq. (13)], $\langle \partial U/\partial\lambda\ \rangle_\lambda = \langle U_f(\mathbf{r}_0,\mathbf{r}_f) - U_i(\mathbf{r}_0,\mathbf{r}_i)\rangle_\lambda$ for TI. This analysis is extremely efficient (only a small fraction of all the interactions in the system need to be evaluated) and can be run repeatedly to obtain component contributions (i.e., estimates of the contribution of different parts of the system) to the free energy change. Near the endpoints (λ = 0 or 1), van der Waals singularities can cause convergence prob-

lems,[453] which can be circumvented with the use of a soft core potential (see later). The BLOCK facility also has built-in functionalities for carrying out slow-growth free energy simulations. Several publications provide illustrative applications of the BLOCK facility.[223,428,449,450] Yang et al. used free energy simulations with BLOCK to develop a detailed mechanism for F1F0-ATP synthase.[14] BLOCK was also used in a study analyzing how DNA repair proteins distinguish the mutagenic lesion 8-oxoguanine from its normal counterpart, guaninine.[454] Because of its generality, the module continues to form the basis for new methodological developments (also, see later).

The TSM Module

The thermodynamic simulation methods (TSM) module[429,430] was developed concurrently with the BLOCK facility to implement TI- and TP-based free energy methods. TSM, like BLOCK, partitions the system into multiple components ("reactants," "products," and the "environment") and permits simulations to be carried out either for a fixed value of λ or in slow-growth mode. Although mostly a dual-topology method, one so-called collocated atom, can be shared between the reactant and product state, conformational free energy surfaces can be constructed within the TSM framework.[430] Applications of the TSM-based methods include protein–ligand,[455,456] protein–DNA[457,458] interaction free energies, and conformational free energies.[430,459]

The PERT Module

The PERT module can be used to calculate alchemical, as well as conformational free energy differences. In contrast to the BLOCK and TSM free energy modules just described, PERT uses a single topology-type hybrid potential energy function $U(\mathbf{r};\ \lambda)$.[448,449] All energy terms, therefore, involve the same coordinate set $\mathbf{r}$; i.e., the energy function has the form of eq. (10), rather than eq. (13). Although the energy in PERT has a linear dependence on λ, in accord with eq. (10), a variant of the method employs a "soft core" potential (see later). In the case of an alchemical free energy mutation in which the number of atoms is not the same in the initial and final states, so-called "dummy" atoms must be introduced.

A PERT calculation is initiated by specifying the part of the system to be subjected to the alchemical mutation. This information is used to construct three nonbonded pair lists: one each for (i) interactions in the unchanged part of the system, (ii) interactions with and within the initial state, and (iii) interactions with and within the final state. The separate lists are needed for efficiency so that nonbonded terms between atoms in the unchanged part of the system are only computed once. Bonded and restraint energy terms, on the other hand, are computed twice, once for the initial state, $U_{0,\mathrm{bonded}}(\mathbf{r}) + U_{\mathrm{i,bonded}}(\mathbf{r})$, and once for the final state, $U_{0,\mathrm{bonded}}(\mathbf{r}) + U_{\mathrm{f,bonded}}(\mathbf{r})$. (The computational overhead of computing $U_{0,\mathrm{bonded}}(\mathbf{r})$ twice is acceptable since calculation of bonded interactions is computationally inexpensive.) The initial PSF, as well as the harmonic, dihedral angle, NOE and general geometric (GEO option of the MMFP module) restraint lists, are saved as the initial state ($\lambda = 0$). The PSF and the three types of restraints can then be modified to effect the alchemical mutation and/or a conformational change leading to the end state ($\lambda = 1$). The command *MKPRes* can be used to automatically generate the PSF patch defining the hybrid residues that are needed for carrying out alchemical free energy simulations. In a procedure that has similarities with both the single- and dual-topology approaches in free energy calculations of mutations, the command defines hybrid residues containing dummy atoms in such a way that all covalent bond contributions are held constant throughout the calculations and only the nonbonded interactions are altered. Use of this command avoids the cumbersome (and error-prone) process of modifying the PSF manually.

When PERT is active, energy calculations, minimizations, normal mode calculations and MD simulations can be carried out for any value of λ, $0 \leq \lambda \leq 1$. In MD one can specify the change of the coupling parameter as a function of simulation length, as well as how many steps are used for (re-) equilibration *versus* accumulation of the respective ensemble averages required for TI and TP. A λ schedule file can be read which allows explicit control of λ windows. This schedule is usually determined from a short exploratory simulation so that the fluctuation of the energy difference in any given window is on the order of $k_B T$. PERT computes the quantities required to compute free energy differences by TI and TP "on the fly," so that in normal usage no postprocessing of trajectories is needed.

PERT includes all contributions resulting from alchemical changes of bonded energy terms.[449,450,452,460] Special attention is required if SHAKE[354] is applied to bonds that have different lengths in the initial and final state. Following an approach outlined by van Gunsteren et al.,[461] constraint free energy contributions are computed using a modified SHAKE routine.[450] PERT runs in parallel and supports SSBP and GSBP, as well as the Ewald-based methods for computing electrostatic interactions. PERT, like BLOCK, can produce an atom-based free energy partitioning that provides useful insights when comparing similar free energy simulations.[462] PERT has also been used in methodological studies focusing on the treatment of bonded interactions in alchemical free energy simulations,[449,450,452] as well as in an analysis of the effect of conformational substates on the precision and accuracy of free energy estimates.[463] In addition, PERT has been employed in several application-oriented studies. A set of optimal atomic radii for PB continuum electrostatics has been developed via a series of charging free energy computations executed with PERT.[192,193] Deng and Roux computed hydration free energies of amino acid side chain analogs.[76] The calculated values are in good agreement with experiment[464] and with the results of a more involved approach.[75] Boresch et al. computed relative solvation free energy differences of phosphophenol derivatives[462]; the results help to explain the binding affinities of the corresponding phosphotyrosine mimetics to protein tyrosine phosphatase and SH2 domains. Several studies using PERT have been carried out to determine absolute binding free energies.[465,76,466–469] The "virtual bond" algorithm introduced by Boresch et al.[466] is an implementation of the double decoupling approach formulated by Gilson et al.[470] whose derivations generalized the restraint potential methods previously introduced to correctly account for the standard state in computing the binding affinity of small molecules for protein

cavities.[465,471,472] Roux and coworkers have studied absolute binding free energies in three proteins, the Src homology 2 domain of human Lck,[467] T4 lysozyme,[469] and FKBP12.[468]

Comparison of Methods

Each of the three modules, BLOCK, TSM, and PERT, has different strengths and weaknesses. This subsection attempts to provide some guidance for users in choosing the one that is the most appropriate tool for a given problem.

An important decision is whether to use a single- (PERT) or a dual-topology (BLOCK, TSM) free energy method. For alchemical mutations of small to medium complexity (e.g., the change of a methyl group into a hydroxyl group), single-topology treatments are relatively direct and can be set up easily. For complicated mutations, particularly those involving changes in connectivity or ring formation, a pure single-topology approach is not possible,[448] and the use of a dual topology method is necessary. The PERT method, while primarily intended for single topology applications, can be used in a dual topology mode with an appropriate set of dummy atoms.[450,460] In applications involving combined QM/MM calculations, dual topology has been favored,[454,473] although single topology calculations using the PERT module are possible for simple alchemical transformations.[474,475,474] TSM can be used to calculate free energy and entropy differences simultaneously. PERT offers the best support for Ewald summation. PERT requires no post-processing, which can have practical advantages in distributed computing environments. On the other hand, BLOCK is a more versatile energy partitioning tool. For example, it is relatively straightforward to use BLOCK to compute free energy differences using Bennett's acceptance ratio method (BAR)[476,477] and generalizations thereof based on Crooks' theorem.[478,479]

Many of the free energy methods in CHARMM have been implemented by modifying the standard CHARMM energy routines, rather than introducing new ones. This approach makes the standard routines more complex, but it facilitates the integration of the new methods with preexisting CHARMM functionality. For example, Ewald summation has recently been introduced in BLOCK (A. van der Vaart, private communication), is partly supported by TSM, and is fully supported by PERT. On the other hand, PERT in some cases requires the generic energy routines, which are not optimized for performance. In addition, the PSSP method (a soft core method; see later) can only be used for selected combinations of nonbonded options. Whether these limitations are relevant depends on the specific requirements of the application.

The Weighted-Histogram Analysis Method

Postprocessing of information from free energy simulations can be used to achieve more precise estimates of free energy changes using the weighted-histogram analysis method (WHAM).[480,481] WHAM minimizes the error in the estimates by finding optimal weighting factors for the combination of simulation data from overlapping windows with an iterative procedure. It makes use of all the available data in the most efficient manner, and can be used to calculate any kind of ensemble average based on the conformations sampled in the simulations[482] including the potential-of-mean-force along coordinates[481,483–488] and free energy differences between different states.[192,489,490]

Soft Core Potentials

In alchemical free energy simulations, the use of a hybrid potential energy function containing a steep repulsive term (e.g. r^{-12} LJ) can result in the "van der Waals endpoint" problem,[453] particularly when the number of atoms changes in the alchemical transformation and the coupling has a simple linear form. Near the endpoints (i.e., at $\lambda = 0$ or 1), extremely large changes in the forces as a function of λ, which arise from the repulsive term, can occur between "overlapping" atoms. Techniques for overcoming this problem include the use of an analytic approximation[453] and the introduction of soft-core (SC) potentials for LJ and electrostatic interactions.[442,443] In the SC method, the distance r between two atoms is replaced by $\sqrt{r^2 + f(\lambda)\delta}$, where δ is an adjustable parameter; for energy terms belonging to the initial state $f(\lambda) = \lambda$, and for energy terms belonging to the final state $f(\lambda) = 1 - \lambda$. Several versions of SC potentials are available for use with the various free energy modules of CHARMM. The SC method of Zacharias et al.[442] is implemented in PERT for LJ and electrostatic interactions in the PERT-separation-shifted-potential (PSSP).[452,491] The PSSP method has been used in calculations of absolute binding free energies.[466] A corresponding method can be used with the BLOCK module.[492] A related SC technique, based on the Weeks-Chandler-Andersen separation[493] of the repulsive and attractive part of the LJ potential, is also available.[76,467–469] Simulations in 4D space can also reduce the endpoint singularity problem in free energy simulations (see later).[443,494]

Free Energy Calculations with λ-Dynamics

A methodology called λ-dynamics has been developed and implemented in CHARMM.[495,496] It extends the free energy perturbation approach by adding multiple variables to control the evolution of interactions; these variables compete to yield the optimal free energy for the conformation and chemical configuration of a group of ligands with a common receptor. The approach builds on ideas put forward by Jorgensen and Ravimohan,[497] Liu and Berne,[498] and Tidor.[499] In λ-dynamics, a hybrid Hamiltonian (potential), somewhat like that in eq. (10) for free energy simulations, is used to effect a change of one set of chemical parameters into another via a pathway that depends on a number of coupling variables, $\{\lambda_i\}$. In this way, the alchemical mapping of one molecule into another differentially scales the components of the solute–solvent interaction terms. One can also consider multiple chemical species, each coupled to a different λ variable as described in eq. (14), or multiple chemical functionalities on a chemical framework. If there are n types of parameters that are transformed in the overall mapping, and if the transformation of each is controlled by one λ variable, i.e., one member of the set $\{\lambda_i\}$, then the mapping between two molecules may be achieved through the definition of a Hamiltonian of the general form

$$H_{\mathrm{Rxn}}(\{\lambda_i\}) = H_{\mathrm{R}}(\{\lambda_i; i = 1, n\}) + H_{\mathrm{P}}(\{\lambda_i; i = 1, n\}) + H_{\mathrm{Env}}, \quad (14)$$

where H_{Env} includes the kinetic and mutual interaction energy of the atoms which are not being transformed (the environment atoms) and $H_{R(P)}(\{\lambda_i;\ i = 1,n\})$ denotes the reactant (product) Hamiltonian composed of three elements: the kinetic energy of the reactant (product) atoms, the self potential energy of the reactant (product) atoms, i.e., the reactant–reactant (product–product) interaction energy, and the potential energy of interaction between the reactant (product) and the environment atoms. $H_{Rxn}(\{\lambda_i\})$ is a valid mapping for use in free energy simulations if the endpoints, where $\{\lambda_i\} = \{0\}$ and $\{\lambda_i\} = \{1\}$, correspond to the Hamiltonians for the reactant and product states, respectively. The elements in the $\{\lambda_i\}$ vector can take on arbitrary and independent values in intermediate regions. To achieve maximum efficiency in sampling in the λ-space, the suggestion of Liu and Berne was followed and an extended Hamiltonian,[331,500] which contains the set $\{\lambda_i\}$ as dynamic variables, is employed in the CHARMM implementation. The coupling between spatial coordinates and energy parameters is through the λ dependence of H_{Rxn}. This Hamiltonian has parallels to that used by the Pettitt group to explore thermodynamics in the "Grand" ensembles.[500] From the extended Hamiltonian, the equations of motion for the extended system are readily derived.[331] An alternative implementation of the extended Hamiltonian method[501] which also uses the lambda parameter as a dynamical variable, relies on TI to obtain the free energy difference. Trial applications indicate that a more rapid convergence is achieved than with the standard TI approach due to dynamic reduction of λ-coupled conformational barriers in the search space.

Other biases can also be included in the extended system description. One key element, which enables rapid screening calculations to be carried out for multiple ligands binding to a common receptor,[495,502,503] is the imposition of a free energy bias corresponding to half of a given thermodynamic cycle; e.g., the solvation free energy for each species can be added to the extended system Hamiltonian. To compute the relative free energy of binding of L ligands to a common receptor, the potential energy is defined as

$$V_{Rxn}(\{\lambda_i\}, X) = \sum_{i=1}^{L} \lambda_i^2 \cdot (V_i(X) - F_i) + V_{Env}(X) \qquad (15)$$

with $\sum_{i=1}^{L} \lambda_i^2 = 1$ where each ligand is biased by a constant free energy term, F_i, that corresponds to the solvation free energy of that ligand, the total extent of the ligand-receptor interactions (present in the terms $V_i(X)$) is normalized to unity, and X denotes the configuration coordinates of the ligands, solvent, and receptor. By carrying out a λ-dynamics simulation of this extended hybrid system and monitoring the probability of each ligand to achieve unit values of λ, the overall free energy change for any pair of ligands is determined from the expression[504]

$$\frac{P^*(\{\lambda_i = 1, \lambda_{k\neq i} = 0\})}{P^*(\{\lambda_j = 1, \lambda_{k\neq j} = 0\})} = \exp\left(-\beta\left[\Delta A_{ij}^{Rec} - \Delta F_{ij}^{Solv}\right]\right) = \exp\left(-\beta\Delta\Delta A_{ij}^{Bind}\right) \qquad (16)$$

where ΔA_{ij}^{Rec} is the free energy difference for the half cycle corresponding to ligands i and j in the receptor binding pocket, $\Delta F_{ij}^{Solv} = F_i - F_j$ is the free energy half cycle corresponding to solvation of the ligands and was input as a bias in the initial calculations, and $\Delta\Delta A_{ij}^{Bind}$ is the overall relative free energy change for the binding competition between ligands i and j.

Some Recent Developments in Free Energy Methodology

Free energy difference calculations, as described earlier, are being more extensively utilized in biomolecular simulations. The required computer time for obtaining converged results is decreasing and the reliability of the results is improving, even as the processes under study become more complex. Some important conceptual/methodological advances have been introduced recently. One new approach, called the MARE method[478,479] is a general method for estimating free energy changes from multistate data (such as those obtained in replica exchange calculations; see also Section VI.B.) by utilizing all of the simulated data simultaneously. As an example, simulations are done with replica exchange for the alchemical transformations of A to A_1, A_2, and A_3. It is shown that including all of the results in the MARE scheme significantly reduces the error of each one relative to that using the data for A to A_1, A to A_2, and A to A_3. Separately, the formulation reduces the statistical error significantly from previous estimators. The MARE approach was motivated by the original Bennett acceptance ratio method,[476,477,505] which makes use of the maximal likelihood evaluation of a free energy perturbation from one state to another. Complementing the MARE method, a λ-WHAM approach has been introduced to refine free energy derivative histograms with the maximum likelihood method; see ref. 506. The efficiency of conformational sampling for problems where the change in the system is local, as in point mutations in proteins or in ligand binding, can be improved by the simulated scaling method[507] and its replica exchange version,[492] in which only the potential energy of the region of interest is scaled. To realize a random walk in scaling-parameter space, the simulated scaling method has been implemented with a Wang–Landau updating scheme and shows rapid convergence of free energy calculations for model systems.[507] An extension of this approach to chaperoned QM and QM/MM free energy simulations[451] has also been implemented.[508] The chaperone method uses a molecular mechanics force field for the quantum region, so that unphysical geometries are prevented in the $\lambda = 0$ and $\lambda = 1$ limits, where the QM terms are small. The methodological improvements that have been described here are examples of an ongoing effort to broaden the range of biophysically important problems to which free energy simulations can be applied.

VI.B. The MMTSB Tool Set

The exploration of the accessible conformational space required for thermodynamic analysis can be enhanced through the use of advanced sampling techniques such as replica-exchange MD.[509] To assist in doing such calculations, as well as those involving a host of related "ensemble" simulation methods, the Multiscale Modeling Tools for Structural Biology (MMTSB) set of perl-based scripts and libraries[510] has been interfaced with CHARMM. This tool set provides a useful complement to CHARMM for the control and manipulation of large-scale calcu-

lations that are distributed over many computers. One key application in this area is replica-exchange MD, which can be performed within CHARMM. In this technique, several replicas of the system of interest are prepared and simulated independently over a range of temperatures (generally exponentially distributed) and then permitted to exchange with neighbors at intervals chosen in accord with the Metropolis criterion. This enhances the conformational diversity of the members of the composite ensemble by allowing low-temperature, potentially trapped, conformations to access higher temperatures, and overcome barriers. The method has been used together with GBMV implicit solvent to analyze nucleoside conformational preferences.[511] Replica-exchange with CHARMM and the MMTSB tool set have been employed in the study of protein and peptide folding, structure prediction and refinement, and membrane-influenced peptide folding, insertion, and assembly.[132,137,229,302,512,513] Figure 5 illustrates two recent examples of the application of replica-exchange sampling with implicit solvent models based on the GB methodology discussed earlier.[514]

VI.C. Enhanced Sampling via Multiple Copy Methods

Multiple copy methods make possible the enhancement of phase space sampling for a subset of variables of interest (e.g., selected amino acid side chains in a protein), in the context of a surrounding set of such variables or bath (e.g., the remainder of the protein). The inspiration for these methods is based on the time-dependent SCF approximation, a mean field approach developed for the study of dynamical properties in electronic structure calculations.[515] The first application of a multi-copy method to biomolecular systems was the locally enhanced sampling (LES) method introduced by Elber and Karplus[516] in a study of ligand diffusion in myoglobin. Trajectories were simultaneously propagated for multiple copies of the ligands, but for only one copy of the protein, so as to greatly reduce the computational cost of the calculation. A similar approach is now commonly employed to determine which chemical functional groups have a favorable interaction with protein binding sites. The multiple copy simultaneous search method (MCSS)[517–519] floods the active site with multiple copies of small chemical fragments and then performs simultaneous energy minimization or quenched dynamics to find local minima for the different ligands on the receptor-ligand interaction potential energy surface. Using a set of ligands allows the generation of functionality maps for the characterization of intrinsic binding site properties; these maps can subsequently be used as the basis for ligand and combinatorial library design.[519–522] Most of the applications have employed a rigid protein model, in which case the multiple copy approach is a book-keeping convenience relative to the execution of multiple, separate runs. However, an extension of the MCSS method allows the use of a flexible protein, in which case a significant sampling efficiency is realized.[518] The MCSS approach has inspired the analogous experimental approaches of Multiple Solvent Crystal Structures[523] and Structure-Activity-Relationships by NMR.[524] A comparison of the experimental and simulation approaches has been described.[525] Because of its widespread utility in pharmacological research, the MCSS methodology is distributed as a separate program which makes use of CHARMM. The multiple copy approach has also been employed in a number of conformational sampling problems such as the optimization of local side chain conformation,[526] and the global prediction of peptide conformation.[323] Attempts to derive thermodynamic properties from multi-copy simulations have been made,[527] and a number of studies have been carried out to address the meaning of the temperature in the simulations and the appropriate treatment of the ensembles involved.[528–531]

The REPLICA Module

Both LES and MCSS can be activated using the *REPLica* command, which is one of the fundamental system generation and modification facilities in CHARMM. The *REPLica* command was originally implemented so as to support a class of methods that seek to improve the conformational sampling of a (usually small) region of the molecular system by selective replication. In principle, its function is to allow the specification of a part or parts of the molecular system through an atom selection, and to generate a specified number of copies (or replicas) of the selected subsystem's attributes (i.e., topological, structural and selected physical properties). Conceptually, each set of replicas constitutes a separate subsystem that is distinct from the primary system. The *REPLica* command can be issued repeatedly to create multiple subsystems. The key effect of the command is in the nonbonded pair list generation routines, which underpin the calculation of the nonbonded interactions in the energy function. Atoms in different replicas within the same subsystem are excluded from the nonbonded pair list and thus do not interact with each other. Replicas in different subsystems do interact, with appropriate mass and interaction scaling as specified using other CHARMM facilities (e.g., BLOCK, Section VI.A., and Section II.C.). Additional functionality has been built upon the REPLICA formalism in CHARMM to support the location of transition states and the estimation of discretized Feynman path integrals (Section VI.D.).

VI.D. Discretized Feynman Path Integrals

Although QM calculations have an essential role in the evaluation of classical semiempirical potential energy surfaces (see Section III.E.) and the study of chemical reactions and catalysis (see Section III.E. and VII.F.), the inclusion of quantum effects can also be important in the calculation of the equilibrium properties and dynamics of a system, particularly at low temperatures, where the effects can be significant.[24,348] Quantum effects on equilibrium properties can be investigated by exploiting the isomorphism of the discretized Feynman path integral (DFPI) representation of the density matrix with an effective classical system obeying Boltzmann statistics.[532] According to this approach, an effective classical system is simulated in which each quantized particle is replaced by a classical ring polymer, or necklace, of P fictitious particles (beads) with a harmonic spring between nearest neighbors along the ring; each bead interacts with two neighbors and the last bead interacts with the first. The spring constant decreases as a function of temperature and mass of the nuclei, giving rise to more extended ring poly-

mers, which correspond to the DFPI manifestation of familiar quantum effects, such as zero-point vibration and tunneling. MD or Monte Carlo simulations of the effective classical system (in which some or all the particles are described by isomorphic ring polymers) are valid for obtaining ensemble averages, although they do not provide information on the time-dependent quantum dynamics of the system.

In the current CHARMM implementation of DFPI, each quantum atom is represented by the same number of beads.[533] The creation of the beads utilizes the REPLICA facility described earlier. The energy of the ring polymers is a sum of harmonic terms between consecutive beads along the necklace with spring constant $K_{DFI} = Pk_BT/\Lambda^2$, where Λ is the de Broglie thermal wavelength of the quantum particle $\Lambda = (h/2\pi)^2/(mk_BT)$. These interactions are added to the CHARMM energy through the command *PINT*. The interaction with other atoms is introduced by means of the classical CHARMM potential energy function scaled by $1/P$; each bead interacts only with one bead in other quantum atoms, and there is no interaction between beads belonging to the same necklace, except for the spring interaction within the necklace. The attribution and scaling of the different interactions is specified with the *BLOCk* command.[533]

VI.E. Simulation in 4D Space

The addition of a nonphysical fourth spatial dimension to molecular mechanics can increase the efficiency of sampling conformational space.[352] Enhanced sampling of conformations is achieved because barriers in the physical (3D) space can be circumvented by introducing the higher dimensionality of four spatial dimensions. Energy and forces are computed in 4D by adding a fourth value, w, to the atomic coordinates (x,y,z); in CHARMM, this is done through the use of the VER4 dynamics integrator (see also Section V.B.). After initial assignment of the 4D coordinates and velocities, a harmonic energy term allows control of the embedding of the system in the fourth dimension; an increase in the associated force constant of this term leads to smaller w values, thereby projecting the system into 3D space. MD in four dimensions has been applied to problems related to protein structure determination[534,535] and free energy calculations.[443,494] MD in 4D space searches a large enough conformational radius to allow the use of random-coil configurations for initial coordinates.[536] The use of a fourth spatial dimension has been shown to be advantageous for calculating free energies of solvation and of ligand binding affinity whereby the solute nonbonded interactions are coupled to the system through w, and a PMF (4D-PMF) is calculated by umbrella sampling over the range $w = 0$ to $w = 1$ corresponding to the reversible abstraction of the solute from the solvent or binding site.[494,537] In these studies, the approach resulted in accurate solvation free energy estimates, and converged efficiently without the van der Waals endpoint problems experienced with λ-scaling of nonbonded interactions (see Section VI.A.). The 4D-PMF method is simple to implement because it is easily generalized to all LJ and Coulombic nonbonded interactions.

VII. Reaction Paths, Energy, and Free Energy Profiles

An important problem in molecular modeling is the determination of the minimum energy or free energy pathway and the transition rate between two different conformations. Many biomolecular processes involve large-scale conformational changes in the structure of the system.[13,300,538,539] Often the transition is a rare event, occurring on a timescale well beyond the reach of conventional MD (on the order of 100 ns or longer for large systems). Consequently, specialized approaches must be used to observe such transitions in simulation.

Several simulation methods have been developed to determine minimum energy and free energy pathways on multidimensional potential surfaces of complex biomolecules. These methods vary in the details of the path sampling procedures they employ, whether they use reaction coordinates, and, for those that do, the types of reaction coordinates for which they are best suited. Reaction coordinates are the degrees of freedom, or functions thereof, by which the pathway is defined. For many calculations, they are a small number (one to three) of geometric parameters (e.g., RMSD between initial and final states, certain bond angles), but can include order parameters of any type (e.g., fraction of native contacts, number of hydrogen bonds) or number. The term "reaction path," which originated in the study of chemical reactions, is now used more generally to refer to the pathway of a molecule between two end states in conformational or chemical space. Both the minimum energy path (MEP), which provides the energy, and the PMF along a path, which provides the free energy, can be calculated with CHARMM.

The MEP is the path on the potential surface that connects the reactant state to the product state (or two intermediate states if there is a multibarrier transition) by steepest descent from the barrier, or saddle-point, which is the stationary point where the Hessian matrix has a single negative eigenvalue. MEPs provide a useful description if the free energy along the path is dominated by the enthalpy; changes in the vibrational entropy along the path to obtain the free energy can be included *a posteriori*.[540] For processes involving important changes in conformational entropy, the MEP can provide a curvilinear reaction coordinate along which the PMF can be computed.[48] A chain-based method (i.e., one that optimizes the entire path simultaneously) was originally developed by Elber and Karplus[541]; a refinement of the method is referred to as the "self-penalty walk method"[542] and the Replica Path method in CHARMM is based upon it and the REPLICA code. Several other chain-based MEP methods have been developed subsequently—e.g., the Nudged Elastic Band (NEB) method[543,544] and the Zero-Temperature String (ZTS) method.[544–546] All of these methods find a locally optimized path, which is not necessarily the global optimum path; this is a general problem with optimization methods for complex systems. Existing MEP calculation methods include automatic search methods for improving pathway exploration and the location of the globally best path.[547]

Under physiological conditions, molecules can cross low-energy barriers, and more than one transition path can contribute significantly to the transition rate.[166] Hence, a related problem is finding an ensemble of paths or the best average (minimum free

energy) path at non-zero temperatures. One approach makes use of nonequilibrium methods available in CHARMM. It requires that stable states of the reaction are known from experiment, and that suitable order parameters that characterize these states and the distance of a conformation from them can be defined. In such cases, insights into the reaction path can be gained from multiple trajectories generated with targeted or steered MD approaches.[142,548–553] The various methods differ with regard to the form of the bias, which can be either a holonomic constraint or a restraining term added to the energy function, and the schedule with which it is advanced. As a rule, methods that advance the bias more slowly and apply smaller biasing forces are less likely to give rise to dynamic artifacts.[401] Self-guided stochastic methods[416,554] can be useful for exploring the available free energy basins and the paths connecting them in cases where the final state is not known.

The PMF along some chosen reaction coordinate plays a central role in modern transition state theory and its generalization to many-body systems.[555] It can be used to evaluate a transition rate, the dynamical prefactor, and the transmission coefficient. Special biased sampling techniques can be used to calculate these quantities from an MD trajectory. In particular, the PMF can be calculated using the free energy perturbation technique[438] (see Section VI.A.), the umbrella sampling technique (see Section VII.C.),[556] or the Jarzynski equality.[440]

The transmission coefficient can be calculated using the activated dynamics procedure[555,557]; an early example of its application to a biologically interesting system is given in Northrup et al.[296] Alternatively, it is possible to estimate the transmission coefficient in the diffusive limit using an analysis based on the Generalized Langevin Equation.[558–560] More generally, transition path sampling (TPS) methods[395,561–563,401] sample the dynamics of a system without bias but require harvesting many trajectories of lengths comparable to the time it takes for the system to relax from the transition state to a stable state (the "commitment time").

The fundamental importance of determining chemical and physical reaction mechanisms has naturally led to the introduction of many methods for finding reaction paths, as is made clear by the discussion in this section. In general, there is a tradeoff between the computational resources required by methods and the accuracy of the description that they provide. Thus the choice of method depends on the system of interest and the goals of the investigator. In all of the reaction path methods, care must be taken in the labeling of chemically equivalent atoms (e.g. the two δ position atoms or the two ε position atoms in a benzyl ring) in all of the copies, so as to avoid introducing artifactual dihedral angle rotations into the path.[564] This problem often arises when the starting or end structures in a calculation are derived from separate sets of X-ray crystallography data. A facility which relabels chemically equivalent atoms in two structures according to RMSD criteria has recently been developed and will be available in future versions of CHARMM.

VII.A. Chain-Based Path Optimization

The search for a reaction path and the corresponding transition-state(s) is not straightforward if more than a few degrees of freedom are involved. Methods that drive the system along a 1D reaction coordinate (e.g., a torsion angle or the RMS deviation from the product), such as adiabatic minimization with a restraint or targeted MD (see Section VII.D. later), are straightforward to apply. However, finding the appropriate reaction coordinate(s) to describe the transition can be difficult, even in apparently simple reactions. For example, in the *cis–trans* isomerization of the proline peptide bond, the standard backbone torsion angle ω was shown to be inappropriate as a reaction coordinate.[565] An alternative to using a predefined reaction coordinate is to obtain the MEP by optimizing the entire path as described by a chain of conformers. This approach requires an initial guess for the path, which can be as simple as the linear interpolation between the end-states. It is also possible to include in the initial guess a set of predetermined intermediate structures, which are then optimized with the rest of the path. The following three methods in CHARMM use the chain-based path optimization approach.

Replica Path Methods

In the original chain-based optimization method of Elber and Karplus,[541] an initial guess for the path can be provided by a linear interpolation between end states, such that the coordinates of the jth point, R_j, along the path are given by $R_j = R_0 + j\Delta R$, where $\Delta R = (R_0 - R_{M+1})/(M + 1)$, R_0, and R_{M+1} are the coordinates of the fixed endpoints, and M is the number of free path points. A first-order minimization method, the Powell algorithm, is then used to minimize a functional of the form

$$T(R_0, R_{M+1})_L = \frac{1}{L}\sum_{j=1}^{M} V(R_j)\Delta l_j + \lambda \sum_{j=0}^{M} (\Delta l_j - \langle \Delta l \rangle_{\mathrm{rms}})^2 + \lambda' \sum_{j=0}^{M} \Delta t_j^2, \qquad (17)$$

where $V(R_j)$ is the potential energy of the system at path point j, L is the length of the entire path, Δl_j is the length of path segment j (distance between path points j and $j + 1$), $\langle \Delta l \rangle_{\mathrm{rms}}$ is the RMS path segment length, Δt_j^2 is a measure of the rotation and translation of the coordinates of path point j relative to its coordinates at the start of the calculation, and λ and λ' are parameters. Hence, the potential energy of the entire path is minimized while the path segment lengths (second term) and the global rotation and translation of each path point (third term) are restrained. In the self-penalty walk method,[542,566] rigid rotation/translation is constrained by a different method and an additional restraint term is added that is of the form $\rho \sum_{i>j+1}^{M+1} \exp(-r_{i,j}^2/(\lambda'' \langle r \rangle_{\mathrm{rms}})^2)$, where r_{ij} is the distance between two path points, $\langle r \rangle_{\mathrm{rms}}$ is the RMS distance between sequential points, and ρ and λ'' are parameters. This "repulsion" term prevents the path from revisiting the same regions of conformational space. Many current reaction path methods are derivatives of this "self-avoiding" or self-penalty walk method. Methods of this type eliminate the expensive analytic Hessian computation required for the Intrinsic Reaction Coordinate (IRC) method,[567] which is generally used in QM studies of small molecules. Since the self-penalty walk methods use a differentiable target function, they are well suited

for searching and improving paths using high-temperature annealing or self-guided Langevin dynamics[554,568] for the exploration of the conformational space.

The replica path method[289,569] is similar in spirit to the self-penalty walk method, but it utilizes the REPLICA functionality in CHARMM (Section VI.C.) to construct a trial reaction path by replicating the part of the molecule that is involved in the conformational change. This feature allows a partitioning of the system into replicated atoms that are directly involved in the pathway and environment atoms whose positions are the same for all replicas. The method restrains each replica with a penalty function that uses best-fit RMS distances to the two adjacent replicas, thereby circumventing the need for restraining the rotation and translation of the replicas. A restraint on the pathway curvature using the RMSBFD metric is included, in lieu of a temperature-related term used in some other chain-of-states methods, to smooth the pathway and keep it from folding back on itself. For each path point (replica), i, this restraint term involves the angle, α_i, between i, $i + 1$, and $i + 2$; the term is of the form $E_{\text{ang}} = \sum_{i=1}^{m} K_{\text{ang}}(C_{\max} - \cos(\theta_i))^2/2$, where $\theta_i = 180 - \alpha_i$, $C_{\max}$ is the cosine of the angular deviation from linearity above which the restraint is applied, K_{ang} is the force constant determining the stiffness of the path and m is the number of path points. Customized specification of atomic weighting factors can be also used in the RMSBFD calculation to vary the degree of participation of a given atom in the conformational change metric. Atoms selected with zero weight contribute to the energy in the path calculation, but their displacement is not included as part of the path and they are not used in the application of the restraints.

The replica path method in CHARMM can be used with both classical and hybrid QM/MM Hamiltonians. Several QM packages may be used in a parallel scheme (i.e., parallel QM/parallel MM) that can efficiently use hundreds of processors: GAMESS-UK,[266,285,570] GAMESS-US,[286,287] and Q-Chem.[294,295] Parallel efficiency is achieved by computing the quantum energy of each replica in parallel on a different set of processors.[289,569] For single-processor calculations, the SCC-DFTB package can also be used.[571] The QM/MM replica path method is an effective tool for obtaining approximate minimum energy reference pathways. These are obtained either by minimization, or by calculating an average structure for each replica from a Langevin dynamics simulation and then optionally smoothing. The smoothed path is useful for subsequent PMF simulations by umbrella sampling.

A potential problem that can arise with the use of MEP methods for the study of large systems is that there can be "uncorrelated" fluctuations in the total energies due to system motions that are unrelated to the pathway of interest (e.g., the rotation of a water molecule that changes the total energy by several kcal/mol). The replica path method, as well the REPLICA-based NEB method described next, mitigate this problem by treating the environment consistently over the course of the entire path, allowing all replicas to see the same environment. However, the total energy over an optimized zero-temperature path generated with these methods may still be subject to uncorrelated fluctuations when the replicated portion, itself, is large. In these cases, the calculation of the approximate work done over the 0K path can yield meaningful results. The forces from the entire replicated region and environment are included in the work term, but because only their projections along the path contribute, the effect of uncorrelated motions in the distant parts of the replicated regions is diminished. The "0K work" term has been shown to converge to the system energies in the chorismate mutase reaction path for a small replicated region (6 Å in radius),[289,569] For cases in which the replicated region is larger and in which the 0K work term and the system energies do not agree, the former is the more meaningful and reproducible quantity. The off-path simulation method (Woodcock H. L., et al.; in preparation) extends this idea to the computation of PMFs by utilizing a fixed reference pathway and RMSBFD restraints to define an umbrella potential and allow free motion in planes orthogonal to the pathway. These planes can be thought of as having an approximately constant value for the commitment probability. The force vectors resulting from a simulation using these restraints, along with the corresponding distance vectors, are rotated into the frame of the reference pathway for each segment of the path, yielding an average work term, which may be partially curvature corrected.

NEB Methods

The NEB method[543] is another chain-of-states method that is implemented in two different forms as part of the replica path code in CHARMM. The NEB method determines MEPs that are locally exact, given the approximation of using a finite (usually small) set of replicas. The forces acting on each replica are given by

$$\vec{F}_i = -\nabla V(\vec{R}_i)|_{\perp} + (\vec{F}_i^{S} \cdot \hat{\tau}_{||})\hat{\tau}_{||} \tag{18}$$

where $V(R_i)$ is the potential acting on the ith replica, $\hat{\tau}_{||}$ is the pathway tangent vector, $\nabla V(\vec{R}_i)|_{\perp} = \nabla V(\vec{R}_i) - (\nabla V(\vec{R}_i) \cdot \hat{\tau}_{||})\hat{\tau}_{||}$ is the projection of the perpendicular component of $\Delta V(R_i)$ and $(\vec{F}_i^{S} \cdot \hat{\tau}_{||})\hat{\tau}_{||}$ is the parallel component of the spring force introduced to keep the replicas equally spaced along the chain. The two forms of the method implemented in CHARMM differ in the definitions of the spring force and the tangent vector. In addition, one uses RMS distances to calculate pathway step lengths and angles,[572] and the other uses root-mean-square best-fit distance (RMSBFD) values.[297]

In CHARMM, a minimization scheme with superlinear convergence properties has been developed and implemented for the NEB method.[297] The algorithm is based on the adopted basis Newton–Raphson (ABNR) method. During the minimization, each ABNR step is performed self-consistently in a user-defined subspace. The superlinear minimization scheme of NEB has been shown to be more efficient than quenched MD minimization or steepest descent minimization.[297] In addition, the CHARMM implementation of the NEB method is also able to take advantage of the RMSBFD pathway definitions (see Section VII.A.) and to employ flexible weighting options. Also, because the NEB implementation is coupled to the REPLICA code, the parallel/parallel QM/MM pathway functionality in CHARMM can be used to examine bond-forming and bond-breaking processes. In addition to the standard NEB method, CHARMM also

supports the climbing image NEB (CI-NEB).[573] In this method, which is a modification of the original NEB, one of the images is moved to the highest energy saddle point along the path. The CI-NEB is robust with respect to the discretization of the pathway and returns an accurate estimate of the transition state energy. Use of the CI-NEB method following a standard replica path or NEB pathway calculation can save significant computer time when the focus is on transition state properties.

Another chain-of-states method is the recently developed string method[544–546] and its implementation using swarms-of-trajectories.[574] It is similar in spirit to the NEB method, but the replicas are independent during dynamics and minimization (no interreplica restraints), and they are repositioned along the interpolated path after every global iteration. Thus, the string method is, in principle, somewhat simpler to implement and parallelize than NEB. Moreover, the finite temperature string method, unlike NEB, permits the calculation of free energy surfaces. Application has been made to the solvated alanine dipeptide.[575]

Conjugate Peak Refinement (CPR) Method

Another algorithm for finding the MEP is CPR,[576] which is implemented in the TREK module (keyword TRAVEL) of CHARMM. Starting from an initial path, CPR finds a series of structures that closely follow the valleys of the energy surface and determines all saddle points along the path. Unlike the replica path and NEB methods, the CPR algorithm does not utilize the REPLICA functionality in CHARMM. Instead, the method replicates the system internally, and environment atoms can be fixed to reduce the degrees of freedom in the problem. CPR is capable of determining the relevant saddle-point(s) along transition pathways that involve tens of thousands of degrees of freedom. The principle of CPR is to focus the computational work on improving the high-energy segments of the path. An iterative procedure is used, and in each cycle the highest local energy maximum along the path (called the "peak") is found and the path is rebuilt so that the new path circumvents the high-energy region around the peak. This is done by improving, removing or inserting one path-point. Points that are inserted or improved are optimized by a controlled conjugate gradient minimization, which prevents each point from falling into an adjacent minimum and which converges to the saddle-point if the peak was located in a saddle region of the energy surface (i.e., the path was crossing over a barrier). The path refinement is finished when the only remaining energy-peaks along the path are true saddle-points. Because the number of path-points is allowed to vary during the refinement, and no constraints are applied on the path shape, any degree of complexity of the underlying energy surface can be accommodated. The details of this heuristic algorithm are described in Fischer and Karplus[576] and in the CHARMM documentation. Since the parameters of the algorithm are independent of molecular size or the nature of the reaction, they do not need to be reoptimized for new reactions. Thorough minimization of the structures is required. Also, to be compatible with CPR, a potential energy function must have analytic and finite-difference derivatives which correspond (i.e., must pass *TEST FIRSt*; see Section XI.B.). CPR is parallelized and works in combination with QM/MM implementations and with most GB-related continuum solvation methods. For the purpose of energetic analysis or subsequent PMF calculations along the MEP,[48] the resulting CPR path can be effectively smoothed with the NEB method (see earlier) or with the Synchronous Chain Minimization (SCM) method. In SCM, all path points are simultaneously energy-minimized under the constraint that each point must remain on the hyper-plane that bisects its two adjacent path-segments; these planes are periodically updated as the path evolves. To prevent kinks in the path and the descent of path-points into nearby minima, SCM controls the change in the angle between adjacent path segments during the minimizations. SCM is implemented in the TREK module of CHARMM.

Problems to which the CPR algorithm has been applied include: (1) enzymatic catalysis, where the end-states of the substrate can be either conformational isomers (e.g., the rotamase FKBP[577]) or chemically different species (e.g., proton transfer in Triosephosphate-isomerase[254]); (2) the study of membrane channel permeation, where the substrate in the two end-states can be placed on either side of the membrane (e.g., sugar-chain translocation across maltoporin[578]); (3) ligand entry paths into buried binding sites, which can be explored by using reactant states where the ligand is placed in various locations on the protein surface (e.g., retinoic acid escape[36]); and (4) pathways for large-scale conformational change between different crystal structures of proteins.[579] The robustness of the CPR method allows it to be used in automatically mapping the connectivity of complex energy surfaces and, with graph-theoretical best-path searching algorithms, in identifying the globally lowest path in a dense network of subtransitions.[547] CHARMM scripts enabling this functionality can be found in the "support" directory.

VII.B. Nonequilibrium Trajectory Methods

Several methods for determining a reaction path between a product and a reactant follow the nonequilibrium trajectory of the system starting in the reactant basin while a biasing potential is applied to drive the system towards the product basin. In most cases, the trajectories generated according to such a scenario are irreversible; i.e., the system does not necessarily return to the initial state if the biasing potential is turned off because barriers along the pathway are usually present in both directions. The resulting trajectories are generally found to provide useful insights concerning the character of the transition pathway. Moreover, once a pathway has been calculated, it is possible to determine the free energy associated with it by umbrella sampling or alternative methods.[48] Also, in some cases the underlying equilibrium PMF can be calculated via the nonequilibrium approach due to Jarzynski,[440] though accurate estimates are difficult to achieve.[580] A number of such nonequilibrium methods are supported in CHARMM. They are targeted molecular dynamics (TMD),[548] self-guided Langevin dynamics (SGLD),[554,568] steered molecular dynamics (SMD),[401,549–551] and the half–quadratic biased MD (HQBMD) method.[142] In addition to these specialized nonequilibrium methods, CHARMM provides a number of general potential energy restraints (described in Section III.F.), along with a dedicated restraint facility called RXNCOR, that can be used to control the progress of a trajectory.

Targeted Molecular Dynamics

In 1993, a constrained dynamics method called TMD was developed to simulate the pathways of conformational transitions of biomolecular structures that occur on time scales much longer than are accessible in conventional MD simulations.[548] If the atomic structures of two conformations of a protein are known, this method can be used to identify a transition pathway from a starting conformer to the target conformer by applying a single time-dependent holonomic constraint based on the (mass-weighted) RMSD between the two conformers. The general form of the constraint is

$$\Phi(\vec{X},t)=\left(\sum_{i=1}^{N} m_i|\vec{x}_i(t)-\vec{x}_{i,F}|^2/\sum_{i=1}^{N} m_i\right)-\eta^2(t)=0 \qquad (19)$$

where N is the number of atoms in the system, $\vec{x}_{i,F}$ is the position of atom i in the target conformer, $\vec{x}_i(t)$ is the position of atom i at time t, $\eta(t)$ is the desired mass-weighted RMSD between the system and the target structure at time t, m_i is the mass of atom i, and $\vec{X}=\{\vec{x}_1,\vec{x}_2,\ldots,\vec{x}_N\}$. At each step of the MD simulation, the system is first allowed to evolve according the physical (unperturbed) potential energy function. The constraint forces, $\vec{F}_i^c=\partial\Phi/\partial\vec{x}_i$, then perturb the structure so as to satisfy eq. (19); for each atom, the force is proportional to the difference between the atom's coordinates in the current and target structures—i.e., $\vec{F}_i^c(t)\propto(\vec{x}_i(t)-\vec{x}_{i,F})$. Application of the constraint (with the mass weightings) conserves the position of the center of mass of the system, provided that the centers of mass of the current and target conformers are the same and all of the atoms are included in the constraint. Although the method imposes no *a priori* restrictions on the time-dependence of the constraint parameter $\eta(t)$, which controls the rate of convergence of the initial conformer to the target, the parameter is commonly made to decrease linearly with time (but see RP-TMD below), until it reaches a user-defined tolerance. As an alternative to this type of holonomic constraint, a harmonic restraint can be used in TMD.[552]

In CHARMM, the TMD constraint can be based on all atoms or a chosen subset of atoms (second atom selection in the *TMD* command); the remaining degrees of freedom in the system are allowed to relax according to the physical potential energy surface throughout the simulation. If the atom selection (typically, the protein mainchain atoms) does not include all the atoms in the system, application of the constraint does not in general preserve the center-of-mass of the system. As the holonomic constraint employed in TMD does not conserve angular momentum, the target structure can be superimposed onto the simulated structure by a least-squares fit at a user-specified frequency (by use of the *INRT* option and the first atom selection in the *TMD* command) so as to remove overall rotation. The TMD constraint can be used in conjunction with other CHARMM constraints such as SHAKE, which fixes bond lengths. As with other methods that introduce external forces, the use of Langevin dynamics is recommended with this method to control the temperature so as to obtain smooth trajectories. TMD permits simulations to be performed at any desired temperature; this is an advantage in the study of biomolecules and other systems with significant entropic contributions, since pathways generated at ambient temperature are often more realistic than the minimum-energy pathway. The TMD method in CHARMM has been widely used. An example is the determination of the reaction paths for the transition between the GTP-bound and GDP-bound conformations of the molecular switch I and II regions of oncogene protein $p21^{ras}$,[581] which recognize distinct sets of partner proteins on the cell signal transduction pathway.[582] An interaction that occurs along the pathway and not in the end states was identified by the simulations and subsequently verified by experiment.[583,584] The TMD method, which is particularly suited to model large-scale motions, has also been used to determine the transition pathways for the rigid-body-like domain motions of GroEL[585,586] and F_1-ATP synthase.[124]

Two variants of the TMD method are implemented in CHARMM, ζ-TMD, and RP-TMD. In the ζ-TMD method, the constraint is a function of both the initial and final structures, rather than just the latter. The form of the constraint is: $\zeta(t)-\zeta_0(t)\le\zeta_{tol}$, where

$$\zeta(t)=\frac{-1}{1+e^{-C_\zeta R_1(t)}}+\frac{1}{1+e^{-C_\zeta R_2(t)}}, \qquad (20)$$

ζ_{tol} is a tolerance, $\zeta_0(t)$ is the desired value of the restraint at time t, C_ζ is a constant, and $R_1(t)$ and $R_2(t)$ are the RMS deviations from the two target structures. This form of the TMD method is especially useful when the current structure is distant from either target or when the desired path does not involve a monotonic decrease in the RMSD from one target. The second variant is the restricted perturbation TMD method (RP-TMD),[553] which limits either the sum of the atomic perturbations or the maximal atomic perturbation at each step of the dynamics trajectory. It is designed to prevent large barrier crossings, so that the resulting paths can be closer to the actual PMF path than those obtained in the other TMD formulations.

A useful approach for simulations of biomolecules is to start with TMD or related methods with a large constraint that provides a path between the end states, and to gradually reduce the constraint so that the resulting paths approach the true path in the absence of constraint.[401]

The Half Quadratic Biased Molecular Dynamics (HQBMD) Method

HQBMD is a method that forces a macromolecule to move between states characterized by the value of a reaction coordinate, which changes with time along the trajectory. The method is related to the minimum biasing technique introduced by Harvey and Gabb[587] and has been applied to simulate stretch-induced protein unfolding,[142,170] the denaturation of a protein *in vacuo*[588] and in implicit solvent,[589] and the unbinding process for a hapten-antibody complex.[167] The perturbation is a half-quadratic potential that depends on time through a reaction coordinate ρ, which is a function of all or a subset of the Cartesian coordinates of the system. The perturbation has the form

$$W(r,t) = \begin{cases} \dfrac{\alpha(\rho(t) - \rho_a(t))^2}{2}, & \rho(t) < \rho_a(t) \\ 0, & \rho(t) \geq \rho_a(t) \end{cases} \quad (21)$$

where $\rho_a(t) = \max_{0 \leq \tau \leq t} \rho(\tau)$.

The minimum of the half quadratic perturbation "moves" as the reaction proceeds (i.e., as the reaction coordinate ρ increases). The reaction coordinate ρ is chosen in accord with the problem being studied. One such coordinate currently implemented in CHARMM is

$$\rho(t) = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j>i}^{N} (r_{ij}(t) - r_{ij}^F)^2. \quad (22)$$

This coordinate corresponds to the mean-square distance deviation from a reference conformation (F) of a set of N atoms that is considered sufficient to specify the conformation of the object system being studied; $r_{ij}(t)$ is the instantaneous distance between sites i and j, and r_{ij}^F is the distance between the same pair of sites in the reference structure (F).

If the coordinates of the reference conformation are all set to zero, $\rho(t)$ in eq. (22) (i.e., the average squared interparticle distance) is proportional to both the radius of gyration (R_g) squared and the variance of the position vectors.[‡318] Several other reaction coordinates can be chosen within the HQBMD module. Among these are reaction coordinates which measure the deviation from experimentally measured "phi" values, a name introduced for the effects of mutations on the stability of protein folding transition states,[590–592] and hydrogen exchange protection factors.[591,592] Both are assumed to be related to the number of native contacts or hydrogen bonds, or the deviation from measured NOEs and scalar dipolar couplings. Such biases have been used to sample slow native fluctuations and non-native states which are difficult to characterize by other means.

In an HQBMD calculation, the simulation is started at $t = 0$ with the value of $\rho_a(0)$ set equal to $\rho(0)$, the value of the reaction coordinate for the equilibrated starting configuration. If the reaction coordinate spontaneously increases in the simulation step from t to $t + \Delta t$, i.e., $\rho(t + \Delta t) > \rho_a(t)$, the external perturbation is zero and has no effect on the dynamics. In such a case, $\rho_a(t)$ is updated and $W(r,t)$ is modified accordingly, i.e., $\rho_a(t)$ is set equal to $\rho(t + \Delta t)$. If $\rho(t)$ is smaller than ρ_a, the harmonic force acts on the system to prevent the reaction coordinate from decreasing significantly. The value of α determines the magnitude of the allowed backward fluctuation of the reaction coordinate and modulates the time scale of the reaction. The macroscopic state of the system is never changed since the perturbation is added to the Hamiltonian of the unperturbed system when it is numerically zero. Nevertheless, the perturbation affects the system working like a "ratchet and pawl" device[593] that "selects" the sign of the spontaneous fluctuations biasing the trajectories toward the desired state. If the effective free energy surface is such that the motion of the reaction coordinate is diffusive in the absence of a barrier, the temperature of the system is not expected to change during the conformational transition. However, if there is a free energy barrier along the reaction path, the effect of the directed motion induced by the perturbation is to transform some of the kinetic energy associated with the reaction coordinate into potential energy. To avoid possible artifacts from temperature variation of this type, the simulations should be performed in the presence of a thermal bath using, e.g., Nosé–Hoover, or Langevin dynamics. The HQBMD method allows one to sample regions of the configurational space that are separated by either thermodynamic or kinetic (on a simulation time scale) barriers and determine low energy pathways. Other techniques, such as umbrella sampling, can be used to estimate the free energy profile along these pathways. For comparative purposes all the reaction coordinates available in the HQBMD module can also be manipulated by means of a harmonic potential whose minimum is displaced at constant velocity, in accord with a number of AFM experiments; this method is referred to as SMD.[142,549,551,594]

The AFM Method

The implementation of the AFM method in CHARMM has been motivated by single-molecule experimental techniques, which offer a new perspective on molecular properties.[595,596] Such experimental techniques can be simulated in CHARMM by, for example, using AFM SMD to mimic the effect of a cantilever moving at constant speed, or by applying the biased MD approach described earlier (AFM BMD) or a constant force (CF) to mimic a force-clamp experiment. Alternatively, a force (constant or periodically varying in time) can be applied to selected atoms in a specified direction (*PULL* command). The PULL force vector can be specified directly; alternatively, it can be specified indirectly in terms of an electric field, **E**, which gives a force, q**E**, acting on an atom with charge q.

Self-Guided Stochastic Methods

To enhance searching efficiency and facilitate the study of conformational changes in which the final state is not known, two self-guided stochastic simulation methods are available in CHARMM: momentum-enhanced hybrid Monte Carlo (MEHMC)[416] and self-guided Langevin dynamics (SGLD).[554] These approaches address several problems[416,597] inherent in the earlier self-guided molecular dynamics (SGMD) algorithm that motivated them.[568] They are much more robust than SGMD because they balance the use of information about the average motion from previous steps in the simulation with appropriate forms of dissipation.[416] As a result, MEHMC and SGLD can enhance the conformational search efficiency by accelerating the motion of the system without significantly altering the ensemble of conformations explored. Two parameters are used to control an MEHMC or SGLD simulation. One is the local averaging time, which defines the slow motions that are to be enhanced. The other is the guiding factor, which controls the degree of enhancement. The application of these methods in peptide folding simulations[598] and in the exact calculation of thermo-

[‡]Specifically, $\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j>i}^{N} r_{ij}^2 = \frac{1}{N}\sum_{i=1}^{N}(\vec{x}_i - \langle\vec{x}\rangle)^2 = R_g^2 = \langle\vec{x}^2\rangle - \langle\vec{x}\rangle^2 = Var(\vec{x})$, where $\vec{x}_i$ is the position vector of atom i, $\langle\vec{x}\rangle$ is the mean position vector (center of geometry), and $\langle\vec{x}^2\rangle$ is the mean squared position vector. The double sum over squared interparticle distances is therefore expressible exactly as functionals of single sums.

dynamic[416] and kinetic[599] observables has shown promising enhancements in conformational search efficiency.

VII.C. Potentials of Mean Force and Umbrella Sampling

MD simulations produce a series of states whose equilibrium and kinetic properties can be estimated. However, sampling the conformational changes involved in very slow processes by brute force simulations may be impractical. One way to improve sampling is by the introduction of systematic biases along one or more appropriately chosen reaction coordinates that describe the progress of the conformational change.[556] Several of the general restraints in CHARMM (see Section III.F.) can be used to introduce such a bias, but CHARMM also provides the dedicated reaction coordinate facility RXNCOR and the adaptive umbrella sampling module (ADUMB) to support biased simulations. The RXNCOR module[600] applies biasing energy restraints along a chosen reaction coordinate. A general framework is provided to define the reaction coordinates as a function of appropriately chosen degrees of freedom of the molecular system. To analyze the biased simulations, the PMF of the reaction coordinate and the value of the reaction coordinate versus time can be printed out.[408] The adaptive umbrella (ADUMB) sampling module[408] permits one to define umbrella sampling coordinates, and to carry out a series of biased simulations, in which the biases are adapted to obtain uniform sampling of the chosen coordinates. Ensemble averages are obtained as a weighted average of properties of the conformations from the biased simulations. The adaptive umbrella sampling module implements the Weighted Histogram Analysis Method[480,482,483,485,489] (see Section VI.A.) to determine weighting factors required to calculate the estimates for the unbiased system. The ADUMB module of CHARMM supports multidimensional adaptive umbrella sampling,[408] and multicanonical simulations.[405,601] The former is used to obtain uniform sampling of the space spanned by the chosen coordinates if several coordinates are of interest. The latter uses the potential energy of the system as one of the umbrella sampling coordinates, with the result that high and low energy conformers are sampled with comparable probability. These biasing methods have been shown to be efficient.[488] Since the effect of biases on the convergence of free energy values depends on the system and the property of interest, selection of the best biases to speed convergence has to be done on a case-by-case basis. Several biasing potentials have been combined with umbrella sampling to determine the free energy surfaces associated with conformational changes in biomolecules. For example, biasing potentials applied to proteins and peptides have been based on the radius of gyration,[298] native contact fraction (the fraction of contacts relative to the native protein structure),[299,602] RMS deviation relative to reference conformations,[603,604] the center-of-charge along a proton wire,[560] the position of ions along the axis of membrane channels,[33,91] and the pseudo-dihedral angles controlling DNA base-flipping.[81] An adaptive umbrella sampling approach has also been implemented for studying multidimensional reaction surfaces with combined QM/MM potentials.[605,606] In addition, a cubic spline interpolation procedure has been implemented for calculating an analytical bias potential, given the discrete PMF values at a series of points along a given reaction coordinate.[607] This procedure is particularly useful for studying chemical reactions where the approximate barrier height and shape of the PMF are known. It has been applied to a number of enzymatic reactions with the RXNCOR module.[258,259] These restraint functions are implemented in CHARMM and have been integrated with many of the tools for the analysis of conformational energetics and populations. Their application to protein and peptide folding[300,608] and to enzyme catalysis[258,259] has been reviewed.

Conformational Free-Energy Thermodynamic Integration (CFTI)

The CFTI approach is an extension of the well-known TI method developed for free energy simulations.[609] It is aimed at exploring multidimensional free energy surfaces.[610] The free energy gradient with respect to a selected set of conformational coordinates is calculated from a single simulation in which the coordinates are subjected to holonomic constraints.[610–612] This method is closely related to the "Blue Moon" calculation of the free energy along a reaction coordinate,[613] and has recently been analyzed and generalized to unconstrained simulations.[614]

The free energy derivatives are determined by averaging the forces acting on the constrained coordinates over an MD simulation. The generation of MD trajectories with fixed values of selected coordinates is performed using the holonomic constraint approach, which is part of the TSM method of Tobias and Brooks.[357,615] The basic TI formula for the derivative of the free energy G with respect to a conformational coordinate ξ is[616]

$$\frac{\partial G}{\partial \xi} = \left\langle \frac{\partial U}{\partial \xi} \right\rangle_{\xi} + k_{\mathrm{B}} T \left\langle \frac{\partial \ln J}{\partial \xi} \right\rangle_{\xi} \tag{23}$$

where U is the system potential energy, the angled brackets denote an average over a set of structures with ξ fixed, and J is the Jacobian of the transformation from Cartesian coordinates to a complete set of generalized coordinates, ξ (i.e., such that all conformations of the system may be represented by ξ). A generalization of the TI formula to several dimensions has also been developed.[610]

Multidimensional free energy gradients are calculated from the forces acting on chosen atoms and are evaluated at essentially no extra cost compared to a standard MD simulation. The method uses only local information about the free energy surface, which may be sampled more densely in regions of interest and less densely elsewhere. All the "soft" degrees of freedom in the system, e.g., all flexible dihedrals in a peptide, can be constrained to obtain both a complete free energy gradient surface and fast convergence of thermodynamic averages.[612,617]

The free energy gradient makes possible different approaches to exploring the molecular free energy surface. A series of calculations for a range of coordinate values allows for the calculation of free energy gradient maps, which can be integrated to yield free energy surfaces or free energy profiles linking conformations of interest.[612,617] The free energy gradient can also be used to perform an optimization of the free energy surface to locate free energy minima corresponding to stable structures.[611] Free energy profiles connecting the stable states may then be generated, and the free energy gradient integrated along them to yield conformational free energies and transition state barriers

on the molecular free energy surface. Numerical second derivatives of the free energy with respect to the coordinates of interest can be calculated, providing a measure of stiffness or stability.[611] The CFTI method has been applied to the exploration of free energy surfaces of several peptide and peptidomimetic systems: various helix types,[612] β-sheets and collagen triple-helices,[612] model β-peptides,[617] and the opioid peptide DPDPE in solution.[618]

VII.D. Transition Path Sampling

The TPS algorithm of Chandler and coworkers[561,562] uses Monte Carlo methods to sample the space of whole dynamic trajectories. Such simulations not only permit determination of the mechanisms of rare events but also the calculation of their rates. In other words, time-dependent phenomena can be investigated using importance sampling tools whose use has been traditionally limited to equilibrium properties.

The implementation of TPS in CHARMM[563] can be activated through options for the reaction coordinate definition (*RXNCor*) and MD (*DYNAmics*) commands. Two types of Monte Carlo moves are provided. In "shooting" moves,[561,562,619,620] a phase space point from an existing trajectory is selected, a perturbation is made (typically to the velocities in a deterministic system and to the random force in a stochastic one), and part or all of the trajectory is regenerated by integrating from the perturbed point to one or both endpoints. "Shifting" moves correspond to reptation in path space and involve extending the trajectory at one end by integration and shortening the trajectory at the other end. In both cases, new trajectories are accepted if and only if they satisfy the constraints that define the path ensemble of interest. Most often, these constraints are such that the endpoints of trajectories must have order parameter values corresponding to the reactant and product basins of an activated process, in which case the computational advantage over straightforward MD derives from the fact that TPS eliminates the waiting time for spontaneous fluctuations to the transition state region. Because trial paths are generated from existing ones, the method can be difficult to initiate in complex systems. To address this issue, a method for annealing biased paths to unbiased ones was developed recently and implemented in CHARMM.[401]

The interpretation of TPS (and more generally, MD) simulations to delineate a mechanism requires identifying molecular features specific to the transition state ensemble (defined here to be configurations with equal likelihoods of committing to reactant and product basins in additional simulations initiated with randomized momenta).[409,621] Because trial-and-error approaches to this task can require prohibitively large investments of human and computer time, Ma and Dinner[621] adapted automatic means for obtaining quantitative structure-activity relationships (QSARs) to commitment probability (p_B) prediction. The genetic neural network (GNN) QSAR method of So and Karplus[622,623] was used to determine the functional dependence of p_B on sets of up to four coordinates from a database of candidates, and to select the combination that gave the best fit. Application of this method enabled the identification of a collective solvent coordinate for the $C_{7eq} \rightarrow \alpha_R$ isomerization in the alanine dipeptide.[621] The TPS,[562] bias annealing,[401] and GNN[621] methods were recently combined to elucidate a mechanism for DNA damage recognition by the DNA repair protein O^6-alkylguanine DNA-alkyltransferase (AGT).[624]

VII.E. Coarse-Grained Elastic Models

Coarse-grained modeling approaches, which are based on reduced descriptions of molecules, are being increasingly utilized in studies of large systems, such as macromolecules and complexes. They can provide useful information at a fraction of the cost of the corresponding atomistic calculations (see also Section IX.D.). One type of coarse-grained model, the simplified elastic model, represents the protein by its Cα atoms and the potential energy by harmonic energy terms corresponding to springs between these atoms. Both "single-basin" and "multibasin" models have been developed. In the single-basin models, fluctuations of the system in the neighborhood of a single stable state, usually an unperturbed crystal structure, are of interest. The first such model to be introduced is the so-called Elastic Network Model (ENM).[380] More elaborate treatments are the Gaussian Network Model (GNM),[625] the Anisotropic Network Model (ANM),[381] and the recently introduced Generalized ANM (GANM),[626] which combines elements of the other models. Since the potential is harmonic, a normal mode analysis yields exact equilibrium properties, and the models have been used, for example, to give estimates for relative B factors that appear to be in reasonable agreement with experiment.[627] As a component of the vibrational analysis module VIBRAN in CHARMM, both the GNM and ANM calculations can be invoked with the GANM option, for which a selection is available to specify the atoms that are included in the coarse-grained network. An external file unit is provided for reading in other network parameters. On the basis of an ENM potential in the presence of external force perturbations, a linear response-type approach involving nonequilibrium simulations has been used to predict large conformational displacements in proteins.[628] Another single-basin coarse-grained method available in CHARMM is based on a Gō-like model.[512] An extension of coarse-grained models replaces an atomic description by force centers distributed in a uniform way inside an electron density envelope for the system obtained from cryo-EM.[629–631] An α-carbon-based model has also been used to study the coupling between allosteric transitions of the *E. Coli* chaperonin GroEL and the folding of a model substrate protein.[632] The results support those obtained with the TMD method and an all-atom representation for GroEL and the protein substrate.[586]

For systems that undergo large conformational changes, an approximate transition pathway or pathways between stable states can be determined through the use of a "multibasin" extension of the elastic network-type methods called the Plastic Network Model (PNM),[633] which incorporates ideas from valence bond theory.[634,635] For a two-state system, the PNM method constructs a 2×2 phenomenological Hamiltonian, where the diagonal elements are the ENM energy of each conformer, and the off-diagonal elements are a pre-defined mixing constant (or coupling parameter). The ground state energy of the system is the lowest eigenenergy of the diagonalized PNM Hamiltonian. The PNM module in CHARMM provides a simple

yet smooth and continuous coarse-grained potential, which can be used with the reaction path methods and nonequilibrium dynamics methods described in the previous parts of Section VII for the study of transition pathways between multiple protein conformations. The PNM method has been used with the TREK module in CHARMM to obtain free energy pathways for the open-to-closed conformational transition in adenylate kinase (ADK).[633] Recently, coarse-grained simulations combining PNM and TMD (Section VII.B.) have been performed to elucidate the torque generating mechanism of F1-ATPase during its hydrolysis cycle.[636] The PNM method can also be used as a conformationally adaptive rigidification potential with an all-atom force field in nonequilibrium all-atom simulations to prevent artifactual structural deformations induced by the use of simulation times that are much shorter than the actual transition times.

VII.F. Chemical Reactions and the Treatment of Nuclear Quantum Effects

The computational techniques described earlier, including reaction path optimizations, umbrella sampling and free energy simulations as well as combined QM/MM potential functions, provide the tools for modeling chemical reactions in condensed phases and in enzymes. The study of reactions was set forth as an important goal in the original CHARMM paper in 1983,[22] and was realized a few years later in the study of an S_N2 reaction in aqueous solution as the first application of a QM/MM potential in an MD free energy simulation.[248] Subsequent QM/MM studies, including detailed analyses of the energetic contributions of specific residues, have provided further insights into the roles of enzymes in lowering activation barriers.[251,258,637,638]

Transition state theory (TST) provides a fundamental approach for describing the rates of reactions in the gas phase, in solution, and in enzymes.[259] The central quantity is the free energy (PMF) along the reaction coordinate. The latter is expressed in terms of geometrical parameters, such as a dihedral angle in peptide bond isomerization or the difference between the bond distances for bonds being broken and formed in a proton transfer process[639] (see Figure 7). The free energy can also be determined as a function of a collective solvent reaction coordinate defined by the energy gap between the effective diabatic potentials of the reactant and product states.[640,641] The associated transmission coefficient, which determines the fraction of the trajectories that, having reached the transition state, go on to the product, can be calculated from multiple trajectories, starting from the transition state ensemble generated during the PMF simulations.[555,557] This approach was first applied to the enzyme triose phosphate isomerase,[642] for which the calculated transmission coefficient was found to be 0.4, indicating that the asymmetric stretch coordinate of the transferring proton is a good choice. In a later study of the enzymatic reaction catalyzed by haloalkane dehalogenase, in which the computed free energy barrier was 11 kcal/mol lower in the enzyme than in the corresponding reaction in aqueous solution, the transmission coefficient was found to be 0.53 in the enzyme, *versus* 0.26 in solution.[643] Applications to chemical reactions in solution and in enzymes have been reviewed.[258,259,639,644] TPS (Section VII.D.) provides a method that can be used to study the reactions for

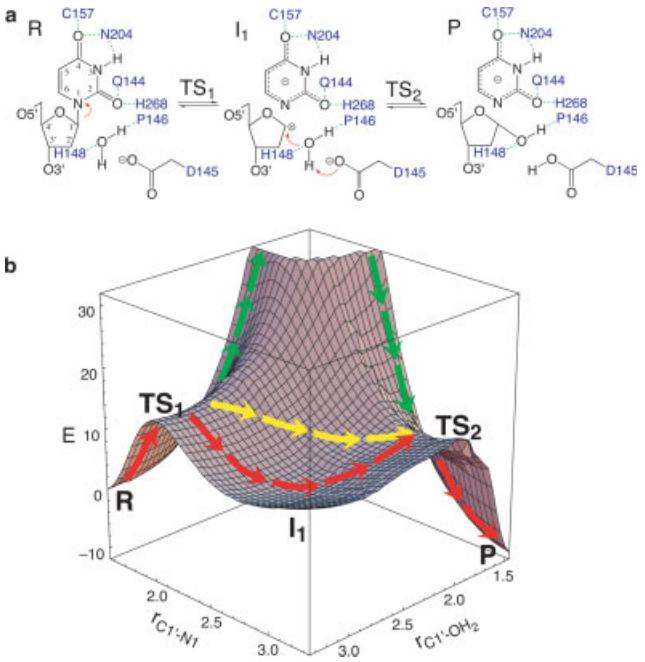

Figure 7. Reaction mechanism of the excision of misincorporated deoxyuridine from DNA by the uracil-DNA glycosylase UDG. (a) Schematic diagram. Electron transfers are indicated in red, hydrogen bonds in green and enzyme residues in blue. The dashed line to C157 indicates a Cα—Hα···O4 hydrogen bond. (b) Adiabatic potential energy surface as a function of $r_{C1'-N1}$ and $r_{C1'-OH2}$. In the region $r_{C1'-N1} \leq 2.20$ Å and $r_{C1'-OH2} \leq 2.00$ Å, the points above 32 kcal/mol are not shown for clarity. Red arrows follow the lowest energy pathway (stepwise dissociative); green arrows follow a perfect associative pathway; and yellow arrows follow a concerted pathway starting from the reactant structure. The states indicated are reactant (R), product (P), transition states (TS_1 and TS_2), and the oxocorbenium cation/anion intermediate (I_1) (From Dinner et al.[637]).

cases where the transition state is not known. A recent study with CHARMM of the hydride transfer reaction catalyzed by lactate dehydrogenase found that residues aligned along the donor and acceptor atoms of the hydride transfer reaction but distant from the active site are involved in the reaction.[645] These residues participate in compression and relaxation motions that help to bring the donor and acceptor atoms together so as to increase the tunneling probability.[646]

In contrast to most processes commonly studied with classical MD simulations (see Section V.B.), reactions involving the motion of hydrogen atoms and more generally reactions at low temperature have non-negligible quantum dynamical effects and require the use of quantized vibrations and the inclusion of tunneling corrections. Quantum dynamics is essential for treating kinetic isotope effects (KIEs) of chemical reactions, which are of great interest because the ratio of the rates between light and heavy isotopic reactions provides the most direct experimental method for characterizing the transition state of a chemical reaction. The CHARMMRATE module, which implements ensemble-averaged variational transition state theory with multidimen-

sional tunneling (EA-VTST/MT), provides a procedure for introducing quantized nuclear motion, given the classical PMF obtained from MD simulations, into the calculation of the rate constants of enzymatic reactions. The EA-VTST/MT method combines the POLYRATE program, for computing rates of gas-phase reactions[647–649] with free energy simulation methods employing combined QM/MM potentials in CHARMM.[248,256] In the EA-VTST/MT method, the classical PMF is first converted into a quasiclassical result, which includes quantum effects for all bound vibrational coordinates (but not in the reaction coordinate at the transition state), by making use of instantaneous normal mode frequencies along the reaction coordinate. This is followed by incorporating the contributions from nuclear tunneling in the reaction coordinate at the transition state based on optimized tunneling paths averaged over the transition state ensemble. In this procedure, the quantized system evolves in a fixed protein and solvent field; this "frozen bath" approximation is sufficient in many cases. Corrections to the frozen bath approximation can be introduced in computing the tunneling transmission coefficient by allowing for relaxation of the protein environment.[644]

Nuclear quantum effects can also be incorporated into enzyme kinetics modeling through Feynman path integral simulations, employing both classical[533] and combined QM/MM potential functions.[650,651] For combined QM/MM potentials, a Fock matrix updating procedure has been implemented into the QUB (Quantum Update in Bisection sampling) module for centroid path integral simulations, such that only the matrix elements for atoms that are treated with the path integral approach need to be recomputed. A method has been developed that combines the path integral approach with free energy simulations and umbrella sampling (PI-FEP/UM). This method yields improved convergence in computed KIEs.[650] As in the EA-VTST/MT method, the classical PMF is first determined by umbrella sampling. Centroid path integral simulations are then performed to obtain nuclear quantum contributions. Finally, free energy perturbation simulations are carried out to change the atomic masses to heavy ones by using the bisection sampling scheme to obtain KIEs.[650] The PI-FEP/UM calculations include both quantized vibrational free energies and tunneling. The method has been applied to several chemical reactions in solution and in enzymes, and KIEs have been determined for hydrogen and heavier elements (carbon and nitrogen).[650,652]

VIII. Analysis Techniques

The large amounts of data generated by MD and Monte Carlo simulations would be of limited utility without analysis facilities for deriving pertinent information about the system from them. During a simulation, CHARMM can intermittently write to the output file the values of all energy terms, as specified by the user in the *DYNAmics* command, together with some basic statistics (short-term and long-term averages, fluctuations and drifts). In addition, CHARMM can write the energy values, binary coordinates, velocities, and forces at user-specified intervals to files in a compact text format. All other analysis of the simulation, with a few exceptions (e.g., free energy calculations with PERT), is done via post-processing of the coordinate and/or velocity trajectory files that are generated in the simulation. CHARMM has comprehensive and flexible analysis facilities, which allow the efficient extraction of information from individual structures or trajectories for the calculation of many system properties. In this section, a description of the tools available for the analysis of static structures is given first, followed by a description of tools for the extraction and analysis of averaged and time-dependent information from trajectories. The section ends with a discussion of modules for more specialized analyses. Together with the general atom selection mechanism, these modules allow a very wide range of analysis to be performed. Should the need to program some new analysis functionality arise, there is a set of predefined hooks into various parts of CHARMM that allow relatively straightforward modifications to be implemented without changes to other parts of the program (see Section IX.A.).

The generation of the binary trajectory file during an MD simulation with CHARMM is controlled by the *DYNAmics* command. The trajectory I/O commands (*TRAJectory READ*/*WRITE*/*INQUire*) allow individual snapshots to be extracted from a trajectory (*TRAJectory READ*), so that all CHARMM analyses and processing functions for individual structures, as well as external programs, can be applied to a trajectory by using the looping capability of the CHARMM scripting language. This mode of analysis is thus very general, and allows operations to be performed on subsets of atoms that may change between snapshots on the basis, for example, of geometric criteria. New trajectories, with a subset of atoms or with coordinates recentered around a solute or superposed onto a reference structure, can also be constructed from one or several existing trajectory files.

VIII.A. Individual Structures

Structure

A large number of geometric characteristics of a structure can be determined using the coordinate manipulation (CORMAN) and internal coordinate (IC) modules (see Sections IX.B. and C.). Some examples are individual atom positions, distances between atoms, bond angles or torsion angles, and properties involving a larger number of atoms, such as the radius of gyration, least squares plane, accessible surface area, occupied and empty volumes, ring puckering, or helix axis and dipole moment. There are commands to find all distances, or just the minimum or maximum distances, between two sets of atoms specified with the general selection facility. Lists of hydrogen bonds and pairwise contacts between selected sets of atoms, as well as histograms of atom densities (radially or along the coordinate axes) can be easily generated. Coordinate differences, or RMS-deviations with or without least-squares superposition, can be calculated between two different coordinate sets (i.e., the main and comparison sets). Protein secondary structure can be analyzed using the definition of α- and β-structures proposed by Kabsch and Sander.[653]

Energetics

The potential energy of the whole system, a subset of the system, or the interaction energy between two subsets (*INTEraction* command) can be computed. Following an energy evaluation, the forces acting on all atoms, a breakdown of the energy into contributions from each atom, and the pressure are available. The user has control over which energy terms to include in the analysis, and the values of the individual terms are accessible at the CHARMM script level as variables.

VIII.B. Trajectories

A CHARMM trajectory, which is stored in one or more files, can be analyzed directly by several CHARMM commands and/or modules (e.g., *COOR, IC*, *VIBRan*, *CORRel*, *NMR*, *NOE*, *RDFSol*, *MONItor*). Prior to analysis, CHARMM trajectories can be processed by the *MERGe* command, for example, to reduce the number of coordinate sets in the trajectory, to remove a set of atoms (this has to be accompanied by the creation of a matching PSF), to orient the system with respect to a reference structure, or to undo the effects of recentering of molecules due to the use of PBC in the simulation.

Average Properties

In the CORMAN module a number of average properties can be calculated, including the average structure and RMS fluctuations around the average; distance and contact matrices (*COOR DMAT*),[299] which can be projected onto a reference distance matrix for analysis of, e.g., native contacts; and the distance fluctuation matrix and positional covariance matrix (*COOR COVA*), which can be used to reveal regions that move together.[31,654–656] Other average quantities which can be calculated include hydrogen bond average numbers and average lifetimes, histograms of hydrogen bond lifetimes and lengths; density, charge or dipole histograms; and internal coordinate averages. The pairwise RMSD can be calculated between all frames in one or two trajectories (in the latter case, element a_{ij} is the RMSD between frames i and j in trajectories 1 and 2, respectively). The *MONItor* command collects statistics on transitions between different minima for specified dihedral angles.

Techniques of conformational clustering are important tools for analyzing the nature of the conformational space sampled during the course of a molecular simulation. Clustering methods based on K-means or hierarchical techniques[298] can provide estimates of the extent and nature of conformational basins sampled during the simulation. A K-means clustering algorithm is implemented in CHARMM.[657] This algorithm requires input of a time series for specific sets of conformational variables - for example, sets of flexible torsion angles for a molecular system throughout the course of an MD trajectory - and a maximum radius for the Euclidian root-mean-square variation within any cluster. The K-means clustering algorithm then uses a simple neural-network scheme to iterate to a self-consistent set of clusters in the space of the specified variables. The clustering methodology is integrated with the CORREL and *MANTime* correlation function and time series manipulation methodologies in CHARMM and thus permits the flexible construction and combination of various time series for cluster analysis.

Another clustering technique implemented in CHARMM involves the projection of pairwise RMSDs between selected atoms in N frames of a trajectory onto a 2D plane, such that the Cartesian distances between the representative 2D points gives an approximation (least squared fit) to the RMS deviations between the actual structures.[171,658] Other clustering methods can easily be introduced into CHARMM using the appropriate scripts. An example is given in Krivov and Karplus.[166]

Time-Dependent Properties

Time series of several predefined types of geometric and energetic variables can be extracted for user-selected sets of atoms in the correlation module (CORREL) in an efficient manner, since the trajectory is processed only once to extract all the data. These time series can then be further manipulated; for example a vector time series can be normalized or converted to spherical coordinates, an angle time series can be made continuous, or the angle formed by two vector time series can be computed at each time point. The time series can be read from or written to external files. Auto and cross correlation functions can be computed from the time series data, either directly or using a second order Legendre polynomial.

Examples of time-dependent properties that the CORREL module can extract from a trajectory for a selected set of atoms include fluctuations in vectors, components, and lengths defined by atom positions; energy and hydrogen bond properties; and the dipole moment for selected atoms or for a solvent shell of specified thickness. See Supporting Information for a more complete list.

NMR Analysis and NOE Distance Restraints

The NMR facility may be used to analyze a number of NMR-related properties from a trajectory. Among the possible properties are those related to dipole-dipole fluctuations that govern the relaxation rates in solution NMR, such as T1, T2, NOE, ROE, and the Lipari–Szabo generalized order parameter,[324] as well as nonisotropically averaged properties observed for oriented membranes and liquid crystals, such as chemical shift anisotropy (CSA) and deuterium quadrupolar splitting and dipolar coupling order parameters.[31,137,659,660] Entropies associated with the generalized order parameters are estimated using the simple diffusion-in-a-cone model.[661] A trajectory can be analyzed as a whole, or in a series of windows of user specified duration, with or without removal of overall translation/rotation individually for each window; in the multiwindow case, averages and standard deviations of the extracted properties are reported. For trajectories created with a polar hydrogen representation, the NMR facility can add missing hydrogens for use in calculations involving proton NMR measurements. The NOE module, which is primarily used to introduce distance restraints based on NOE data for structure refinement,[301] also allows the analysis of how well a structure fits the restraints (see Section III.F.).

Solvent Analysis

The aqueous environment of biological macromolecules plays an essential role in their function. One of the advantages of MD

simulations of systems with explicit solvent is the ability to obtain a description at the atomic level of the interactions of the solvent with the macromolecule. Accordingly, CHARMM contains a suite of utilities for the analysis of solvent properties. In addition to the general analysis modules (e.g., CORREL), there is a facility (*COOR ANALysis*) for direct analysis of solvent properties. This makes possible the calculation of solvent–solvent, solvent–solute or solute–solute pair correlation functions with an excluded volume correction; translational and rotational diffusion, in shells of user-specified thickness around a set of atoms; velocity autocorrelation functions; number, charge or dipole density in 3D around a set of atoms; hydration numbers; the distance dependent Kirkwood *g*-factor[54]; and the dipole moment of a shell of solvent molecules. The pair correlation functions, as well as the distance dependent Kirkwood *g*-factor, charge–dipole or dipole–dipole orientational correlations functions between a set of reference atoms and solvent molecules, can also be computed using the RDFSOL module,[662,663] which is more efficient for large systems due to the use of a spatial decomposition when computing interatomic distances. The RDFSOL module is tightly integrated with the CRYSTAL/IMAGE functionality in CHARMM, which is particularly useful for solvent–solvent analyses.

Another useful solvent analysis tool is the *COOR HBONd* command, which uses the lists of hydrogen bond acceptors and donors in the PSF; no explicit H-bonding terms are included in the energy functions, but the acceptor/donor information simplifies the analysis, which is purely geometric. With polar- or all-hydrogen representations, it is advantageous to define the hydrogen bond in terms of the hydrogen and acceptor atoms; the relevant hydrogen atoms in this case are designated as donors. The *COOR HBONd* command takes two user-defined atom selections, one for the hydrogen bond donors (hydrogens) and one for the hydrogen bond acceptors, and determines from them all hydrogen bonds meeting the specified distance and angular criteria and calculates related properties. The calculated properties include the average number of hydrogen bonds, their geometries and lifetimes, and their length and lifetime histograms. The *COOR CONTact* variant of the command performs a similar function, except that it disregards the hydrogen bond donor/acceptor status of the atoms to be analyzed; it is useful, for example, for hydrophobic contact analysis. For the case where a solvent molecules moves in and out of contact with a given set of solute atoms during the simulation, the "intermittent" residence time (i.e., the time during which solvent molecules are present continuously within a given distance of the solute atoms) can be obtained using *COOR ANALysis*, as the relaxation time of the auto-correlation of the function $b_k(t)$; $b_k(t) = 1$ if water molecule k is within the specified volume at time t, and 0 otherwise.[664] For solvent analysis on simulations with periodic boundary conditions, the commands described here take care of the periodicity for simple lattices (for *COOR ANALysis* orthorhombic lattices; for *COOR HBONd* orthorhombic, truncated octahedral, rhombic dodecahedral, and 2D or 3D rhomboidal lattices). For solute–solvent analysis it can be advantageous to preprocess the trajectory such that the solute is placed in the center of each frame (*MERGe RECEnter*). In this way, subsequent analyses of solvent properties in the vicinity of the solute can be performed without the need to account for the periodicity of the system, as would otherwise be necessary for cases in which part of the solute molecule is outside, or near the edge, of the primary box.

VIII.C. Running Statistics

The ESTATS facility calculates running averages and standard deviations (fluctuations) of the energies of the system and its components "on-the-fly" during an MD simulation or any other calculation that serially calls the main energy routines. It collects the data at a user-specified step length for a user-specified interval during the calculation. The averages and fluctuations can be written to standard output or external files; they can also be assigned to CHARMM script variables.

IX. Miscellaneous Tools and Applications

To use CHARMM functionality for production calculations such as MD simulations, free energy estimates, and reaction path sampling, the initial state of the system has to be set up properly. CHARMM has an extensive set of model-building facilities that includes a suite of tools for manipulating the Cartesian and internal coordinates of the system, and an automated procedure for constructing the topologies of large biopolymers (proteins, nucleic acids, and carbohydrates) from their constituent units. As part of its model-building capabilities, CHARMM also has a course-grained macromolecular docking facility called EMAP. For analyzing the results of calculations, the coordinate manipulation tools can be used in conjunction with the highly flexible scripting language (Section II.C.), the extensive set of analysis tools described in Section VIII, and novel analysis routines implemented directly in the CHARMM code by the user through designated "generic" subroutines. Although CHARMM data files can be used by external graphics programs for visualization of the initial system as well as structures resulting from production calculations, CHARMM has its own internal graphics facility, which has particular strengths. This section presents an overview of these CHARMM facilities, as well as some additional details related to CHARMM use.

IX.A. Some Details of CHARMM Use

Generation of the Molecular System

Simulations of biomolecules and their environment in CHARMM make use of a basic protocol that is required to establish the critical data files. The reader should refer to the methodology introduced in Section II.A. CHARMM calculations are all initiated by specifying (and reading in) the topology file and parameter file for the system of interest. As noted in Section III, CHARMM provides topology and parameter files for proteins, nucleic acids, lipids, carbohydrates, certain solvents and many other relevant small molecules for a number of force fields, including those currently under development. Once specified in this way, the system being simulated is defined in terms of a set of "segments" consisting of groups of atoms called "residues." Residues in CHARMM can represent a particular

amino acid or nucleotide, a solvent molecule, etc. A set of residues is grouped together and "generated" using the *GENErate* command into a particular CHARMM segment of an internal file structure called the PSF; many biological macromolecules (proteins, nucleic acids) are linear polymers, and the *GENErate* command uses rules, as specified in the topology file, for covalently linking adjacent residues into a linear chain. The designation PSF was originally used for proteins but now is a general term used for describing the atomic connectivity, atom types and atomic charges for all of the molecules studied in CHARMM. Several segments can be generated by repeated application of the *GENErate* command, and these segments can be modified using *PATChes* to provide disulfide bond connectivity, alternate protonation states, modified terminal groups etc. Generally, each individual protein (or nucleic acid) chain is denoted as a separate segment; together with solvent, ligand or counter ion "segments," the chain segments make up the PSF. Once the PSF is generated, the atomic coordinates may be read in or built using the internal coordinate (*IC*) commands or the HBUILD routine to place hydrogen atoms[665] and complete the structure. Examples of CHARMM input scripts can be found on the CHARMM website (www.charmm.org) and in the "test" directory of all CHARMM distribution packages.

Data Files

Most of the information needed to specify the molecular system (RTF, parameters, coordinates) in CHARMM is stored in simple text files. The only main data file used by CHARMM that is in a binary format is the trajectory file, and CHARMM has built-in commands (*DYNA FORMat/UNFOrmat*) to convert this to/from a text file for interchange between computer systems with different binary representations. External data (text) files, e.g. containing a list of dihedral angles to be used with the internal coordinate manipulation commands for model building, can be streamed directly into the CHARMM input file via the *STREam* command. The CHARMM user specifies all file locations, file names and file formats to be used—the program makes no hidden assumptions about file locations or file-name extensions.

Atom Selections

The need to specify a subset of atoms, common to many operations in CHARMM, is met by a general recursive atom selection facility. Atom sets can be selected based on a number of properties including: atom number, IUPAC name or chemical type; segment identifier; residue identifier, name or number; distance from a point or other atom(s); connectivity (bonded to a selected atom, all atoms belonging to the same residue or group); the Cartesian coordinates; or any of several other properties contained in internal CHARMM arrays (e.g., charge, mass, force). Ranges and wildcards are allowed where appropriate, so that a single specification can encompass multiple atoms. Selections can be combined using Boolean operators (*.NOT., .AND., .OR.*), and they may also be given a name for later reference with the *DEFIne* command. For example, the command *DEFIne INTERESTING SELEct TYPE C*.AND. IRES* 40:50 *END* specifies the selection of all carbon atoms in residues 40 through 50, inclusive, and assigns this subset of atoms to the name "INTERESTING."

Units

CHARMM uses a mixed set of units that are commonly used by chemists. The distinct system of units for most commands is the "AKMA" system, where distances are measured in **A**ngstroms, energies in **k**cal/**m**ol, masses in **A**tomic mass units and charge in units of electron charge. Using this system, 20 AKMA time units is roughly 0.978 picoseconds. For convenience, all input and output of the time is in picoseconds. Other common units are also included; for example, vibrational frequencies are provided in wavenumbers (cm^{-1}). The documentation should be consulted for details on units.

Adding Functionality

CHARMM has a mechanism for allowing users to implement their own special-purpose subroutines without altering other parts of the program. Six main "hooks" into CHARMM are provided as templates for such modifications. USERSB is an empty subroutine called by the USER command, intended as a general CHARMM subroutine template; USERE calculates an additional user-supplied energy term; USRSEL carries out a user-supplied atom selection; USERNM specifies a user-supplied vector for normal mode analysis; USRTIM specifies a user-supplied time series for use with the CORREL facility; USRACM is a user-supplied accumulation routine called at the end of each step of dynamics for direct statistical analysis, as an alternative to post-processing analysis. This interface mechanism is designed for short, one-time efforts. If a user-supplied subroutine is of general use, the routine should be rewritten to conform to CHARMM coding standards and incorporated into the program as an additional feature (see Section XI.A.).

IX.B. Coordinate Manipulation and Analysis Tools

The coordinate manipulation (CORMAN) facility (*COORdinate* command) primarily handles the manipulation and analysis of structure and dynamics based on Cartesian coordinates. Seven functions of this facility were described in the first CHARMM article.[22] The facility now comprises a much more extensive set of command options. There are two primary sets of coordinates, the main set and the comparison set, and the various coordinate manipulation commands can be used with any subset of either set. The options also function with image atoms defined by periodicity or symmetry. In addition, a second comparison set can be used with the *SECOnd* option for all of the commands (COMP2 keyword); this is useful when there are two comparison structures, or when the main or first comparison coordinate set is being used for another function. The coordinate arrays can be assigned the system velocities (e.g., the comparison coordinates contain the velocities at the end of an MD simulation) or the system forces. A weighting array may be employed as a general utility (4th) array; mass weighting of the coordinate arrays (often used when they are assigned the system velocities or forces) is invoked with the *MASS* option. Examples of the

coordinate manipulation aspect of the *COOR* command are *COOR ORIEnt RMS*, which performs a best-fit of one structure with another (minimizes RMS difference) and *COOR AVERage*, which generates an interpolated structure. An example of the coordinate analysis aspect of the command is the *COOR COVAriance* option, which calculates a covariance matrix from the system's dynamic fluctuations. See Supporting Information for a more complete list. For more information and specific references for these command options, see the "corman.doc" section of the CHARMM documentation.

IX.C. Internal Coordinate Tools

The internal coordinate (INTCOR) facility (*IC* command) primarily deals with the interconversion between internal coordinates and Cartesian coordinates and the analysis of structure and dynamics based on internal coordinates. The original form of this facility has been previously described.[22] Together with the *COOR* command and options, the *IC* command options provide a complete nongraphical model-building facility. The facility now contains two independent internal coordinate table structures, the main and secondary IC tables. Each row of the tables has 10 components (four atom identifiers, two distance values, two angle values, one dihedral angle value, and a logical flag indicating whether the four atoms represent a linear or branched topology). Given the positions (Cartesian coordinates) of any three of the atoms in a row, the position of the fourth atom can be defined in relative terms with three values: a bond distance, a bond angle, and a dihedral angle specification. For a chain of connected atoms (such as a protein), the information in the internal coordinate tables allows the Cartesian coordinates of all the atoms of the chain to be calculated from any three adjacent atoms with known positions. The need for the calculation to be able to proceed in either direction along the chain (e.g., from the N-terminal end to the C-terminal end of a polypeptide chain, or *vice versa*) led to the symmetric structure of the rows in the IC table (bond length–bond angle–dihedral angle–bond angle–bond length). By necessity, the IC tables overspecify the structure. CHARMM employs an improper dihedral angle internal coordinate to specify the geometry at branch points, in which the central atom, from which the branching occurs, is the 3rd atom in the entry. The *IC* command options include *IC GENErate,* which generates an IC table for the selected atoms; *IC BUILd,* which transforms the internal coordinates to Cartesian coordinates; and *IC RANDom*, which randomizes selected torsion angles. See Supporting Information for a more complete list.

The internal coordinate tables are used by several other parts of CHARMM. The MCMA (Section V.D.) method uses them extensively for generating move sets.[143] The tables are also used for internal coordinate restraints, which may be used to restrain the system to particular internal coordinate values (*CONS IC* command). The vibrational analysis tools use the IC tables to present internal derivatives for normal modes of vibration. The IC tables are also used in adaptive umbrella sampling (Section VII.C.) and conformational searching with the Z Module (Section V.D.) or GALGOR facilities. The latter employs a genetic algorithm and is designed for docking small flexible ligands and rigid proteins.[666] For more information on any of these commands and features and for specific references, see "intcor.doc."

IX.D. EMAP: Molecular Modeling with Map Objects

High-resolution electron microscopy (EM) is rapidly emerging as a powerful method for obtaining low-resolution (10–30 Å) structures of macromolecular assemblies composed of hundreds of thousands or millions of atoms.[667] Docking of the individual macromolecular components, whose structures are available at high resolution, into the low-resolution EM maps of these assemblies can provide insights into the functional architecture of the macromolecular complexes; an example is given by the model for the actomyosin complex.[668,669] The EMAP facility in CHARMM is designed to carry out this kind of macromolecular fitting in an efficient way.

Conventional molecular modeling is performed at atomic resolution and relies on X-ray and NMR experiments to provide structural information, but the direct manipulation of very large biomolecular assemblies using atomic models is very computationally demanding. To mitigate this problem, methods for protein-protein docking, for example, often employ coarse-graining or other simplifying approximations.[670–672] The EMAP facility uses map objects, which are essentially rigid representations of macromolecules that lack a well-defined internal chemical structure, but are composed, instead, of spatial distributions of certain properties, such as electron density, charges, or van der Waals "core" (see below).[673] EMAP allows the user to fit map objects corresponding to individual structural components (e.g., individual protein molecules) to larger, multicomponent target map objects (e.g., single-particle EM maps of the complexes). The movement of the map objects is carried out through the use of data structures called rigid domains, which contain the position vector and orientation matrix associated with the map objects they represent. The fitting process for large macromolecules using these reduced represention is computationally more efficient than it would be using all-atom (conventional) models. Some macromolecular flexibility can be included by "blurring" the spatial distributions of molecular properties.

Several utilities are available to compare map objects and calculate interactions between them. Four types of cross correlation functions are implemented to examine the match between map objects: density correlation, Laplacian correlation, core-weighted density correlation, and core-weighted Laplacian correlation.[673] The "core" corresponds to the interior of the structure, specifically that part of the structure whose density distributions are unlikely to overlap with those of adjacent structures; the structure is mapped to a 3D grid and a "core index," which is a measure of the depth of burial, is calculated for each gridpoint in the structure with an iterative procedure that is based on the position of each gridpoint relative to the surface, its Laplacian-filtered density, and the core index of neighboring gridpoints. The core-weighted correlation function gives more accurate results than direct density correlations for locating correct matches. A grid-threading Monte Carlo (GTMC) algorithm has been implemented to search for the best fit of map objects.[673] The GTMC method combined with the core-weighted density

correlation function has been applied to study the molecular architecture and mechanism of an icosahedral pyruvate dehydrogenase complex.[674,675] Also, map–map interactions determined with the EMAP facility have been successfully applied in a protein–protein binding study.[676]

IX.E. CHARMM Graphics

Computer visualization has become an integral part of interpreting and understanding molecular data, and CHARMM provides several means of facilitating this process. One approach to molecular visualization in CHARMM utilizes an X11 window and a subcommand parser (GRAPHX). X11 is a widely supported graphics standard that is supplied on most Unix-based systems and is available as added software for other machines. The X11 display is "passive," i.e., the graphics window changes in response to typed commands (and not the mouse). This affords flexibility through the use of a scripting language, so that, for example, repeated complex tasks can be invoked via a single command (*STREam*). Commands are available to change atom size and color, change bond thickness, add atom-based labels, control which parts of the PSF are drawn, scale the image size, switch in and out of side-by-side stereo mode, define clipping planes, enable depth cueing, and perform other standard graphics operations. The immediate graphical feedback can also serve as a learning aid for new users of the CHARMM program. Examples of figures generated with the use of the CHARMM graphics facility appear in Woodcock et al.[571] The GRAPHX rendering model has been kept simple, so that even a large molecular system can be rendered quickly; stored trajectories for the system can be rendered directly to the screen to produce "on-the-fly" animations of an MD simulation. Details are given in the CHARMM documentation.

The graphics facility has aspects that make it well suited for use with other parts of CHARMM. The first is its direct use of the internal data structures of CHARMM, including the PSF, without an I/O step. This can facilitate the design of CHARMM input scripts (by allowing immediate visualization of coordinate manipulations, for example), especially when image atom transformations are involved. The fact that bonds are drawn as they are defined by the PSF, and not by interatomic distance searches, is also useful for the diagnosis of model-building problems or in multiscale modeling applications. A second feature of the facility is that, through the use of the general atom selection feature in CHARMM, the coloring of atoms can be based on many of the atom-related properties that are either stored or can be computed during a CHARMM run. For example, atoms can be colored according to their interaction energy or the forces from the last energy evaluation.

In addition to the CHARMM graphics facility, molecular visualization based on CHARMM calculations can be performed with external graphics programs such as VMD[677] and Python/VPython,[678–680] in conjunction with appropriately formatted CHARMM output files. Standard file formats for CHARMM output files include (of generality) Brookhaven PDB format, CARD coordinate file format (with or without the PSF), or binary coordinate trajectory file format (with the PSF). In addition to these standard file formats, the CHARMM graphics facility

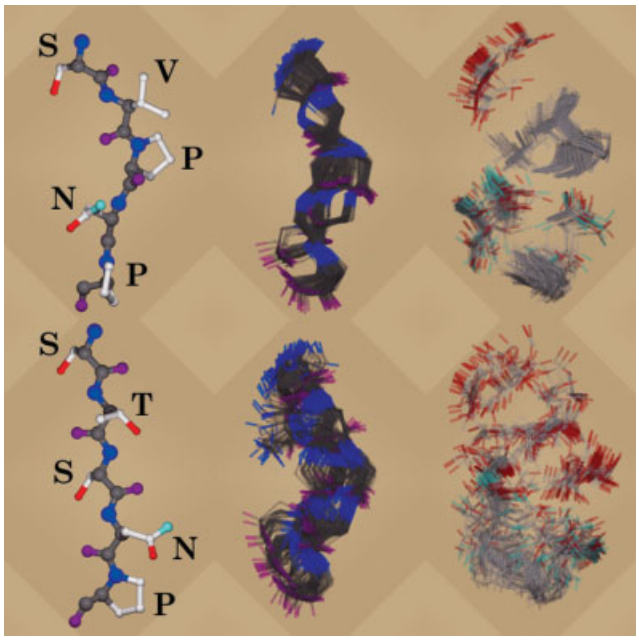

Figure 8. Six-panel figure depicting the results of a simulated annealing procedure for an antigenic peptide (top row) and an escape mutant (bottom row). The left hand column shows the peptide sequences in the reference orientation used to align the backbones for the middle and right hand columns. The middle column shows the aligned backbones and the right hand column shows only the side chains, in the same alignment, for the final coordinates from 100 simulated annealing runs. The small Val-Pro hydrophobic patch readily apparent in the top right panel is a likely antibody recognition site.[724] Each panel was produced from POV-Ray files exported via the CHARMM graphics facility; the files were edited to add the background and transparency features, and then processed into images via the POV-Ray program.

(which can be compiled without X11) provides for several others, notably a PostScript format (a close copy of the X11 screen drawing), and the output of molecular coordinates as a scene description for POV-Ray, a widely used and freely available ray-tracing program (www.povray.org). The primary use of the ray-tracing export facility in CHARMM is to produce high-quality figures for publications.[681–685] Examples of the output of this facility are shown in Figure 8. The image files produced can be combined to make animations in the MPEG video format. The use of the CHARMM graphics facility with these external graphics programs allows the generation of publication-quality graphics in a reproducible, script-based manner.

Accelrys has historically provided two graphics programs, Insight II and QUANTA, which can be used for graphical representation of CHARMM results. An automatic parameter estimation option for the CHARMm (commercial version) force field developed by F. A. Momany and R. Rone is available in QUANTA.[686] In recent years, progress has been made in providing a closely integrated CHARMm interface in a product called Discovery Studio (http://accelrys.com/products/discovery-studio/), which contains a library of preconfigured CHARMm

workflows created "behind the scenes" using the workflow management program Pipeline Pilot. An automated force field typing utility is available for use with all CHARMM/CHARMm force fields from the Discovery Studio interface.

X. Performance

Performance is one of the primary concerns in macromolecular simulations because longer simulation times (10–100 or more ns) are now often of interest for systems of increasing size. Many of the questions being addressed (e.g., free energy differences due to mutations) are more quantitative and require lengthy calculations to minimize the statistical error. To minimize the numerical error, double precision for floating point operations is used in much of CHARMM. The application of this standard, which is important for the reliability of the results, particularly in long simulations, carries with it a significant computational cost.

The performance of a program involves factors in at least three general categories: (1) the efficiency of the code running on a single processor, (2) the scalability of the code to many processors in parallel, and (3) the portability of the code to new computer hardware. This section describes the status of developments in the CHARMM program that concern these attributes and provides some relevant performance benchmarks.

X.A. Scalar Enhancements (FASTer options), Semiautomatic Code Expansion

A first step toward improving code performance involves single-processor enhancements. Recent developments include improvements in the optimized Ewald-direct calculation (real-space part of the Ewald sum) and the periodic boundary list routines. In addition, in the CHARMM program there are several ways for the user to carry out performance optimizations. They are controlled by the choice of the compiler preprocessor keywords and use the runtime *FASTer* and *LOOKup* commands. The optimal preprocessor keywords and *FASTer* command options to use in a given calculation depend not only on the problem (system size, type of calculation), but also on the computer environment, since processor architectures and compilers differ. Although there are general guidelines, it is generally up to the user to determine which compilation and runtime options result in the most efficient code in a given case.

EXPAND Preprocessor Keyword

A number of preprocessor keywords are concerned with obtaining the best performance for individual systems. This subsection describes the use of the EXPAND and associated keywords. Other performance-related preprocessor keywords are discussed later; for a more complete discussion of the preprocessor, see Section XI.A.

The "EXPAND" preprocessor keyword is designed specifically to enhance the performance of the CHARMM code through preprocessor-level optimizations that supplement the intrinsic optimization procedures of modern Fortran compilers. The "EXPAND" keyword instructs the compiler preprocessor to automatically expand the innermost loops in the selected routines. This is useful because there are many IF statements in the loops of the nonbonded interaction energy routines that are needed to support a variety of CHARMM methods; expansion moves these IF statements out of the loops. More recently this kind of expansion has been extended to whole subroutines. The procedure essentially introduces variables into the name of a subroutine that correspond to branches of its internal IF constructs, so that the subroutine is transformed into a "generic" parent subroutine. At compile time, the parent subroutine is automatically replaced by numerous daughter routines, each occurring within a larger IF block structure as specific instances of the variable parent subroutine, but with their internal IF statements removed. Hence, in this expansion procedure, a subroutine can be written and tested as a single routine with many internal constant IF tests, and then expanded into a large set of efficient routines that lack the IF tests. Expansion of subroutines with this technique can improve performance by 10–30%, depending on the code and the compiler.

FASTer Command

The *FASTer* command controls the use of the fast energy routines in CHARMM, which are essentially streamlined, optimized versions of the slower, full-feature routines. Many internal IF statements, as well as analysis and print options, second derivatives, and support for several nonbonded energy options are absent from the "fastest" versions of the fast routines. This significantly speeds up their execution times, but places some restrictions on their use. The options for the *FASTer* command are: *OFF, DEFAult, GENEric, ON,* and *EXPAnd*. The *OFF* option disables the faster routines entirely and invokes the slow, full-feature energy routines. The *DEFAult* option causes the use of the fast routines when possible. The *GENEric* option invokes the "generic" versions of the fast subroutines, which support most CHARMM methods and options, including second derivatives. The *ON* option invokes the faster but more limited fast routines, and it is the default in CHARMM. The *EXPAnd* option also invokes the faster routines, but with expansion as described earlier, and it must be used in connection with the use of the EXPAND preprocessor keyword during compilation. The *EXPAnd* option generally gives the best performance, but as mentioned, some methods and nonbonded energy options are not supported in connection with it. (See the CHARMM documentation, under "energy.doc," for further details.) Using *FASTer ON* (without code expansion or lookup tables) the single-processor performance on a standard 23,000-atom joint AMBER-CHARMM (JAC) benchmark (DHFR with explicit solvent, periodic boundary conditions and PME on an IBM p-Series, Power4+ CPUs) for CHARMM (161 ps/day of MD simulation time) is similar to that of Amber 8 (PMEMD, 128 ps/day), NAMD (Version 2.5, 135 ps/day), and Amber 9 (PMEMD, 197 ps/day); see also later.

Lookup Tables

In simulations of large systems with an explicit representation of solvent (usually water), the calculation of the solvent–solvent nonbonded interactions consumes a significant fraction (often on

the order of 90%) of the total CPU time. The evaluation of each interatomic interaction requires several floating-point operations, including division, and square root operations that are quite expensive. One approach to increasing speed is to code the routines that handle these types of limited but time-consuming operations in assembly language; however, assembly language is difficult to modify and to port to different computer architectures. Although it is used in GROMACS,[687] it does not appear to significantly increase the speed of the code over what can be achieved with lookup tables. Lookup tables circumvent the need for many of the floating-point calculations and hence achieve an important single-processor speedup. Tables are easy to set up for any functional form using the same high-level programming language that is used for the rest of the code (i.e., Fortran 95). However, if there are many kinds of interactions, the tables can require so much memory that the speed advantages of this approach are diminished because of inefficient cache-memory use. In CHARMM, a table lookup routine has been implemented with separate tables for solvent-solvent, solute–solvent and solute–solute interactions (LOOKUP precompiler keyword; *LOOKup* command). These lookup tables (one set containing the forces and, optionally, one set containing the energies for each combination of atom types) are indexed using the square of the interatomic distance, thus avoiding the square root. The lookup routine can perform linear interpolation between table entries for increased accuracy. This approach is memory-efficient for solvent–solvent interactions due to the small number of atom types involved (typically two for the common three-site water models), since only three force tables (O—O, H—H, O—H) and possibly three energy tables are required. The magnitude of the speedup due to the use of the lookup table depends both on the size and composition of the molecular system as well as on the computer system. The operation count in the inner loop is reduced by ~50%, which is reflected in typical speedups of 1.5–2 when compared with the standard fast energy routines in CHARMM, with the higher number obtained for systems whose interactions are dominated by solvent[688]; for a system consisting of 46000 TIP3P water molecules, without PBC, list update, or PME, 100 MD steps take 90 s with the lookup tables, compared to 190 s with standard CHARMM or 129 s with GROMACS. In four spherical cutoff benchmarks[688] (systems ranging from 14,000 to 140,000 atoms), the double precision lookup code is faster than the assembler code in GROMACS, also in double precision. The table lookup method has been implemented in CHARMM for use with atom-based spherical cutoffs or the real space part of PME, with or without PBC, and it runs in parallel. In NVE simulations using the lookup tables with linear interpolation, energy has been shown to be well conserved.[688]

X.B. Parallel Computation

As many systems of biological interest, such as solvated protein complexes and membranes, are large, and since long simulations of such systems are often required, the performance of massively parallel MD calculations on supercomputers or clusters of hundreds or more PCs has become an integral part of the field of computational biophysics. There are many facets to parallel MD methods, and the reader is referred to any of several articles on the subject for a more thorough treatment[689–693] The most important element in the different methods is the choice of parallelization model, which determines the manner in which the "work" of a calculation is distributed among the CPUs. For molecular mechanics/dynamics calculations, there are at least three general classes of models: (1) atom decomposition (replicated data), (2) force decomposition, and (3) spatial or domain decomposition.

In atom decomposition, for a computer system with p CPUs, each CPU is essentially assigned every pth pass through a loop. For the bond energies, for example, a given CPU handles every pth bond. For the nonbonded (van der Waals and electrostatic) energies, which for large systems require the most computer time, each CPU handles the interactions for every pth atom. One of the advantages of this scheme is the load balance is very good—i.e., the distribution of tasks among the CPUs is uniform. In CHARMM, the loss of performance due to load balance in the atom decomposition model is typically less than 5%, and the model performs well for up to 32–64 CPUs, particularly on shared-memory machines such as the IBM SP2, the SGI Altix series, and the CRAY XT4. After recent enhancements, such as the implementation of a column-FFT (COLFFT keyword) for PME calculations, which reduces communication costs by partitioning the system into 1-D "columns" and reorganizing the FFT calculation, the atom decomposition model scales with a parallel efficiency of ~0.6 using 32 CPUs and ~0.3 to 0.4 using 64 CPUs on a Cray XT4 (dual-core AMD Opteron processors) for MD simulations of systems of 50,000–400,000 atoms with PBC and PME (see Table 2a). On this machine, the scaling is similar for the largest and smallest systems. On a distributed memory cluster (8 Gb/s infiniband interconnects; see Table 2b) the scaling is approximately the same or better at 32 CPUs, but has a somewhat wider range (~0.2 to 0.5) for 64 CPUs, with scaling for the larger systems that is poorer than on the shared memory machine. This level of scaling is often considered adequate for applications on many computer systems, and, for certain applications, even on machines having a very large number of processors—e.g., for the generation of many independent MD trajectories, (each of which is propagated on a fraction of the CPUs). The disadvantage of the atom decompositon model is that the communication costs are high for large numbers of CPUs, because all of the data in the system must be updated on each CPU. This cost is significantly reduced by the use of "recursive doubling" or "hypercube" algorithms,[694] which change the number of necessary communication calls from P to $\log_2 P$. Still, for large systems and large numbers of CPUs, the time spent on communication dominates the total run times (wall-clock times), especially on distributed-memory clusters of CPUs (as illustrated earlier), and the scheme becomes inefficient. The atom decomposition model, which was the first one to be implemented in CHARMM, is the most thoroughly integrated with the various CHARMM functionalities. It is the default, and is still widely used, particularly on many "local" clusters, which have up to 100 or 200 CPUs that are shared among multiple users. While most modern-day efforts to parallelize biomolecular simulation programs focus on standard MD with either spherical cutoffs or PME for long-range electrostatic interactions, in

Table 2. Approximate Scaling Behavior of the CHARMM Atom Decomposition (AD) Model.

	COLFFT	DEFAULT
(a) Shared-memory supercomputer		
1	100	100
2	91–95	90–95
4	87–91	78–90
8	82–95	78–83
16	71–79	66–74
32	56–63	50–60
64	39–45	28–38
128	20–28	12–21
(b) A distributed memory cluster		
1	100	100
2	94–99	93–97
4	91–96	88–94
8	86–89	82–86
16	73–80	69–75
32	61–68	56–65
64	17–53	24–47
128	27–40	22–25

The table lists the percent parallel efficiency ranges of the AD model for various numbers of processors carrying out MD simulations of proteins in an explicit water environment (50,000–400,000 atoms total) on a) a shared-memory supercomputer (Cray XT4, 2.6 GHz dual-core AMD Opteron nodes) and b) a distributed memory cluster (dual-core 2.8 GHz AMD Opteron nodes, w/8 Gb/s Infiniband interconnects). The simulations were carried out with periodic boundary conditions, PME for long-range electrostatics, an update frequency of 25 steps, an image update frequency of 50 steps, and the BYCB listbuilder. The "COLFFT" columns gives the results with the recently introduced COLFFT code for faster PME calculations on large numbers of CPUs. On the larger systems and for smaller numbers of CPUs (1–4), the default code has faster (2–10%) absolute times (not shown).

CHARMM, many of the other modules/methods that are available also run well in parallel under the atom decomposition model. The ones that are most commonly used are: QM/MM methods, the EEF1 solvation model, the replica (molecular replication) methods, the TREK reaction-path facility, the PERT free energy methods, TMD, the HQBM external perturbation facility, adaptive umbrella sampling, soft core potentials, the Drude oscillator polarizable model, and the VV2 operator-splitting velocity Verlet integrator. For the communication scheme, CHARMM uses a customized version of MPI, called CMPI,[695] which includes specialized operations optimized for hypercube communication topologies and which can be useful more generally for synchronous communication schemes in networks with higher latency.

In the force decomposition model,[689,690] the $N \times N$ matrix of nonbonded interparticle interactions is partitioned into p pieces and the set of N atoms is partitioned into b blocks, where $p = b(b + 1)/2$. Each of the p pieces is assigned to a different CPU. The communication cost is reduced relative to that of the atom decomposition model, because each CPU must only obtain the data of the CPUs assigned to the same columns or rows of the interaction matrix, rather than all other CPUs. In principle, the amount of data per CPU per communication call (the width of the blocks in the interaction matrix) drops with increasing numbers of CPUs until the limit of $b = N$ is reached (one atom per CPU per call). The disadvantage of the scheme is mainly that the number of necessary communication calls still rises with the square root of the number of CPUs, since the numbers of CPUs in each row and column increase in this way. A force decomposition scheme has been partially implemented in CHARMM[696] and further developments (particularly improvements in load-balancing) are in progress.

Spatial (domain) decomposition schemes are essential for the effective use of large shared-memory supercomputers and commodity clusters of thousands of processors. The central idea in this approach is to partition the molecular system into spatial regions and then to map or assign the CPUs to nonoverlapping subsets of these regions. The partitioning of space, the assignment of CPUs, and the partitioning of the calculation, can be done in a number of ways,[692,693,697–700] but the spatial decomposition methods all have in common the important attribute that the data in each region is communicated only to nearby regions. This property reduces the communication costs of spatial decomposition schemes relative to those of the other methods for large numbers of CPUs. If the system is partitioned into cubical regions whose side length exceeds the nonbonded cutoff distance, the CPU assigned to a given cube must at most obtain data from the 26 surrounding cubes.[698] In the direct implementation of this method, each CPU is responsible for the calculation of (about half of) the interactions involving the atoms in its assigned regions. The disadvantages of the method include the fact that load balancing is not straightforward, especially in irregularly shaped systems or ones with inhomogeneous densities. Also, unless more sophisticated modifications are implemented, the maximal number of regions to which CPUs can be assigned is the total number of cubes in the system, or roughly V/r^3, where V is the volume circumscribing the system and r is the cube side length (e.g., nonbonded cutoff distance). To overcome the latter limitation, some programs, such as NAMD[693] use what is essentially a combination of force and spatial decomposition methods. A more recent development in spatial decomposition models is the introduction of so-called *neutral territory methods*,[691,692] in which the spatial assignments of the CPUs are done in a manner similar to that described earlier, but in which each CPU is responsible for the interactions involving atoms that are often in regions outside its own. In the "midpoint" method, for example, a CPU is responsible for an interaction if the midpoint between the interacting atoms is within $r/2$ of its region.[692] Compared to conventional domain decomposition approaches, these methods reduce the "import volume" or amount of data each processor must communicate with its neighbors, and hence they can be more efficient for larger numbers (e.g., 1024) CPUs. Recently, a spatial decomposition model based on the BYCC list-builder[314] has been partially implemented in CHARMM. The scheme, which is under development, makes use of the fact that in the cubical partitioning approach described earlier, each CPU must obtain the data from only those CPUs assigned to regions within the "shell" of cubes surrounding its own region. It achieves good load-balancing by making adjustments to the spatial assignments of the CPUs

during execution. Refinements, including support of periodic boundary conditions and other facilities in CHARMM, are currently underway. More detailed information on the parallelization of CHARMM, including a list of modules that run in parallel, may be obtained from the "parallel.doc" section of the CHARMM documentation.

X.C. Portability

Because of the variety of available computer hardware and software platforms, and because of continual changes and improvements in them over time, it is important for a program to be portable. For example, in the past, supercomputers were based on vector processors, and it was possible to compile CHARMM executables that were optimized for several specific vector architectures[701] (using the CRAYVEC, PARVEC, and VECTOR preprocessor keywords); these features were removed (with CHARMM version 31) because the architectures were no longer of interest (although the features are available in older versions of the program, which are archived at Harvard). Modern-day, high-performance computer systems are based on multiprocessor architectures (of up to 100,000 processors or more). A number of different architectures exist, from so-called Beowulf clusters connected by widely available off-the-shelf network communication equipment, to massively parallel systems from major computer vendors (e.g., the CRAY TX4 or the IBM Blue Gene) with much faster and more specialized connections that improve interprocessor communication. CHARMM has been ported to nearly all these machines, in addition to Macs and PCs, and most other currently available machines, processors, operating systems, and compilers. It also runs on clusters of special-purpose "MDGRAPE" MD computers[702] and with certain accelerator hardware tools (e.g. "MD Server" at NEC). Efforts to port the CHARMM code to graphical CPUs (GPUs) are currently ongoing.

To make this portability possible, CHARMM development standards have limited dependencies on vendor-specific programming language extensions. In addition, CHARMM has a hierarchical set of communication routines that make it easily adaptable to different parallel libraries.[695] In most cases, no source code modifications are required to optimize CHARMM's parallelism for a new machine architecture, e.g., any of the variety of multicore processors and systems that have been introduced in recent years.[703] There are several levels of communication routines, the highest of which is called from the standard energy routines and is independent of the specific parallel architecture and machine type. The lowest level routines directly call "send" and "receive" primitives from the system libraries. The precompiler determines which routines are included in a CHARMM compilation (as specified in the "build/pref.dat" file). The use of the optimal routines for a given system and machine type significantly improves the performance of the code in some cases.

XI. Program Management

CHARMM has over 550,000 lines of source code, is under continual evolution, and has to serve a large user community. These conditions create a set of administrative challenges. The contributions of a large group of developers from different parts of the world (see also Section XIII), often to overlapping parts of the code, must be systematized, integrated, organized, documented, and tested in a manner that allows the program to continue to grow in an error-free manner while preserving its many preexisting functions. In addition, the composition and distribution of the various versions of the program must be managed. This section describes some of the administrative and testing procedures that have been put in place, as well as the program's documentation and official website (charmm.org). The program's general organization, extent of usage, language history, and preprocessor function are also reviewed.

Table 3. Additional CHARMM Developers.

Cristobal Alambra	Thomas A. Halgren	Tibor Rudas
Ioan Andricioaei	Sergio Hassan	Paul Sherwood
Jay L. Banks	Jie Hu	Tom Simonson
Robert Best	Toshiko Ichiye	Jeremy Smith
Arnaud Blondel	Mary E. Karpen	Lingchun Song
John Brady	Jana Khandogin	David J. States
Robert E. Bruccoleri	Jeyapandian Kottalam	Peter J. Steinbach
Axel Brunger	Ansuman Lahiri	Roland Stote
Jhih-Wei Chu	Michael S. Lee	John Straub
Michael Crowley	Paul Lyne	Sundaramoothi Swaminathan
Ryszard Czerminski	Ao Ma	Walter Thiel
Yuqing Deng	Dan T. Major	Douglas J. Tobias
Ron Elber	Paul Maragakis	Don G. Truhlar
Marcus Elstner	Francois Marchand	Arjan van der Vaart
Jeff Evanseck	Robert Nagle	Herman van Vlijmen
Scott Feller	Kwangho Nam	Joanna Wiorkiewicz
Martin J. Field	Tom Ngo	Masa Watanabe
Stephen H. Fleischman	Barry D. Olafson	Thomas B. Woolf
Mireia Garcia-Viloca	Riccardo Pellarin	Hyung-June Woo
Bruce Gelin	David Perahia	Wangshen Xie
Urs Haberthuer	B. Montgomery Pettitt	William S. Young
Michael F. Hagan	Walter E. Reiher III	

Past and present CHARMM developers (in addition to the authors of the article).

XI.A. Administration and Distribution of CHARMM

General Administration and Code Distribution

Through the collaborative efforts of many developers (see Table 3) and the CHARMM manager, the ongoing administration of the CHARMM program has evolved over more than 15 years into a stable procedure that makes possible the continued development of the program as a robust, versatile, and well-integrated molecular simulation package. There are two versions of the program: one that is available only to current CHARMM developers as a basis for code enhancements, and one that is released, also as source code, to a large and growing community of users. Two of the central functions of CHARMM administration are (1) deciding which new features are to be included in the release version of the program and (2) creating a new developmental version. Every 6 months, revised versions are distributed. New

features and enhancements are incorporated into the developmental revision and bugs are fixed in the release revision. At present, December 30 and June 30 are the deadlines for submission of developments for the February 15 and August 15 distributions, respectively. Submissions normally include either new source files or modified versions of preexisting source files, or both, as well as the required documentation, testcases, and release notes (see also "developer.doc"). After collection of all the submitted code, interdependent modifications are merged, conflicts are resolved, and the integration is finally confirmed by checking all test cases. The CVS (Concurrent Versions System) repository is then updated to include the new developmental and release versions; all versions since c24 are archived in this repository; versions 22 and 23, which predated the use of CVS, are archived separately.

The CHARMM program is distributed as source code to individual academic research groups (see http://www.charmm.org/info/license.shtml for current information on how to obtain a license). For-profit companies should contact Accelrys Inc. (www.accelrys.com).

Organization of the Code

CHARMM distribution packages include the program source, the documentation, and the support data. The content of the current version, c34b1, is listed in Table 4. The "ChangeLog" files contain release notes of versions 23 through 34 (see www.charmm.org Web site). The source code is located in the "source" directory. Each subdirectory of "source" contains the source files of a given module, with the notable exception of the "include" files, which are collected in the "source/fcm" directory. The preprocessor (prefx), which is required to install an executable, and a set of shell scripts that are useful for modifying the program code are found in the "tool" directory. The compilation of CHARMM requires the use of the Makefile corresponding to the given platform; this file is created in the "build" directory, where installation takes place, and where the subdirectory "UNX" contains Makefile templates for the machines supported by CHARMM. A C-shell script, "install.com," drives the installation procedure. The current version of the force field parameter files is located in the "toppar" directory. Previous versions of these files can be found in the "toppar_history" subdirectory. The "doc" directory comprises the full set of documentation files. The "support" directory contains miscellaneous files that are either required for certain CHARMM functions (e.g., specialized parameter files) or useful as adjuncts (e.g., helpful input scripts). The subdirectory "support/aspara" contains implicit solvation parameter files and "support/bpot" contains stochastic boundary potential files (see also http://mmtsb.org/webservices/sbmdpotential.html). The "support/form" subdirectory contains forms for reporting user problems, bugs and development projects, and "support/htmldoc" contains facilities for converting info document files into html files. A few examples of image transformation files are included in the "support/imtran" subdirectory. The "support/MMFF" subdirectory contains a number of parameter files required for use of the MMFF.

Table 4. CHARMM Version c34b1 Package Contents.

Directory	Subdirectory	Contents
build	UNX	Makefiles and installation scripts
ChangeLogs		Release notes
doc		Documentation
source	adumb	ADaptive UMBrella sampling simulation
	cadint	CADPAC interface
	cff	Consistent Force Field
	charmm	Parsing and initialization routines
	correl	Time series and correlation functions
	dimb	Diagonalization In a Mixed Basis method
	dynamc	Dynamics integrator subroutines
	emap	MAP Object Manipulation
	energy	Energy subroutines
	fcm	Include files
	flucq	QM/MM Fluctuating Charge Potential
	gamint	QM/MM method interface to GAMESS-US
	gener	PSF generation and manipulation
	graphics	Graphics subprograms
	gukint	QM/MM method interface to GAMESS-UK
	image	Periodic boundary methods
	io	File I/O subroutines
	machdep	Machine dependent codes
	manip	Various structure and energy manipulation methods
	mbond	Multi-body dynamics
	mc	Monte Carlo simulation
	minmiz	Minimization programs
	misc	Miscellaneous energy and structure programs
	mmff	Merck Molecular Force Field
	mndint	QM/MM method interface to MNDO97q
	moldyn	Multi-body MOLDYN codes
	molvib	Molecular vibrational analysis facility
	nbonds	Non-bonded energy routines
	pert	Free energy simulation
	pipf	Polarizable Intermolecular Potential Functions
	prate	POLYRATE interface
	quantum	QM/MM method interface to MOPAC
	rxncor	Reaction coordinate manipulation
	sccdftbint	QM/MM method interface to SCCDFTB
	shapes	Molecular shape descriptor method
	solvation	Reference Interaction Site Model
	squantm	QM/MM method interface to SQUANTM
	util	String and memory space management codes
	vibran	Vibrational analysis facility
support	aspara	Implicit solvation parameter files
	bpot	Stochastic boundary potential files
	form	Forms to report problems and fixes
	htmldoc	Info to html file conversion scripts
	imtran	image transformation files
	MMFF	Merck Molecular Force Field parameter files
	trek	TReK initial path examples
test	c20test	Version c20 testcase input files
	c22test	Version c22 testcase input files
	...	...
	c34test	Version c34 testcase input files
	data	Data files for testcases
tool		Installation scripts
toppar		Topology and force field parameter files

Language History

Because the development of the program that would eventually become CHARMM began in the mid-1970s (see Epilogue), before FORTRAN 77 was widely available, data structures and advanced flow control were incorporated into the program design. The early versions of CHARMM were written in FLECS, since it supported a variety of control statements such as block-*if*, *unless*, *when-else*, *conditional*, *select*, *repeat*, *while* and *until*. To generate the FORTRAN source, the FLECS source was processed by the FLECS compiler, *flexfort*. Data structures for the connectivity (PSF), residue topology (RTF), force field parameters (PARM), images, etc., were built in FORTRAN array common blocks. A HEAP and STACK structure were also implemented using very long 1D arrays in the common block to enable internal program memory management. HEAP can be expanded using the malloc function of the 'C' language. In 1993, the FLECS source was converted into standard FORTRAN 77, and the parts of the code that were not convertible were eliminated. Since version 24 (1994), all CHARMM source code has been FORTRAN/Fortran-based except for a few routines involving machine-specific operations, which are written in 'C'. As of July 2005, new developments are required to be written in Fortran 95 (and allowed extensions). The Fortran 77 portion of the code is currently being converted to Fortran 95.

The Preprocessor and Its Function

CHARMM is implemented as a single, large cohesive program that is developed for use on a variety of hardware platforms with numerous compile options. The customization of the executable from a single source is accomplished by the use of a CHARMM-specific preprocessor, PREFX, which reads source files as input and produces FORTRAN files for subsequent compilation. PREFX was developed within the CHARMM community in 1989 and provides the following capabilities:

- Allows selective compile of code based on passed or derived flags.
- Supports a size directive allowing executables to support larger (or smaller) system sizes.
- Handles the inclusion of FORTRAN include files in a general manner.
- Allows semi-automatic code expansion and subroutine expansion (see Section X.A.).
- Allows comments on source lines following a "!" (a nonstandard feature in F77).
- Handles the conversion to single precision.
- Checks noncomment lines for lengths exceeding 72 places (important for CHARMM versions preceding c35).
- Inserts keyword lists into selected FORTRAN arrays (or prints them on execution).
- Processes inline substitution of variable or subroutine names.

The determination of what modules/methods are included in a CHARMM executable depends upon the keyword list in the "build/*platform*/pref.dat" file. The keywords in this list correspond to various methods and capabilities of the CHARMM program (e.g., "GBMV" module), and the preprocessor uses them to select the parts of the code to be compiled. For convenience, the default pref.dat list is extensive, so that "out-of-the-box" compilations of CHARMM may result in executables containing features that are not necessary for the user's intended application, and this may in some cases reduce speed. The user may improve the performance of the executable by removing the preprocessor keywords corresponding to methods that are not needed, and then recompiling. Although the various methods in CHARMM are designed to be modular, there exist significant interdependencies, so that the user is advised to carry out these preprocessor keyword list modifications with care and to check the results for consistency in test calculations.

Version Chronology

A chronology of the developmental and release versions of CHARMM since the distribution of version 22 on January 1, 1991 is displayed in Table 5. CHARMM version 19 was finalized with the accompanying parameter set PARAM19 in 1989. Earlier versions were distributed at varying time intervals. When the FLECS to FORTRAN source code conversion was completed, the need for a version control system was recognized, and the CVS system was introduced into the management of CHARMM with version 24 in 1994. Since then, all files in the CHARMM program have been subject to CVS control. As of c24a1, CHARMM program distributions were divided into developmental and release versions. Developmental versions carry newly introduced features and enhancements that are in the testing phase, and release versions contain only stable and tested modules. The current convention for version numbering began with version 26. In "c*nn*(*a*/*b*)*m*," c is for CHARMM, *nn* is the version number, *a* (alpha) is for developmental, *b* (beta) is for release, and *m* is the revision number. For example, c32a1 is CHARMM 32 developmental revision 1 and c31b1 is CHARMM 31 release revision 1.

The last column of Table 5 lists new methods and features introduced into each developmental revision, most of which have been described in this review. Interfaces have been implemented for MOPAC QM/MM, GAMESS-US, GAMESS-UK, Q-Chem, CADPAC, POLYRATE, and SCC-DFTB programs. Three independent free energy simulation modules were implemented in version 22. As detailed in Section III.D., a large number of implicit solvation and implicit membrane models have been incorporated into the energy code. They are: PBEQ, EEF1, ACE, SASA, GENBORN, GBMV, GBSW, COSMO, SCPISM, FACTS, GB/IM, IMM1, and their variants. Parallelization of CPU intensive code began as early as 1992. The current version supports a variety of parallel platforms based on SOCKETS, PVM, MPI, LAMMPI, and MPICH. In 2003, CHARMM was modified to accommodate simulations of systems as large as 10^{10} atoms. Segment, residue, atom type, and residue ID names were expanded to eight characters. The data file format was also expanded in a manner that ensures backward compatibility. The changes were implemented in c30a2x, finalized in c31a1, and released in c31b1.

XI.B. Testing

An essential requirement for efficient code development and porting to new machine and processor architectures is the avail-

Table 5. Chronology of Developmental and Release Versions of CHARMM Since 1990.

Year	Developmental	Release	New features[a]
1991		c22.0.b, c22.0.b1	BLOCK, PERT, TSM
1992		c22, c22g1, c22g2	QUANTUM, CRYSTAL
	C23al, c23a2		Parallel code, TNPACK
1993	C23f, c23f1, c23f2		FLECS to FORTRAN 77 conversion, RISM, MMFP, REPLICA
1994	C24a1, c24a2	c23f3, c23f4	Cluster[b], GAMESS interface, SSBP, CVS
1995	C24a3	c23f5	FMA, 4D dynamics, DIMB
	C25a0	c24b1	PBOUND
1996	C25a1, c25a2	c24b2, c24g1	MMFF, PBEQ
1997	C25a3	c24g2	Lambda dynamics, CADPAC interface
	C26a1	c25b1	MBO(N)D[c]
1998	C26a2	c25b1	LONEPAIR, GALGOR
	C27a1	c26b1	MC, EEF1, ACE, ADUMB, CFF
1999	C27a2	c26b2	BYCBIM, BYCC
	C28a1	c27b1	GHO
2000	C28a2, c28a3	c27b2, c27b3	POLYRATE interface, HQBMD, TMD, GAMESS-UK interface
2001	C28a4	c27b4	
	C29a1	c28b1	SASA
2002	C29a2	c28b2	
	C30a1	c29b0, c29b1	CMAP, GBMV, EMAP, SCC-DFTB
2003	C30a2, c30a2x	c29b2	CHEQ, EXPAND
	C31a1	c30b1	GBSW, GCMC, TREK, SGLD, TPS Q-Chem
2004	C31a2	c30b2	SCPISM, BNM, *DTSC*
	C32a1	c31b1	IPS
2005	C32a2	c31b2	
	C33a1	c32b1	*PBCUBES, APBS*, GSBP, PIPF
2006	C33a2	c32b2	PHMD, RUSH, SQUANTM
	C34a1	c33b1	*TAMD, SMA, CORSOL, PROTO*
2007	C34a2	c33b2	ZEROM
	C35a1	c34b1	PNM, FACTS, *CROSS*, LOOKUP, *RXNCONS*, MSCALE

[a]For features not described in text (italics), see documentation for details.
[b]Clustering analysis in the CORREL module.
[c]No longer supported.

ability of an effective suite of test cases. Test cases are continuously added to CHARMM to test newly implemented features across various platforms and machine types and also to provide users with example input files. In addition, old test cases are used to test newly added methods or features for compatibility with the rest of the code. This is done by verifying that the new CHARMM code generates the expected results for the old testcases. In the "test" directory, subdirectories corresponding to each CHARMM version contain test case input files for the features that were added in that version. The "test/data" subdirectory contains data files needed to run the test cases. In c34b1/test, there are 460 test case input files contained in 21 subdirectories.

Modifying the potential energy function requires extensive testing of its derivatives. A basic test for the coding of potential energy functions is to verify that the analytical forces $\mathbf{F}_i$ are consistent with the variation of the total potential energy $E(\mathbf{r}_1,\mathbf{r}_2,\ldots,\mathbf{r}_i,\ldots,\mathbf{r}_N)$. In CHARMM, this can be tested explicitly using the *TEST FIRSt* command, which compares the analytical forces to the finite-difference estimates of the forces; for the latter, the x-component for the ith atom is given as:

$$\mathbf{F}_{i,x} = \lim_{\Delta x \to \mathbf{0}} \frac{E(\mathbf{r}_1, \mathbf{r}_2, \ldots, x_i - \Delta x/2, y_i, z_i, \ldots, \mathbf{r}_N) - E(\mathbf{r}_1, \mathbf{r}_2, \ldots, x_i + \Delta x/2, y_i, z_i, \ldots, \mathbf{r}_N)}{\Delta x} \quad (24)$$

This test is clearly essential for the proper function of energy minimization algorithms, the correct dynamical propagation in MD simulations, and the accuracy and consistency of free energy difference calculations. Running *TEST FIRSt*, preferably with several values of Δx, is particularly important when new terms are added to the potential energy (e.g., RMSD restraints, QM/MM interactions, PBEQ forces, etc.), to ensure that the analytical energy gradient has been coded correctly. In addition, *TEST FIRSt* allows the perturbation of the unit cell within the CRYSTAL facility, as is required for the testing of the virial computation. The analogous *TEST SECOnd* command is used to test components of the Hessian compu-

tation against the finite differences of the gradient. A variant of this code is used to calculate the Hessian by finite differences when the analytic second derivatives are not available (*DIAG FINIte* subcommand of *VIBRan*).

XI.C. CHARMM Distribution and Usage

The usage of the CHARMM program in the scientific community can be measured in a number of ways. From 2002 to August 2007, a total of 714 academic CHARMM licenses were issued through Harvard University. The number of active CHARMm (commercial version) licenses issued by Accelrys as of early 2007 was ~400; this included 20 government licenses, and the rest were about evenly split between academic and commercial institutions. (In many cases, a single institutional license issued by Accelrys represents multiple end-user licenses.) According to the Science Citation Index, as of January 2009 the original (1983) CHARMM article had been cited ~7800 times and the two other articles describing the CHARMM force field[38,62] an additional 3000 times. The total number has grown steadily since the 1983 publication and now averages ~700/year.

XI.D. The CHARMM Web Site and Documentation

Charmm.org

In 2003, the Web site http://www.charmm.org was created to serve the community of CHARMM users and developers. This Web site contains basic information, links to CHARMM developers' homepages and resources, and the CHARMM forums. It is an active Web site and is expected to remain an important and up-to-date resource for CHARMM users and developers. The most heavily used areas of the Web site are the forums, where CHARMM-related discussions take place on a variety of topics; moderators volunteer their time to assist novice users and answer questions. There are currently more than 1100 registered users who have posted more than 7000 messages in 30 regular forums arranged in the following five major groups:

User Discussion and Questions—General CHARMM usage forum.

CHARMM Interfaces—Discussions regarding the use of CHARMM with other programs.

CHARMM Community—News, events, bug reports, and suggestions.

CHARMM Information—General CHARMM information and searchable documentation.

Restricted Discussion—Communication among developers.

CHARMM Documentation

The CHARMM documentation consists of a set of text files in the "doc" subdirectory of all CHARMM distributions that are also available as HTML files on the CHARMM Web site. Commands and features of all methods are documented, with descriptions of syntax, options, and usage. Examples of their use are also provided in many specific cases, along with some theoretical background and implementation details. The CHARMM Developer Guide ("developer.doc") provides basic programming information for CHARMM developers. It describes the program's organization, coding standards and rules, documentation standards, developer tools, preprocessor function and usage, compilation procedures, and code submission protocols. All of the ".doc" text files are written in the info format and can be read with the emacs editor. These info document files can also be converted into HTML files for web browsers with the "support/htmldoc/doc2html.com" script. In addition, CHARMM lecture notes are available on the charmm.org Web site. They are derived from a course that was first given at Harvard by a group of CHARMM developers in 1982 and that has been updated and presented at a variety of locations over the years, primarily at the NIH. Notes for roughly half the lectures are available. Readers who wish to obtain practical experience with CHARMM are referred to *A Guide to Biomolecular Simulations* by O. Becker and M. Karplus,[704] which is based on a course in Molecular Biophysics that was given at Harvard for several years.

XII. Concluding Discussion

The primary purpose of the current article has been to review the developments in the CHARMM program that have taken place since the initial CHARMM publication.[22] In addition, the article has discussed some of the theory and principles on which the method developments are based and many of the biomolecular research problems to which they have been applied. A review of this length, which represents a body of work spanning more than 25 years and encompassing contributions from hundreds of individual scientists, would be impossible to summarize in a few concluding paragraphs. However, there are several useful observations that can be made from an overview of the entire article. These concluding observations all center on the role of complexity in biomolecular simulation. Their consideration is relevant not only to the development and use of CHARMM, but also to biomolecular simulations more generally. It provides some guidance for the investigator in applying CHARMM and other programs to problems of interest involving macromolecular systems, and suggests a framework for thinking about the problems, themselves.

The first set of observations relates to the utility of simple models. As computational speed continues to increase, the tendency in biomolecular simulations is to use ever more complex potential energy functions that describe systems in greater detail, presumably with higher accuracy. Early extended-atom models were followed by polar hydrogen models and then all-atom models. More recently, polarizable models have been introduced, and even QM (first-principle) energy functions are used in some cases. For the representation of the aqueous environment around biomolecules, the development of implicit solvation models has followed a corresponding progression, which began with simple distance-dependent dielectric functions. Surface-area based models were then developed, and these have led in turn to more complicated representations of the solvation energy density. The latter are now being partly superseded by more accurate models, e.g., ones using an approximate or full PB electrostatics treatment of the solvent. At the same time, there has been the development of explicit representations of aqueous solvent, from van der Waals spheres to more sophisticated multipoint charge and

polarizable water models. As is demonstrated throughout the course of this paper and as is evident from the published literature, the more detailed or complex models are important. What is equally noteworthy, however, is that their existence does not necessarily displace the simpler models, which often continue to be used.

There are several reasons for this. The most obvious reason is that simple models tend to be faster or more efficient than complex ones. For a given set of computational resources, the simpler model in most cases offers the possibility of addressing a larger problem. An example is seen in MD simulations that are carried out with QM potential energy functions, e.g., when molecular mechanics potentials are not adequate. For large systems, full QM simulations are currently very limited in their utility for obtaining meaningful statistics (accessible simulation times are on the order of ps), because of the computational cost. A more useful approach, which is employed, for example, in studies of chemical reactions catalyzed by proteins, is based on QM/MM methods. It provides a suitable compromise: the parts of the system where the electronic structure changes of interest occur are treated with quantum mechanics and the rest of system is treated with (classical) molecular mechanics. At the other extreme of the scale of molecular simulations, "coarse-graining" methods have been used increasingly in recent years. They introduce simplifications that eliminate many or all of the individual atoms and thereby run counter to the trend of ever-increasing detail in simulation methodology. Coarse-graining enables simulations of very large systems, such as multimeric protein complexes, for which atomic level detail cannot be obtained experimentally, or for which obtaining similar results with an atomistic simulation requires much greater computational resources. An example is the use of an elastic network model to perform a normal mode calculation on the structure of a large multimeric protein complex obtained from cryo-electron microscopy data.

There are also less obvious reasons why simple models continue to be used. One is that the approximations that are inherent in the simpler model may be more appropriate, given the other aspects of a calculation. A good example of this involves the representation of solvent in structure prediction studies (e.g., MC studies or grid searches), in which there may be large displacements of the solute (e.g., protein) of interest at each step in the calculation. The use of explicit representations of solvent, i.e., individual water molecules, which generally provide the most detailed treatment of solvation effects, is, for practical purposes, often incompatible with such methods, because it can lead to bad solute–solvent contacts in a high fraction of the sampled solute conformations. In contrast, the use of any implicit solvation model—even the simplest surface-area based ones—circumvents this problem, because the relaxation of the aqueous environment around the solute is effectively instantaneous. Another reason for the use of simple models is that the data they generate are often more easily interpreted. For example, implicit solvation models introduce an effective free energy of solvation through a mean field approximation, which represents an average over the many degrees of freedom of the explicit solvent water molecules that would otherwise be present in the calculation. Another example is seen in the analysis of pairwise atomic electrostatic interactions, which is generally more straightforward with the use of a simple point-charge model than it is with a full QM potential energy function. Overall, the success of models at many different levels of complexity, as described throughout this article, underscores the principle that use of the simplest model capturing the essential features of the system or process under study may optimize the investigator's chance of obtaining and interpreting the data necessary to achieve useful insights.

A second set of observations in the paper concerns the complexity of methods and the systems to which they can be applied. Some of the methods described in the paper for application to large biomolecular systems were formulated for smaller systems. An example is a straightforward MD simulation, which can be successfully "scaled" from small systems to large ones essentially by increasing the number of atoms. It might be tempting to hypothesize, from this type of observation, that if a computational method is well formulated and has been validated on small systems, it should be directly applicable to large systems as well. However, the majority of methods in CHARMM, many of which are discussed in this article, have been specifically developed or modified for application to large, biologically relevant molecules—i.e., they differ significantly from related methods developed for small or homogeneous systems. For example, energy-based search facilities for small molecules did not have, nor did they require, the range of functionality possessed by the analogous facilities in CHARMM (e.g., the Monte Carlo or grid search modules). The study of large systems has also provided the main impetus for the development of more sophisticated path sampling techniques, solvation models, and free energy methods.

A prime example of the inadequacy of "simple scaling" can be found in the application of reaction path methods. If the simple methods for finding reaction pathways in small chemical reactions were directly applicable to conformational changes in proteins, most of the methods in Section VII would be unnecessary; but in fact, many reaction path methods that appear promising when tested on small systems (e.g., the alanine dipeptide) fail in proteins or other large systems. This is due in part to the fact that adequate sampling in large, inhomogeneous or asymmetric systems is qualitatively more difficult to achieve than in most small systems. The computational cost for a single step of a given sampling method will, at best, grow linearly with the number of atoms included, so that a given number of sampling steps is substantially more costly when performed for a whole protein, say, than a small drug-like molecule. Moreover, the size of the conformational space of a molecule grows exponentially with the number of degrees of freedom, so that far more steps are required to sample the same fraction of conformations for larger systems. In addition to the sampling problem, large conformational fluctuations (e.g., in protein folding), the effects of bulk solvation, and the contribution of entropic changes are much more important, in absolute energetic terms, in transition paths of large systems than in most small molecule reactions. A separate but related example is that small molecules have a much more uniform solvent exposure than large globular molecules, which have interior or buried regions. In the latter, the most accurate implicit solvation models must take into account both the direct interaction with the solvent and the dielectric

effects, as a function of the solvent exposure of different regions of the molecule, which can also vary with conformation. Finally, even a "straightforward" classical MD simulation of a very large system such as a solvated multimeric protein will likely differ from that of a small or homogeneous system, if for no other reason than the calculation must be parallelized for meaningful statistics to be obtained in an acceptable length of (real) time. As illustrated by these examples, a principal reason why CHARMM has evolved into such a multifaceted program is that large, complex systems are qualitatively "different," and their study requires its own set of methods.

A third set of observations involves the "simplicity" of the CHARMM program itself and the important role it has had in the program's capacity to grow. This article makes clear that one of the features that has been vital to the success of CHARMM as a tool for molecular biophysics research is its ability to incorporate new methods and functions. There are at least two major factors in its ability to accomplish this. First, although the program has evolved to become quite large and complicated, its global organizational structure remains relatively simple, in accord with Figure 1. One advantage of this simplicity is that the structure is more easily understood, modified, and expanded upon. As mentioned in the Introduction, CHARMM has been able to develop over the years without requiring large-scale reorganization. Although the code has of course undergone continual modifications and improvements, the basic structure dates almost to its inception three decades ago. The other factor, which is related, is that while CHARMM is to some extent modular, it lacks the complex structural coding hierarchies that characterize formally object-oriented programs. This exacts a certain cost, e.g., with regard to data encapsulation, but the benefit is transparency. Both of these types of organizational simplicity have "lowered the barrier"—not to imply that it is negligible—to the introduction of new methods, functions, and other modifications into the program over the years. In this sense, the complexity of CHARMM as it stands today, i.e., its diversity of function and its capacity to continue to expand, can be said to have arisen in large part from the simplicity of its design.

XIII. Epilogue: The History and Future of CHARMM (Martin Karplus)

XIII.A. Historical Perspective on CHARMM and Its Evolution

It is of interest to document why and how a program such as CHARMM, which has involved the sustained efforts of a large group of people for many years, came into existence. Initially, the primary purpose of the program was to provide the group at Harvard with a vehicle for doing research. It is to the credit of the group of researchers who originally developed the program that much of their early work has served as a foundation for the subsequent growth of CHARMM into a rich research tool used by the global scientific community. In an academic setting, like that at Harvard, there is no permanent support staff to take on the task of program development in an organized fashion. One of the strengths of academic scientific research in America, in contrast to that in much of Europe, is the independence of assistant professors and the intellectual renewal that is brought about by graduate students and postdoctoral fellows, who then move on to their own positions. However, the lack of a permanent staff causes some difficulties. I realized that in my research group, the only way to preserve program developments by individuals working on a many different research projects with the common thread of a focus on microscopic and mesoscopic systems (e.g., from small molecules in solution to large proteins) was to have an all-encompassing program like CHARMM. The price of having a single program is, of course, the complexity that comes with size, but CHARMM is now a major research tool for the scientific community in large part because of this diversity of function. The modularity of the program has made it possible to adjust relatively easily to new demands and new possibilities. The CHARMM Development Project, which is administratively at Harvard University but involves all of the developers, is a continuing, collaborative effort to advance the CHARMM program as a state-of-the-art tool for macromolecular simulations. It is one of the great successes of the project that many persons have been able to work together to develop the program over a 30-year period (see Table 3) and that the structure is in place to continue the developments into the foreseeable future.

CHARMM began with a program, now referred to as "Pre-CHARMM," which was developed by Bruce Gelin during his years (1967–1976) as a graduate student in the Chemistry Department at Harvard University.[705] He had begun to do theoretical work in molecular quantum mechanics and started by studying the application of the random-phase approximation to two-electron problems. He was collaborating with Neal Ostlund who was a postdoctoral fellow at Harvard at the time. Soon, however, Gelin was drafted and, as a member of the Military Police, ended up in a laboratory that was concerned with drug use (LSD, etc.) in the US Army. This aroused his interest in biology and when he returned to Harvard to finish his degree, he wanted to change his area of research to deal with biological problems. This fitted in well with my own interests. Attila Szabo had just finished a statistical mechanical model of hemoglobin cooperativity[706] that was based on crystallographic studies and their interpretation by Max Perutz. This work raised a number of questions concerning the energetics of ligand binding in hemoglobin and its coupling to protein structural changes involved in the transition from the unliganded to the liganded state (the T to R transition). The best approach to such a problem was to have available a way of calculating the energy of the protein as a function of the atomic positions. The specific objective of Gelin's research was to introduce the effect of ligand binding on the heme group as a perturbation (undoming of the heme) and to use energy minimization to determine the response of the protein. To do such a calculation on the available computers (an IBM 7090 at Columbia University was our workhorse at the time) required considerable courage and a program with which one could construct the energy function for a protein as large as a single hemoglobin chain (about 145 amino acid residues in length). We did not have such a program and Gelin began to develop software that would make it possible to start out with a given amino acid sequence (e.g., that of the hemoglobin alpha

chain) and a set of coordinates (e.g., those obtained from the X-ray structure of deoxyhemoglobin) and to use this information to calculate the energy of the system and its derivatives as a function of the coordinates. Developing such a program was a major task, but Gelin had just the right combination of abilities to carry it out. The result was Pre-CHARMM (it did not have a name at that time). Although not trivial to use, the program was applied to a variety of problems, including Gelin's pioneering study of aromatic ring flips in the bovine pancreatic trypsin inhibitor,[23] as well as the hemoglobin study already mentioned,[707] and Dave Case's analysis on ligand escape after photodissociation in myoglobin.[708] This work predated the MD simulation of BPTI,[4] which served as the basis for the application of such simulation methods to a wide-range of problems in structural biology.[11–13]

Gelin would have had a very difficult time constructing such a program if there had not been prior work by other groups on protein energy calculations. The two major inputs came from Schneior Lifson's group at the Weizmann Institute in Rehovot and Harold Scheraga's group at Cornell University. When I first decided to take up calculational approaches to biology, I needed a place where I could work with a good library and a congenial group of people who knew more about what I wanted to do than I did. I took a leave from Harvard University in the fall semester of 1968 and went to join Shneior Lifson's group at the Weizmann Institute in Rehovot for 6 months. There I met Arieh Warshel who came to Harvard as a postdoctoral fellow and brought his CFF program.[86] At Harvard, he developed a program for what would now be called π-electron QM/MM calculations for the ground and excited states of polyenes.[244] His presence and the availability of the program was an important resource for Gelin, who was also aware of Michael Levitt's pioneering protein energy calculations.[709] For the choice of the energy function to represent a protein and for many of the parameters used in the original extended atom model (all H atoms were treated implicitly), the work of Scheraga's group, and in particular, the studies of Gibson and Scheraga,[710] were an invaluable resource.

It soon became evident that for an evergrowing group of research uses, it would be very important to have a program that was easier to use, adapt, and develop. This need led to the first version of the present CHARMM program, by the authors of the 1983 article.[22] Each one had a different background and different ideas about how to develop the best program. As a result of many discussions, some rather heated, the first version of the program was born. When we searched for a name for the program, we tried to find something for which GANDALF could be an acronym; my daughter Reba was at the time very much involved with the stories by Tolkien. This was unsuccessful; so, Bob Bruccoleri, one of the original CHARMM developers, came up with the name HARMM (HArvard Macromolecular Mechanics), which might have served as a warning for the uninitiated user but seemed inappropriate to me. The addition of 'C' for Chemistry led to the present name.

Because of the growing importance of macromolecular simulations in drug design by pharmaceutical companies, an entrepreneurial lawyer, Jeff Wales and his neighbor, Andy Ferrara, came to me in 1985 with the idea of establishing a company that was based on distributing the CHARMM program to industry. This seemed a good idea, particularly because the original concept was that Harvard would make the CHARMM program available and the company, initially called Polygen, would transform our academic tool into a commercial program. Only part of the plan came to fruition: i.e., what has been distributed over the years by the various incarnations of the company (Polygen, Molecular Simulations, Inc., and now Accelrys, Inc.) has been the Harvard program, with few changes other than the introduction of license keys. However, the graphical programs QUANTA and INSIGHT have been of considerable utility as front-ends to CHARMM, particularly for inexperienced users. Recently, Accelrys has begun to contribute to CHARMm and CHARMM in the same way as other "developers." An example is the GB-based implicit solvation model for membranes.[136] Also, Accelrys has developed a number of scripts, particularly for side chain and loop predictions (see www.accelrys.com for details).

One major concern I had in working out the arrangements with Polygen was that the academic distribution of CHARMM remain under Harvard's (my) control. This was important to me because I wanted to keep the research aspect of CHARMM clear of interference by commercial objectives and to make certain that the program could be distributed at a reasonable price for academic and other (e.g., government) not-for-profit institutions. Toward the latter goal, the criterion I decided on was that the price should be as low as possible, but high enough so that people would request the CHARMM program only if they had a genuine intention of using it, rather than merely wanting to add another program to their collection. To distinguish the academic and commercial versions, which I hoped would be significantly different, as mentioned earlier, the slightly different names—CHARMM (academic) and CHARMm (commercial)—were agreed upon.

At about this time, I met Rod Hubbard who was very impressed with the possibilities of macromolecular simulations and had the idea of developing a graphics program to illustrate the results. I invited him to come to Harvard, where he developed a program, called HYDRA for its seven modules or "heads." It was an exciting project. Every day, Hubbard would show on the computer screen what he had developed overnight, and group members would try and use it, find the problems in the present version, and suggest new functionalities that would be helpful in research. In this way, mainly through Hubbard's outstanding ability at graphical programming, a very useful graphical program was developed in record time. It is unfortunate that this paradigm is not followed more generally to avoid programs that please the developers but not the prospective users. The graphical interface program QUANTA, which was developed from HYDRA by Rod Hubbard and people at Polygen, has remained an important tool for users of CHARMM until now.

CHARMM has "evolved" for more than 30 years, and the community of CHARMM developers is now sufficiently dispersed that there is an annual meeting to discuss recent additions and developments. It begins with 1 or 2 days during which the developers present recent work. (There are 30 or more presentations.) This is followed by a half-day session during which the content of the next developmental version of CHARMM is discussed, and the parts of the existing developmental version that

will be added to the release version are selected. Usually, new developmental and release versions are generated each year in August, with an update incorporating bug fixes released in February. The critical task of integrating the various developer contributions while resolving conflicts and ensuring standard coding practices is led by Youngdo Won, the CHARMM manager (see also Section XI), who assumes the ultimate responsibility for preparing the new versions.

One contribution of CHARMM, in addition to its function as a simulation program, is that a number of other programs for macromolecular simulations are direct, though not necessarily planned, descendants of CHARMM; for example, Paul Weiner brought pre-CHARMM to Peter Kollman's group and developed the first version of AMBER from it. Similarly, Wilfred van Gunsteren was a postdoc in my group, took pre-CHARMM with him and used it as a basis for GROMOS. These programs, and many others that are less-widely available but had their origins in CHARMM, are now independently developed and each one has certain features that make it unique. Finally, X-PLOR was a planned derivative of CHARMM. It began while Axel Brünger was at Harvard, when the utility of MD in a simulated annealing mode for X-ray structure refinement and NMR structure determination became clear.[711] The great success of X-PLOR, and now CNS and CNX, has been due in large part to Axel Brünger, their primary developer.

XIII.B. Perspectives for the Future

There are two components to the future of CHARMM, one administrative and the other scientific. For both, the future looks bright. On the administrative side, a plan is in place for an executive committee (Bernard R. Brooks, Charles L. Brooks III, and Martin Karplus) to formally take charge of the program and its evolution at the appropriate time. To achieve this, an agreement between Harvard, as the copyright holder of the program, and two other institutions (NIH for Bernard Brooks and University of Michigan for Charles Brooks) has been codified. In this way, it is expected that the development and distribution of the CHARMM program will continue as it has in the past.

On the scientific side, it is appropriate to begin by quoting from the Concluding Discussion of the original CHARMM article[22]:

"Our work focuses on the chemistry of condensed phases, with particular emphasis on the study of macromolecular systems found in biology. The program has been employed in projects ranging from the exploration of macromolecular solvation to protein–DNA interactions and many associated studies of constituent small-molecule properties. The very large size and lack of symmetry of these systems presents us with challenging computational requirements. The methods developed to deal with these demands have application in other areas of theoretical (e.g., fluid and polymer mechanics) and experimental (e.g., crystallography, structure refinement, NMR, and other spectroscopy interpretation) study. By simulating biological macromolecules, we hope to improve our understanding of their properties and of the forces acting within them. Such knowledge will in turn help to elucidate their function and the mechanisms involved in macromolecular structure and assembly, binding site recognition, and specificity. Enzymes are among the most efficient and versatile catalysts known. The chemical and physical understanding of proteins gained through simulation will be directly applicable to understanding these unique catalysts. Combined molecular orbital and empirical energy function calculations are planned to examine the detailed interaction of molecular mechanics with electronic structure. Nucleic acids and their transformations, which play an essential role in genetics, are being studied."

Much of what was written 25 years ago is still valid today and most of the research listed as "in preparation" in the 1983 article has been completed, published, and incorporated into the CHARMM program. One important example is the development and widespread application of QM/MM methodology.

Given the great and continuing increase in computer power (the first petaflop machine has recently been reported), simulations will most likely evolve in several ways. As I describe below, the extensions to larger systems and longer simulation times is one direction. In addition, the fact that multiple simulations can be done as a routine matter makes possible the determination of statistical errors in the results. In reducing systematic errors, the use of more accurate and complex force fields (e.g., polarization, QM/MM) will likely play a role. Also, faster computations will aid in the development of improved models of biological phenomena, because shorter turnaround times for nanosecond simulations will permit the testing of more ideas. Moreover, the possibility of more accurate calculations, including free energy simulations, using generalized force fields should be instrumental in making computer-aided ligand design a reality.

An exciting recent development in MD is that the simulation time scales becoming available with modern computers (100 ns to μs or even longer[243]) are making it possible to directly simulate biologically important events. This is analogous, in an inverse sense, to the fact that while experiments on the ps time scale were an important development, it was only when the time resolution was extended to femtoseconds that the actual events involved in chemical reactions could be observed.[712,713] A striking recent result is that, by running multiple simulations of 10 ns duration, the visualization of water molecules migrating through a model of the aquaporin channel has been achieved (see Fig. 9).[714,715] Another example is the observation in MD simulations of the formation of detergent micelles[681,682] and phospholipid bilayers.[716] That certain of these simulations were done with other programs (e.g., GROMACS[21] and NAMD[693]) shows how much the field has matured. It is becoming ever more evident that cells are made up not of isolated proteins, but of protein complexes, which have the essential functional roles. The structures of such large multisubunit complexes are being determined at an increasing rate. In all of them (they are almost all "molecular machines") conformational change is directly involved in function. One example where such simulations have helped to elucidate the mechanism, in this case the synthesis of ATP, is the use of free energy and targeted MD simulations of the enzyme ATP synthase.[124,717] Another complex that is now being studied by molecular and normal mode dynamics is the ribosome, whose structure was determined recently. The simulation of such large systems for the time required to obtain meaningful results is now possible and broadens the role of simulation programs like CHARMM in molecular biophysics.

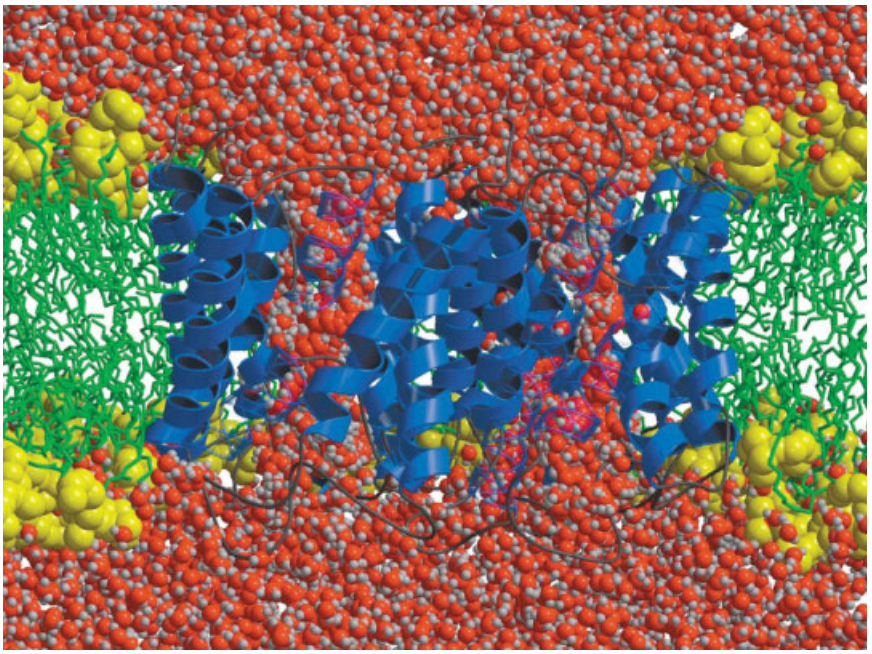

Figure 9. Water molecules migrating through a model of the aquaporin channel, depicted by a superposition of 100 snapshots from a 10-ns dynamics trajectory. The aquaporin tetramer is shown in blue and the lipid bilayer membrane in which it is embedded is shown in yellow (head groups) and green (hydrocarbon tails).[714]

The next step is the evolution of MD simulations from molecular and supramolecular systems to the cellular scale. Studies of the formation of such assemblages will be more demanding. The simulation of more complex cellular activities, such as synaptic transmission[718] and the dismantling of the nuclear membrane on cell division by the motor protein cytoplasmic dynein[719] are two examples of interest. Much of this work will build on the detailed knowledge of the structure and dynamics of the channels, enzymes and other cellular components. Global simulations are likely to be initiated with less detailed models. A recent example is provided by the use of simplified normal mode calculations for the cowpea chlorotic mottle virus as a way of interpreting low resolution (28 Å) cryoelectron microscope data indicating the swelling of this virus at low pH,[720] or dynamics of processes involved in ribosomal translocation.[721] However, the ultimate descriptions, which will necessarily include such details as the possible effects of mechanical stress in a contracting neuromuscular synapse on its channels and other components, will require atomistic simulations.

Given the continuing improvements in MD simulations, another development will be their routine use by experimentalists as a tool, like any other, for improving the interpretation and understanding of the data. This has, of course, been true for many years as part of high-resolution structure determinations[301,711] and it is now beginning to occur in the interpretation of the structural results by the scientists who obtained them.[722] When MD is a routine part of structural biology, it will become clearer what refinements and extensions of the methodology are most needed to improve the results and to perfect the constructive interplay between the simulations and experiment. The exposure of limitations by such applications will, in turn, provide challenges for the simulation experts, and catalyze new developments in the field. I hope that before long such an interplay between experiments and simulations will be an integral part of molecular biology, as it is now in chemistry.

Acknowledgments

In a multiauthor article, there is a legitimate concern that the people involved receive the credit that they deserve. In general, this is not possible without listing the contributions of each individual, as some journals are now requiring. For an article of this length and complexity, however, any attempt at such specific attribution of credit is impractical. All the authors contributed to the writing and rewriting of significant portions of the text. The corresponding authors, designated by asterisks, were also involved in planning the manuscript and overseeing sections in the early stages of the writing. In both groups (starred and unstarred), the listing is alphabetical. One author, R.J. Petrella, needs to be mentioned individually because, in addition to writing a significant portion of the article, he was instrumental in transforming a large number of separate write-ups into what is very nearly a unified whole.

The authors thank the referees for their helpful comments and David A. Case for serving as the editor of the paper. A number of people, other than those in the author list, have read and commented on the manuscript. They include Kwangho Nam, Arjan van der Vaart, Ioan Andricioaei, and Tom Darden.

In addition to all of the authors of the paper, many other scientists have participated significantly in the development of CHARMM through the years. See Table 3; this list is included with all distributions of the program (in "charmm_main.src").

Support for the development of CHARMM, *per se*, and for researchers concerned with CHARMM development, have come from many sources, including NSF, NIH, DOE, Accelrys, and CNRS. It is not possible to list all of the grants individually, but NIH grant RR023920 is acknowledged for its direct support of the ongoing CHARMM conversion project. Part of the research in the B.R. Brooks group was supported by the Intramural Research Program of the NIH, NHLBI.

References

1. Alder, B. J.; Wainwright, T. E. J Chem Phys 1957, 27, 1208.
2. Rahman, A. Phys Rev 1964, 136, 405.
3. Rahman, A.; Stillinger, F. H. J Chem Phys 1971, 55, 3336.
4. McCammon, J. A.; Gelin, B. R.; Karplus, M. Nature 1977, 267, 585.
5. Hockney, R. W.; Eastwood, J. W. Computer Simulation Using Particles; McGraw-Hill: New York, 1981.
6. McCammon, J. A.; Harvey, S. Dynamics of Proteins and Nucleic Acids; Cambridge University Press: Cambridge, 1987.
7. Brooks, C. L., III; Karplus, M.; Pettitt, B. M. Proteins: A Theoretical Perspective of Dynamics, Structure, and Thermodynamics; Wiley: New York, 1988.
8. Allen, M. P.; Tildesley, D. J. Computer Simulation of Liquids; Oxford University Press: Oxford, 1989.
9. van Gunsteren, W. F.; Weiner, P. K.; Wilkinson, A. J., Eds. Computer Simulation of Biomolecular Systems. Theoretical and Experimental Applications; ESCOM: Leiden, 1993.
10. Becker, O. M.; MacKerell, A. D., Jr.; Roux, B.; Watanabe, M., Eds. Computational Biochemistry and Biophysics; Marcel Dekker: New York, 2001.
11. Karplus, M.; McCammon, J. A. Nat Struct Biol 2002, 9, 646.
12. Karplus, M.; Barbara, P., Eds. Molecular Dynamics Simulations of Biomolecules, Accts Chem Res 2002.
13. Karplus, M.; Kuriyan, J. Proc Natl Acad Sci USA 2005, 102, 6679.
14. Yang, W.; Gao, Y. Q.; Cui, Q.; Ma, J.; Karplus, M. Proc Natl Acad Sci USA 2003, 100, 874.
15. Mao, H. Z.; Weber, J. Proc Natl Acad Sci USA 2007, 104, 18478.
16. Banks, J. L.; Beard, H. S.; Cao, Y. X.; Cho, A. E.; Damm, W.; Farid, R.; Felts, A. K.; Halgren, T. A.; Mainz, D. T.; Maple, J. R.; Murphy, R.; Philipp, D. M.; Repasky, M. P.; Zhang, L. Y.; Berne, B. J.; Friesner, R. A.; Gallicchio, E.; Levy, R. M. J Comput Chem 2005, 26, 1752.
17. Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. J Comput Chem 2005, 26, 1781.
18. Christen, M.; Hünenberger, P. H.; Bakowies, D.; Baron, R.; Bürgi, R.; Geerke, D. P.; Heinz, T. N.; Kastenholz, M. A.; Kräutler, V.; Oostenbrink, C.; Peter, C.; Trzesniak, D.; van Gunsteren, W. F. J Comput Chem 2005, 26, 1719.
19. Jorgensen, W. L.; Tirado-Rives, J. J Comput Chem 2005, 26, 1689.
20. Case, D. A.; Cheatham, T. E., III; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M., Jr.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. J Comput Chem 2005, 26, 1668.
21. van der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. C. J Comput Chem 2005, 26, 1701.
22. Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S. J Comput Chem 1983, 4, 187.
23. Gelin, B. R.; Karplus, M. Proc Natl Acad Sci USA 1975, 72, 2002.
24. Brooks, B.; Karplus, M. Proc Natl Acad Sci USA 1983, 80, 6571.
25. Jo, S.; Kim, T.; Iyer, V. G.; Im, W. J Comput Chem 2008, 29, 1859.
26. Miller, B. T.; Singh, R. P.; Klauda, J. B.; Hodoscek, M.; Brooks, B. R.; Woodcock, H. L. J Chem Inf Model 2008, 48, 1920.
27. Brooks, C. L.; Karplus, M. J Chem Phys 1983, 79, 6312.
28. Beglov, D.; Roux, B. J Chem Phys 1994, 100, 9050.
29. Im, W., Berneche, S., Roux, B. J Chem Phys 2001, 114, 2924.
30. Archontis, G.; Simonson, T.; Karplus, M. J Mol Biol 2001, 306, 307.
31. Woolf, T. B.; Roux, B. Proc Natl Acad Sci USA 1994, 91, 11631.
32. Berneche, S.; Nina, M.; Roux, B. Biophys J 1998, 75, 1603.
33. Bernèche, S.; Roux, B. Nature 2001, 414, 73.
34. Lüdemann, S. K.; Lounnas, V.; Wade, R. C. J Mol Biol 2000, 303, 797.
35. Carlsson, P.; Burendahl, S.; Nilsson, L. Biophys J 2006, 91, 3151.
36. Blondel, A.; Renaud, J. P.; Fischer, S.; Moras, D.; Karplus, M. J Mol Biol 1999, 291, 101.
37. Pellarin, R.; Caflisch, A. J Mol Biol 2006, 360, 882.
38. MacKerell, A. D., Jr.; Bashford, D.; Belott, 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.; Wiórkiewicz-Kuczera, J.; Yin, D.; Karplus, M. J Phys Chem B 1998, 102, 3586.
39. Foloppe, N.; MacKerell, A. D., Jr. J Comput Chem 2000, 21, 86.
40. MacKerell, A. D., Jr.; Banavali, N. K. J Comput Chem 2000, 21, 105.
41. MacKerell, A. D., Jr.; Banavali, N.; Foloppe, N. Biopolymers 2000, 56, 257.
42. Morozov, A. V.; Kortemme, T.; Tsemekhman, K.; Baker, D. Proc Natl Acad Sci USA 2004, 101, 6946.
43. Buckingham, A. D.; Fowler, P. W.; Hutson, J. M. Chem Rev 1988, 88, 963.
44. Morse, P. M. Phys Rev 1929, 34, 57.
45. Gelin, B. R.; Karplus, M. Biochemistry 1979, 18, 1256.
46. Nilsson, L.; Karplus, M. J Comput Chem 1986, 7, 591.
47. Reiher, W. E., III. Theoretical Studies of Hydrogen Bonding, PhD Thesis, Chemistry Department, Harvard University: Cambridge, MA, 1985.
48. Neria, E.; Fischer, S.; Karplus, M. J Chem Phys 1996, 105, 1902.
49. MacKerell, A. D., Jr. In Computational Biochemistry and Biophysics; Becker, O. M.; MacKerell, A. D., Jr.; Roux, B.; Watanabe, M., Eds.; Marcel Dekker: New York, 2001, pp 7–38.
50. Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. J Chem Phys 1983, 79, 926.
51. Durell, S. R.; Brooks, B. R.; Ben-Naim, A. J Phys Chem 1994, 98, 2198.
52. Mark, P.; Nilsson, L. J Phys Chem A 2001, 105, 9954.
53. Mark, P.; Nilsson, L. J Comp Chem 2002, 23, 1211.
54. Höchtl, P.; Boresch, S.; Bitomsky, W.; Steinhauser, O. J Chem Phys 1998, 109, 4927.
55. Pettitt, B. M.; Karplus, M. Chem Phys Lett 1985, 121, 194.
56. MacKerell, A. D., Jr.; Brooks, B. R.; Brooks, C. L., III; Nilsson, L.; Roux, B.; Won, Y.; Karplus, M. In Encyclopedia of Computational Chemistry; Schleyer, P. V. R.; Allinger, N. L.; Clark, T.; Gasteiger, J.; Kollman, P. A.; Schaefer, H. F., III; Schreiner, P. R., Eds.; Wiley: Chichester, UK, 1998, pp 271–277.
57. Feller, S. E.; MacKerell, A. D., Jr. J Phys Chem B 2000, 104, 7510.

58. Feller, S. E.; Gawrisch, K.; MacKerell, A. D., Jr. J Am Chem Soc 2002, 124, 318.
59. Klauda, J. B.; Brooks, B. R.; MacKerell, A. D., Jr.; Venable, R. M.; Pastor, R. W. J Phys Chem B 2005, 109, 5300.
60. Kuttel, M.; Brady, J. W.; Naidoo, K. J. J Comput Chem 2002, 23, 1236.
61. Guvench, O.; Greene, S. N.; Kamath, G.; Brady, J. W.; Venable, R. M.; Pastor, R. W.; Mackerell, A. D., Jr. J Comput Chem 2008, 29, 2543.
62. MacKerell, A. D., Jr.; Wiorkiewicz-Kuczera, J.; Karplus, M. J Am Chem Soc 1995, 117, 11946.
63. Schlenkrich, M.; Brickmann, J.; MacKerell, A. D., Jr.; Karplus, M. In Biological Membranes: A Molecular Perspective from Computation and Experiment; Merz, K. M.; Roux, B., Eds.; Birkhauser: Boston, 1996, pp. 31–81.
64. Feng, M. H.; Philippopoulos, M.; MacKerell, A. D., Jr.; Lim, C. J Am Chem Soc 1996, 118, 11265.
65. Pavelites, J. J.; Gao, J.; Bash, P. A.; Mackerell, A. D., Jr. J Comput Chem 1997, 18, 221.
66. Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. J Am Chem Soc 1995, 117, 2309.
67. Cheatham, T. E.; Cieplak, P.; Kollman, P. A. J Biomol Struct Dyn 1999, 16, 845.
68. Jorgensen, W. L.; Tirado-Rives, J. J Am Chem Soc 1988, 110, 1666.
69. Price, D. J.; Brooks, C. L., III. J Comput Chem 2002, 23, 1045.
70. Langley, D. R. J Biomol Struct Dyn 1998, 16, 487.
71. Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; Hermans, J., Eds. Interaction Models for Water in Relation to Proteins Hydration; Reidel: Dordrecht, 1981.
72. Berendsen, H. J. C.; Grigera, J. R.; Straatsma, T. P. J Phys Chem 1987, 91, 6269.
73. Stillinger, F. H.; Rahman, A. J Chem Phys 1974, 60, 1545.
74. MacKerell, A. D., Jr.; Feig, M.; Brooks, C. L., III. J Comp Chem 2004, 25, 1400.
75. Shirts, M. R.; Pitera, J. W.; Swope, W. C.; Pande, V. S. J Chem Phys 2003, 119, 5740.
76. Deng, Y.; Roux, B. J Phys Chem B 2004, 108, 16567.
77. Foloppe, N.; Nilsson, L. J Phys Chem B 2005, 109, 9119.
78. Reddy, S. Y.; Leclerc, F.; Karplus, M. Biophys J 2003, 84, 1421.
79. Priyakumar, U. D.; MacKerell, A. D., Jr. J Chem Theory Comput 2006, 2, 187.
80. Priyakumar, U. D.; MacKerell, A. D., Jr. J Am Chem Soc 2006, 128, 678.
81. Banavali, N. K.; MacKerell, A. D., Jr. J Mol Biol 2002, 319, 141.
82. Halgren, T. A. J Comput Chem 1996, 17, 490.
83. Halgren, T. A. J Comput Chem 1999, 20, 730.
84. Maple, J. R.; Hwang, M. J.; Stockfisch, T. P.; Dinur, U.; Waldman, M.; Ewig, C. S.; Hagler, A. T. J Comput Chem 1994, 15, 162.
85. Maple, J. R.; Hwang, M. J.; Jalkanen, K. J.; Stockfisch, T. P.; Hagler, A. T. J Comput Chem 1998, 19, 430.
86. Lifson, S.; Warshel, A. J Chem Phys 1969, 49, 5116.
87. MacKerell, A. D., Jr.; Feig, M.; Brooks, C. L., III. J Am Chem Soc 2004, 126, 698.
88. Li, X. F.; Hassan, S. A.; Mehler, E. L. Proteins 2005, 60, 464.
89. Buck, M.; Bouguet-Bonnet, S.; Pastor, R. W.; MacKerell, A. D., Jr. Biophys J 2006, 90, L36.
90. Chen, J.; Won, H. S.; Im, W.; Dyson, H. J.; Brooks, C. L., III. J Biomol NMR 2005, 31, 59.
91. Allen, T. W.; Andersen, O. S.; Roux, B. Proc Natl Acad Sci USA 2004, 101, 117.
92. Patel, S.; Brooks, C. L., III. Proc Natl Acad Sci USA 2008, 105, 10378.
93. Rick, S. W.; Stuart, S. J.; Berne, B. J. J Chem Phys 1994, 101, 6141.
94. Itskowitz, P.; Berkowitz, M. L. J Phys Chem A 1997, 101, 5687.
95. Dick, B. G.; Overhauser, A. W. Physical Review 1958, 112, 90.
96. Gao, J.; Pavelites, J. J.; Habibollazadeh, D. J Phys Chem 1996, 100, 2689.
97. Thole, B. T. Chem Phys 1981, 59, 341.
98. Xie, W. S.; Pu, J. Z.; MacKerell, A. D., Jr.; Gao, J. L. J Chem Theory Comput 2007, 3, 1878.
99. Patel, S.; Brooks, C. L., III. J Comput Chem 2004, 25, 1.
100. Patel, S.; MacKerell, A. D., Jr.; Brooks, C. L., III. J Comp Chem 2004, 25, 1504.
101. Patel, S.; Brooks, C. L., III. J Chem Phys 2005, 123, 164502.
102. Patel, S.; Brooks, C. L., III. J Chem Phys 2005, 122, 24508.
103. Patel, S.; Brooks, C. L., III. J Chem Phys 2006, 124, 204706.
104. Lamoureux, G.; Roux, B. J Chem Phys 2003, 119, 3025.
105. Lamoureux, G.; MacKerell, A. D., Jr.; Roux, B. J Chem Phys 2003, 119, 5185.
106. Lamoureux, G.; Harder, E.; Vorobyov, I. V.; Deng, Y.; Roux, B.; MacKerell, A. D., Jr. Chem Phys Lett 2006, 418, 245.
107. Anisimov, V. M.; Lamoureux, G.; Vorobyov, I. V.; Huang, N.; Roux, B.; MacKerell, A. D., Jr. J Chem Theory Comput 2005, 1, 153.
108. Vorobyov, I. V.; Anisimov, V. M.; MacKerell, A. D., Jr. J Phys Chem B 2005, 109, 18988.
109. Noskov, S. Y.; Lamoureux, G.; Roux, B. J Phys Chem B 2005, 109, 6705.
110. Anisimov, V. M.; Vorobyov, I. V.; Roux, B.; MacKerell, A. D., Jr. J Chem Theory Comput 2007, 3, 1927.
111. Lopes, P. E. M.; Lamoureux, G.; Roux, B.; MacKerell, A. D., Jr. J Phys Chem B 2007, 111, 2873.
112. Vorobyov, I.; Anisimov, V. M.; Greene, S.; Venable, R. M.; Moser, A.; Pastor, R. W.; MacKerell, A. D., Jr. J Chem Theory Comput 2007, 3, 1120.
113. Harder, E.; Anisimov, V. M.; Whitfield, T. W.; MacKerell, A. D., Jr.; Roux, B. J Phys Chem B 2008, 112, 3509.
114. Lamoureux, G.; Roux, B. J Phys Chem B Condens Matter Mater Surf Interfaces Biophys 2006, 110, 3308.
115. Archontis, G.; Leontidis, E. Chem Phys Lett 2006, 420, 199.
116. Archontis, G.; Leontidis, E.; Andreou, G. J Phys Chem B 2005, 109, 17957.
117. Chang, T. M.; Dang, L. X. Chem Rev 2006, 106, 1305.
118. Jungwirth, P.; Tobias, D. J. Chem Rev 2006, 106, 1259.
119. Gao, J.; Habibollazadeh, D.; Shao, L. J Phys Chem 1995, 99, 16460.
120. Tuckerman, M. E.; Martyna, G. J. J Phys Chem B 2000, 104, 159.
121. Schaefer, M.; Bartels, C.; Karplus, M. J Mol Biol 1998, 284, 835.
122. Feig, M.; Brooks, C. L., III. Curr Opin Struct Biol 2004, 14, 217.
123. Ferrara, P.; Caflisch, A. Proc Natl Acad Sci USA 2000, 97, 10780.
124. Ma, J.; Flynn, T. C.; Cui, Q.; Leslie, A.; Walker, J. E.; Karplus, M. Structure 2002, 10, 921.
125. Kollman, P. A. Chem Rev 1993, 93, 2395.
126. Rod, T. H.; Brooks, C. L., III. J Am Chem Soc 2003, 125, 8718.
127. Lee, M. S.; Salsbury, F. R., Jr.; Brooks, C. L., III. Proteins 2004, 56, 738.
128. Mongan, J.; Case, D. A.; McCammon, J. A. J Comput Chem 2004, 25, 2038.
129. Khandogin, J.; Brooks, C. L., III. Biophys J 2005, 89, 141.
130. Lazaridis, T.; Karplus, M. J Mol Biol 1999, 288, 477.
131. Dominy, B. N.; Brooks, C. L., III. J Comput Chem 2001, 23, 147.
132. Feig, M.; Brooks, C. L., III. Proteins 2002, 49, 232.
133. Fiser, A.; Feig, M.; Brooks, C. L., III; Sali, A. Acc Chem Res 2002, 35, 413.
134. Im, W.; Lee, M. S.; Brooks, C. L., III. J Comput Chem 2003, 24, 1691–1702.
135. Lazaridis, T. Proteins 2003, 52, 176.

136. Spassov, V. Z.; Yan, L.; Szalma, S. J Phys Chem B 2002, 106, 8726.
137. Im, W.; Brooks, C. L., III. J Mol Biol 2004, 337, 513.
138. Im, W.; Chen, J.; Brooks, C. L., III. Adv Protein Chem 2006, 72, 173.
139. Lazaridis, T. Proteins 2005, 58, 518.
140. Roux, B.; Yu, H. A.; Karplus, M. J Phys Chem 1990, 94, 4683.
141. Habtemariam, B.; Anisimov, V. M.; MacKerell, A. D., Jr. Nucl Acid Res 2005, 33, 4212.
142. Paci, E.; Karplus, M. J Mol Biol 1999, 288, 441.
143. Steinbach, P. J. Protein Struct Funct Genet 2004, 57, 665.
144. Stultz, C. M. J Phys Chem B 2004, 108, 16525.
145. Huang, A.; Stultz, C. M. Biophys J 2007, 92, 34.
146. Lee, B.; Richards, F. M. J Mol Biol 1971, 55, 379.
147. Eisenberg, D.; McLachlan, A. D. Nature 1986, 319, 199.
148. Wesson, L.; Eisenberg, D. Protein Sci 1992, 1, 227.
149. Ferrara, P.; Apostolakis, J.; Caflisch, A. Proteins 2002, 46, 24.
150. Hasel, W.; Hendrickson, T. F.; Still, W. C. Tetrahedron Comput Methodol 1988, 1, 103.
151. Lazaridis, T.; Karplus, M. Protein Struct Funct Genet 1999, 35, 133.
152. Reddy, V. S.; Giesing, H. A.; Morton, R. T.; Kumar, A.; Post, C. B.; Brooks, C. L., III; Johnson, J. E. Biophys J 1998, 74, 546.
153. Ferrara, P.; Apostolakis, J.; Caflisch, A. Protein Struct Funct Genet 2000, 39, 252.
154. Cavalli, A.; Haberthur, U.; Paci, E.; Caflisch, A. Protein Sci 2003, 12, 1801.
155. Paci, E.; Cavalli, A.; Vendruscolo, M.; Caflisch, A. Proc Natl Acad Sci USA 2003, 100, 8217.
156. Settanni, G.; Rao, F.; Caflisch, A. Proc Natl Acad Sci USA 2005, 102, 628.
157. Gsponer, J.; Caflisch, A. Proc Natl Acad Sci USA 2002, 99, 6719.
158. Rathore, N.; Yan, Q.; de Pablo, J. J. J Chem Phys 2004, 120, 5781.
159. Gsponer, J.; Haberthur, U.; Caflisch, A. Proc Natl Acad Sci USA 2003, 100, 5154.
160. Paci, E.; Gsponer, J.; Salvatella, X.; Vendruscolo, M. J Mol Biol 2004, 340, 555.
161. Cecchini, M.; Curcio, R.; Pappalardo, M.; Melki, R.; Caflisch, A. J Mol Biol 2006, 357, 1306.
162. Guvench, O.; Brooks, C. L., III. J Comput Chem 2004, 25, 1005.
163. Privalov, P. L.; Makhatadze, G. I. J Mol Biol 1993, 232, 660.
164. Makhatadze, G. I.; Privalov, P. L. J Mol Biol 1993, 232, 639.
165. Dinner, A. R.; Lazaridis, T.; Karplus, M. Proc Natl Acad Sci USA 1999, 96, 9068.
166. Krivov, S. V.; Karplus, M. Proc Natl Acad Sci USA 2004, 101, 14766.
167. Paci, E.; Caflisch, A.; Pluckthun, A.; Karplus, M. J Mol Biol 2001, 314, 589.
168. Lazaridis, T.; Karplus, M. Science 1997, 278, 1928.
169. Inuzuka, Y.; Lazaridis, T. Protein Struct Funct Genet 2000, 41, 21.
170. Paci, E.; Karplus, M. Proc Natl Acad Sci USA 2000, 97, 6521.
171. Duan, J.; Nilsson, L. Proteins 2005, 59, 170.
172. Kumar, S.; Sham, Y. Y.; Tsai, C. J.; Nussinov, R. Biophys J 2001, 80, 2439.
173. Levy, Y.; Jortner, J.; Becker, O. M. Proc Natl Acad Sci USA 2001, 98, 2188.
174. Lazaridis, T.; Karplus, M. Biophysical Chemistry 1999, 78, 207.
175. Petrella, R. J.; Karplus, M. J Phys Chem B 2000, 104, 11370.
176. Lazaridis, T. Curr Org Chem 2002, 6, 1319.
177. Gray, J. J.; Moughon, S.; Wang, C.; Schueler-Furman, O.; Kuhlman, B.; Rohl, C. A.; Baker, D. J Mol Biol 2003, 331, 281.
178. Zanghellini, A.; Jiang, L.; Wollacott, A. M.; Cheng, G.; Meiler, J.; Althoff, E. A.; Rothlisberger, D.; Baker, D. Protein Sci 2006, 15, 2785.
179. Masunov, A.; Lazaridis, T. J Am Chem Soc 2003, 125, 1722.
180. Mottamal, M.; Lazaridis, T. Biochemistry 2005, 44, 1607.
181. Hassan, S. A.; Guarnieri, F.; Mehler, E. L. J Phys Chem B 2000, 104, 6478.
182. Hassan, S. A.; Mehler, E. L.; Zhang, D.; Weinstein, H. Protein Struct Funct Genet 2003, 51, 109.
183. Hassan, S. A. J Phys Chem B 2004, 108, 19501.
184. Chandler, D.; Andersen, H. C. J Chem Phys 1972, 57, 1930.
185. Yu, H. A.; Karplus, M. J Chem Phys 1988, 89, 2366.
186. Imai, T.; Hiraoka, R.; Kovalenko, A.; Hirata, F. Proteins 2007, 66, 804.
187. Fogolari, F.; Zuccato, P.; Esposito, G.; Viglino, P. Biophys J 1999, 76, 1.
188. Warwicker, J.; Watson, H. C. J Mol Biol 1982, 157, 671.
189. Klapper, I.; Hagstrom, R.; Fine, R.; Sharp, K.; Honig, B. Proteins 1986, 1, 47.
190. Kovalenko, A.; Hirata, F. J Chem Phys 2000, 112, 10391.
191. Im, W.; Beglov, D.; Roux, B. Comput Phys Commun 1998, 111, 59.
192. Nina, M.; Beglov, D.; Roux, B. J Phys Chem B 1997, 101, 5239.
193. Banavali, N. K.; Roux, B. J Phys Chem B 2002, 106, 11026.
194. Elcock, A. H.; McCammon, J. A. J Am Chem Soc 1996, 118, 3787.
195. Norberg, J. Arch Bioch Bioph 2003, 410, 48.
196. Foloppe, N.; Fisher, L. M.; Howes, R.; Kierstan, P.; Potter, A.; Robertson, A. G. S.; Surgenor, A. E. J Med Chem 2005, 48, 4332.
197. Roux, B. Biophys J 1997, 73, 2980.
198. Chanda, B.; Asamoah, O. K.; Blunck, R.; Roux, B.; Bezanilla, F. Nature 2005, 436, 852.
199. Gallicchio, E.; Levy, R. M. J Comput Chem 2004, 25, 479.
200. Tan, C.; Tan, Y. H.; Luo, R. J Phys Chem B 2007, 111, 12263.
201. Klamt, A.; Schüürmann, G. J Chem Soc Perkin Trans 1993, 22, 799.
202. Dolney, D. M.; Hawkins, G. D.; Winget, P.; Liotard, D. A.; Cramer, C. J.; Truhlar, D. G. J Comput Chem 2000, 21, 340.
203. Klamt, A. J Phys Chem 1995, 99, 2224.
204. Klamt, A.; Jonas, V.; Burger, T.; Lohrenz, J. C. W. J Phys Chem A 1998, 102, 5074.
205. York, D. M.; Karplus, M. J Chem Phys 1999, 103, 11060.
206. Thiel, W. MNDO97, version 5.0, University of Zurich, Zurich, Switzerland, 1998.
207. Gregersen, B. A.; Khandogin, J.; Thiel, W.; York, D. M. J Phys Chem B 2005, 109, 9810.
208. Khandogin, J.; Gregersen, B. A.; Thiel, W.; York, D. M. J Phys Chem B 2005, 109, 9799.
209. Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. J Amer Chem Soc 1990, 112, 6127.
210. Constanciel, R.; Contreras, R. Theor Chim Acta 1984, 65, 1.
211. Qiu, D.; Shenkin, P. S.; Hollinger, F. P.; Still, W. C. J Phys Chem A 1997, 101, 3005.
212. Hawkins, G. D.; Cramer, C. J.; Truhlar, D. G. Chem Phys Lett 1995, 246, 122.
213. Schaefer, M.; Karplus, M. J Phys Chem 1996, 100, 1578.
214. Bashford, D.; Case, D. A. Ann Rev Phys Chem 2000, 51, 129.
215. Ghosh, A.; Rapp, C. S.; Friesner, R. A. J Phys Chem B 1998, 102, 10983.
216. Lee, M. S.; Salsbury, F. R., Jr.; Brooks, C. L., III. J Chem Phys 2002, 116, 10606.
217. Lee, M. S.; Feig, M.; Salsbury, F. R., Jr.; Brooks, C. L., III. J Comput Chem 2003, 24, 1348.
218. Calimet, N.; Schaefer, M.; Simonson, T. Proteins 2001, 45, 144.
219. Dominy, B. N.; Brooks, C. L., III. J Phys Chem B 1999, 103, 3765.

220. Bursulaya, B. D.; Brooks, C. L., III. J Phys Chem B 2000, 104, 12378.
221. Ohkubo, Y. Z.; Brooks, C. L., III. Proc Natl Acad Sci USA 2003, 100, 13916.
222. Noskov, S. Y.; Lim, C. Biophys J 2001, 81, 737.
223. Simonson, T.; Archontis, G.; Karplus, M. J Phys Chem B 1997, 101, 8349.
224. Feig, M.; Onufriev, A.; Lee, M. S.; Im, W.; Case, D. A.; Brooks, C. L., III. J Comp Chem 2004, 25, 265.
225. Ferrara, P.; Gohlke, H.; Price, D.; Klebe, G.; Brooks, C. L., III. J Med Chem 2004, 47, 3032.
226. Borgis, D.; Lévy, N.; Marchi, M. J Chem Phys 2003, 119, 3516.
227. Haberthur, U.; Caflisch, A. J Comput Chem 2008, 29, 701.
228. Schaefer, M.; Bartels, C.; Leclerc, F.; Karplus, M. J Comput Chem 2001, 22, 1857.
229. Chen, J.; W., I.; Brooks, C. L., III. J Am Chem Soc 2006, 128, 3728.
230. Roux, B. Curr Opin Struct Biol 2002, 12, 182.
231. Im, W.; Feig, M.; Brooks, C. L., III. Biophys J 2003, 85, 2900.
232. Tanizaki, S.; Feig, M. J Chem Phys 2005, 122, 124706.
233. Im, W.; Brooks, C. L., III. Proc Natl Acad Sci USA 2005, 102, 6771.
234. Bu, L.; Im, W.; Brooks, C. L., III. Biophys J 2007, 92, 854.
235. Mottamal, M.; Lazaridis, T. Biophys Chem 2006, 122, 50.
236. Lazaridis, T. J Chem Theory Comput 2005, 1, 716.
237. Mottamal, M.; Zhang, J.; Lazaridis, T. Proteins 2006, 62, 996.
238. Mihajlovic, M.; Lazaridis, T. J Phys Chem B 2006, 110, 3375.
239. Bashford, D.; Karplus, M. Biochemistry 1990, 29, 10219.
240. Porat, A.; Lillig, C. H.; Johansson, C.; Fernandes, A. P.; Nilsson, L.; Holmgren, A.; Beckwith, J. Biochemistry 2007, 46, 3366.
241. van Vlijmen, H. W. T.; Schaefer, M.; Karplus, M. Protein Struct Funct Genet 1998, 33, 145.
242. Foloppe, N.; Sagemark, J.; Nordstrand, K.; Berndt, K. D.; Nilsson, L. J Mol Biol 2001, 310, 449.
243. Foloppe, N.; Nilsson, L. Structure 2004, 12, 289.
244. Warshel, A.; Karplus, M. J Am Chem Soc 1972, 94, 5612.
245. Warshel, A.; Levitt, M. J Mol Biol 1976, 103, 227.
246. Singh, U. C.; Kollman, P. A. J Comput Chem 1986, 7, 718.
247. Gao, J. In Rev Comput Chem; Lipkowitz, K. B.; Boyd, D. B., Eds.; VCH: New York, 1995; pp. 119–185.
248. Bash, P. A.; Field, M. J.; Karplus, M. J Am Chem Soc 1987, 109, 8092.
249. Field, M. J.; Bash, P. A.; Karplus, M. J Comput Chem 1990, 11, 700.
250. Stewart, J. J. P. Quant Chem Prog Exch 1990, 10, 86.
251. Bash, P. A.; Field, M. J.; Davenport, R. C.; Petsko, G. A.; Ringe, D.; Karplus, M. Biochemistry 1991, 30, 5826.
252. Chatfield, D. C.; Eurenius, K. P.; Brooks, B. R. Theochem 1998, 423, 79.
253. Cui, Q.; Karplus, M. J Am Chem Soc 2000, 220, HYS-501.
254. Cui, Q.; Karplus, M. J Am Chem Soc 2002, 124, 3093.
255. Alhambra, C.; Wu, L.; Zhang, Z.-Y.; Gao, J. J Am Chem Soc 1998, 120, 3858.
256. Gao, J.; Truhlar, D. G. Annu Rev Phys Chem 2002, 53, 467.
257. Rajamani, R.; Gao, J. J Comput Chem 2002, 23, 96.
258. Gao, J.; Ma, S. H.; Major, D. T.; Nam, K.; Pu, J. Z.; Truhlar, D. G. Chem Rev 2006, 106, 3188.
259. Garcia-Viloca, M.; Gao, J.; Karplus, M.; Truhlar, D. G. Science 2004, 303, 186.
260. Cui, Q.; Elstner, M.; Kaxiras, E.; Frauenheim, T.; Karplus, M. J Phys Chem B 2001, 105, 569.
261. Gregersen, B. A.; Lopez, X.; York, D. M. J Am Chem Soc 2004, 126, 7504.
262. Reuter, N.; Dejaegere, A.; Maigret, B.; Karplus, M. J Phys Chem A 2000, 104, 1720.
263. Gao, J.; Amara, P.; Alhambra, C.; Field, M. J. J Phys Chem A 1998, 102, 4714.
264. Maseras, F.; Morokuma, K. J Comput Chem 1995, 16, 1170.
265. Das, D.; Brooks, B. R. Biophys J 2000, 78, 1969.
266. Das, D.; Eurenius, K. P.; Billings, E. M.; Sherwood, P.; Chatfield, D. C.; Hodoscek, M.; Brooks, B. R. J Chem Phys 2002, 117, 10534.
267. Thery, V.; Rinaldi, D.; Rivail, J. L.; Maigret, B.; Ferenczy, G. G. J Comput Chem 1994, 15, 269.
268. Pu, J.; Gao, J.; Truhlar, D. G. J Phys Chem A 2004, 108, 5454.
269. Pu, J.; Gao, J.; Truhlar, D. G. J Phys Chem A 2004, 108, 632.
270. Pu, J.; Gao, J.; Truhlar, D. G. Chem Phys Chem 2005, 6, 1853.
271. Gao, J.; Xia, X. Science 1992, 258, 631.
272. Gao, J. ACS Symp Ser 1994, 569, 8.
273. Freindorf, M.; Gao, J. J Comput Chem 1996, 17, 386.
274. Riccardi, D.; Li, G.; Cui, Q. J Phys Chem B 2004, 108, 6467.
275. Murphy, R. B.; Philipp, D. M.; Friesner, R. A. J Comput Chem 2000, 21, 1442.
276. Bash, P. A.; Ho, L. L.; MacKerell, A. D., Jr.; Levine, D.; Hallstrom, P. Proc Natl Acad Sci, USA 1996, 93, 3698.
277. Mo, Y. R.; Gao, J. L. J Phys Chem B 2006, 110, 2976.
278. Hensen, C.; Hermann, J. C.; Nam, K.; Ma, S.; Gao, J.; Hoeltje, H.-D. J Med Chem 2004, 47, 6673.
279. Garcia-Viloca, M.; Truhlar, D. G.; Gao, J. J Mol Bio 2003, 327, 549.
280. Nam, K.; Gao, J.; York, D. M. J Chem Theory Comput 2005, 1, 2.
281. Schaefer, P.; Riccardi, D.; Cui, Q. J Chem Phys 2005, 123, 014905.
282. Elstner, M.; Porezag, D.; Juugnickel, G.; Elsner, J.; Haugk, M.; Frauenheim, T.; Sukai, S.; Seifect, G. Phys Rev B 1998, 58, 7260.
283. Cui, Q.; Elstner, M.; Kaxiras, E.; Frauenheim, T.; Karplus, M. J Phys Chem B 2001, 105, 569.
284. Walker, R. C.; Crowley, M. F.; Case, D. A. J Comput Chem 2008, 29, 1019.
285. Guest, M.; van Lenthe, J.; Kendrick, J.; Schoeffel, K.; Sherwood, P.; Harrison, R.; Amos, R.; Buenker, R.; Dupuis, M.; Handy, N.; Hillier, I.; Knowles, P.; Bonacic-Koutecky, V.; von Niessen, W.; Saunders, V.; Stone, A.; Spangler, D.; Wendoloski, J. NRCC Software Catalog 1980, 1.
286. Eurenius, K. P.; Chatfield, D. C.; Brooks, B. R.; Hodoscek, M. Int J Quantum Chem 1996, 60, 1189.
287. Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S.; Windus, T. L.; Dupuis, M.; Montgomery, J. A. J Comput Chem 1993, 14, 1347.
288. Amos, R. D.; Albert, I. L.; Andrews, J. S.; Colwell, D. M.; Handy, N. C.; Jayatilaka, D.; Knowles, P.; Kobayashi, R.; Laidig, K. E.; Ladn, A. M.; Lee, G. L.; Maslen, P. E.; Murray, C. W.; Rice, J. E.; Simandiras, E. D.; Stone, A. J.; Su, M. D.; Tozer, D. J.,The Cambridge Analytical Derivatives Package Issue 6, Cambridge, 1995.
289. Woodcock, H. L.; Hodoscek, M.; Gilbert, A. T. B.; Gill, P. M. W.; Schaefer, H. F., III; Brooks, B. R. J Comput Chem 2007, 28, 1485.
290. Bylaska, E. J.; Jong, W. A. D.; Govind, N.; Kowalski, K.; Straatsma, T. P.; Valiev, M.; Wang, D.; Apra, E.; Windus, T. L.; Hammond, J.; Nichols, P.; Hirata, S.; Hackler, M. T.; Zhao, Y.; Fan, P.-D.; Harrison, R. J.; Dupuis, M.; Smith, D. M. A.; Nieplocha, J.; Tipparaju, V.; Krishnan, M.; Wu, Q.; Voorhis, T. V.; Auer, A. A.; Nooijen, M.; Brown, E.; Cisneros, G.; Fann, G. I.; Fruchtl, H.; Garza, J.; Hirao, K.; Kendall, R.; Nichols, J. A.; Tsemekhman, K.; Wolinski, K.; Anchell, J.; Bernholdt, D.; Borowski, P.; Clark, T.; Clerc, D.; Dachsel, H.; Deegan, M.; Dyall, K.; Elwood, D.; Glendening, E.; Gutowski, M.; Hess, A.; Jaffe, J.; Johnson, B.; Ju, J.; Kobayashi, R.; Kutteh, R.; Lin, Z.; Littlefield,

R.; Long, X.; Meng, B.; Nakajima, T.; Niu, S.; Pollack, L.; Rosing, M.; Sandrone, G.; Stave, M.; Taylor, H.; Thomas, G.; Lenthe, J. V.; Wong, A.; Zhang, Z.; Q-Chem. Pacific Northwest National Laboratory: Richland, Washington, 2007.
291. Kendall, R. A.; Apra, E.; Bernholdt, D. E.; Bylaska, E. J.; Dupuis, M.; Fann, G. I.; Harrison, R. J.; Ju, J. L.; Nichols, J. A.; Nieplocha, J.; Straatsma, T. P.; Windus, T. L.; Wong, A. T. Comput Phys Commun 2000, 128, 260.
292. Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Montgomery, J., J. A.; Vreven, T.; Kudin, K. N.; Burant, J. C.; Millam, J. M.; Iyengar, S. S.; Tomasi, J.; Barone, V.; Mennucci, B.; Cossi, M.; Scalmani, G.; Rega, N.; Petersson, G. A.; Nakatsuji, H.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Klene, M.; Li, X.; Knox, J. E.; Hratchian, H. P.; Cross, J. B.; Bakken, V.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Ayala, P. Y.; Morokuma, K.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Zakrzewski, V. G.; Dapprich, S.; Daniels, A. D.; Strain, M. C.; Farkas, O.; Malick, D. K.; Rabuck, A. D.; Raghavachari, K.; Foresman, J. B.; Ortiz, J. V.; Cui, Q.; Baboul, A. G.; Clifford, S.; Cioslowski, J.; Stefanov, B. B.; Liu, G.; Liashenko, A.; Piskorz, P.; Komaromi, I.; Martin, R. L.; Fox, D. J.; Keith, T.; Al-Laham, M. A.; Peng, C. Y.; Nanayakkara, A.; Challacombe, M.; Gill, P. M. W.; Johnson, B.; Chen, W.; Wong, M. W.; Gonzalez, C.; Pople, J. A. Gaussian 03; Gaussian: Wallingford, CT, 2004.
293. Werner, H.-J.; Knowles, P. J.; Lindh, R.; Manby, F. R.; Schütz, M.; Celani, P.; Korona, T.; Rauhut, G.; Amos, R. D.; Bernhardsson, A.; Berning, A.; Cooper, D. L.; Deegan, M. J. O.; Dobbyn, A. J.; Eckert, F.; Hetzer, C. H. a. G.; Lloyd, A. W.; McNicholas, S. J.; Mura, W. M. a. M. E.; Nicklass, A.; Palmieri, P.; Pitzer, R.; Schumann, U.; Stoll, H.; Stone, A. J.; Tarroni, R.; Thorsteinsson, T. Available at: www.molpro.net, 2006.
294. Kong, J.; White, C. A.; Krylov, A. I.; Sherrill, C. D.; Adamson, R. D.; Furlani, T. R.; Lee, M. S.; Lee, A. M.; Gwaltney, S. R.; Adams, T. R.; Ochsenfeld, C.; Gilbert, A. T. B.; Kedziora, G. S.; Rassolov, V. A.; Maurice, D. R.; Nair, N.; Shao, Y.; Besley, N. A.; Maslen, P. E.; Dombroski, J. P.; Daschel, H.; Zhang, W.; Korambath, P. P.; Baker, J.; Byrd, E. F. C.; Voorhis, T. V.; Oumi, M.; Hirata, S.; Hsu, C. P.; Ishikawa, N.; Florian, J.; Warshel, A.; Johnson, B. G.; Gill, P. M. W.; Head-Gordon, M.; Pople, J. A. J Comput Chem 2000, 21, 1532.
295. Shao, Y.; Molnar, L. F.; Jung, Y.; Kussmann, J.; Ochsenfeld, C.; Brown, S. T.; Gilbert, A. T. B.; Slipchenko, L. V.; Levchenko, S. V.; O'Neill, D. P.; DiStasio, R. A.; Lochan, R. C.; Wang, T.; Beran, G. J. O.; Besley, N. A.; Herbert, J. M.; Lin, C. Y.; Van Voorhis, T.; Chien, S. H.; Sodt, A.; Steele, R. P.; Rassolov, V. A.; Maslen, P. E.; Korambath, P. P.; Adamson, R. D.; Austin, B.; Baker, J.; Byrd, E. F. C.; Dachsel, H.; Doerksen, R. J.; Dreuw, A.; Dunietz, B. D.; Dutoi, A. D.; Furlani, T. R.; Gwaltney, S. R.; Heyden, A.; Hirata, S.; Hsu, C. P.; Kedziora, G.; Khalliulin, R. Z.; Klunzinger, P.; Lee, A. M.; Lee, M. S.; Liang, W.; Lotan, I.; Nair, N.; Peters, B.; Proynov, E. I.; Pieniazek, P. A.; Rhee, Y. M.; Ritchie, J.; Rosta, E.; Sherrill, C. D.; Simmonett, A. C.; Subotnik, J. E.; Woodcock, H. L.; Zhang, W.; Bell, A. T.; Chakraborty, A. K.; Chipman, D. M.; Keil, F. J.; Warshel, A.; Hehre, W. J.; Schaefer, H. F., III; Kong, J.; Krylov, A. I.; Gill, P. M. W.; Head-Gordon, M. Phys Chem Chem Phys 2006, 8, 3172.
296. Northrup, S. H.; Pear, M. R.; Lee, C.-Y.; McCammon, J. A.; Karplus, M. Proc Natl Acad Sci USA 1982, 79, 4035.
297. Chu, J.-W.; Trout, B. L.; Brooks, B. R. J Chem Phys 2003, 119, 12708.
298. Boczko, E. M.; Brooks, C. L., III. Science 1995, 269, 393.
299. Sheinerman, F. B.; Brooks, C. L., III. Proc Natl Acad Sci USA 1998, 95, 1562.
300. Shea, J. E.; Brooks, C. L., III. In Annu Rev Phys Chem; Strauss, H. L., Ed.; Annual Reviews: Palo Alto, 2001, 499.
301. Nilsson, L.; Clore, G. M.; Gronenborn, A. M.; Brunger, A. T.; Karplus, M. J Mol Biol 1986, 188, 455.
302. Chen, J.; Im, W.; Brooks, C. L., III. J Am Chem Soc 2004, 126, 16038.
303. Karplus, M. J Am Chem Soc 1963, 85, 2870.
304. Scott, W. R. P.; Mark, A. E.; van Gunsteren, W. F. J Biomolecular NMR 1998, 12, 501.
305. Brooks, C. L., III; Karplus, M. In Methods in Enzymology; Packer, L., Ed. Academic Press: New York, 1986, pp. 369–400.
306. Feller, S. E.; Pastor, R. W.; Rojnuckarin, A.; Bogusz, S.; Brooks, B. R. J Phys Chem 1996, 100, 17011.
307. Simonson, T.; Archontis, G.; Karplus, M. J Phys Chem B 1999, 103, 6142.
308. Brooks, C. L., III; Montgomery Pettitt, B.; Karplus, M. J Chem Phys 1985, 83, 5897.
309. Loncharich, R. J.; Brooks, B. R. Proteins 1989, 6, 32.
310. Steinbach, P. J.; Brooks, B. R. J Comput Chem 1994, 15, 667.
311. Norberg, J.; Nilsson, L. Biophys J 2000, 79, 1537.
312. Verlet, L. Phys Rev 1969, 159, 98.
313. Yip, V.; Elber, R. J Comput Chem 1989, 10, 921.
314. Petrella, R. J.; Andricioaei, I.; Brooks, B. R.; Karplus, M. J Comput Chem 2003, 24, 222.
315. Stote, R. H.; States, D. J.; Karplus, M. J Chim Phys 1991, 88, 2419.
316. Archontis, G.; Simonson, T.; Moras, D.; Karplus, M. J Mol Biol 1998, 275, 823.
317. Greengard, L.; Rokhlin, V. J Comput Phys 1987, 73, 325.
318. Petrella, R. J.; Karplus, M. J Comput Chem 2005, 26, 755.
319. Lague, P.; Pastor, R. W.; Brooks, B. R. J Phys Chem B 2004, 108, 363.
320. Nilsson, L.; Halle, B. Proc Natl Acad Sci USA 2005, 102, 13867.
321. Carlsson, P.; Koehler, K. F.; Nilsson, L. Mol Endocrinol 2005, 19, 1960.
322. Kitao, A.; Hayward, S.; Go, N. Proteins-Structure Function and Genetics 1998, 33, 496.
323. Mohanty, D.; Elber, R.; Thirumalai, D.; Beglov, D.; Roux, B. J Mol Biol 1997, 272, 423.
324. Marchand, S.; Roux, B. Protein Struct Funct Genet 1998, 33, 265.
325. Nina, M.; Simonson, T. J Phys Chem B 2002, 106, 3696.
326. Banavali, N. K., Im, W., Roux, B. J Chem Phys 2002, 117, 7381.
327. Riccardi, D.; Schaefer, P.; Yang, Y.; Yu, H.; Ghosh, N.; Prat-Resina, X.; Konig, P.; Li, G.; Xu, D.; Guo, H.; Elstner, M.; Cui, Q. J Phys Chem B Condens Matter Mater Surf Interfaces Biophys 2006, 110, 6458.
328. Speelman, B.; Brooks, B. R.; Post, C. B. Biophys J 2001, 80, 121.
329. Garen, J.; Field, M. J.; Kneller, G. R.; Karplus, M.; Smith, J. C. Journal de Chimie Physique 1991, 11–12, 2587.
330. Dolan, E. A.; Venable, R. M.; Pastor, R. W.; Brooks, B. R. Biophys J 2002, 82, 2317.
331. Nosé, S. J Chem Phys 1984, 81, 511.
332. Ewald, P. P. Annalen der Physik 1921, 64, 253.
333. de Leeuw, S. W.; Perram, J. W.; Smith, E. R. Proc R Soc London Ser A 1980, 373, 57.
334. Smith, E. R. Proc R Soc London Ser A 1981, 375, 475.
335. Darden, T.; York, D.; Pedersen, L. J Chem Phys 1993, 98, 10089.
336. Essman, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. J Chem Phys 1995, 103, 8577.
337. Bogusz, S.; Cheatham, T. E.; Brooks, B. R. J Chem Phys 1998, 108, 7070.

338. Figueirido, F.; DelBueno, G. S.; Levy, R. M. J Phys Chem B 1997, 101, 5622.
339. Hummer, G.; Pratt, L. R.; Garcia, A. E. J Chem Phys 1997, 107, 9275.
340. Zhang, Y.; Feller, S.; Brooks, B. R.; Pastor, R. J Chem Phys 1995, 103, 10252.
341. Pollock, E. L.; Glosli, J. Comput Phys Commun 1996, 95, 93.
342. Hodoscek, M.; Billings, E. M.; Cheatham, T. E.; Brooks, B. R. In Proceedings of the International Symposium on Supercomputing: New Horizons of Computational Science; Kluwer Academic, 1999.
343. Wu, X.; Brooks, B. R. J Chem Phys 2005, 122, 44107.
344. Takahashi, K.; Yasuoka, K.; Narumi, T. J Chem Phys 2007, 127: 114511.
345. Klauda, J. B.; Wu, X. W.; Pastor, R. W.; Brooks, B. R. J Phys Chem B 2007, 111, 4393.
346. Derreumaux, P.; Zhang, G. H.; Schlick, T.; Brooks, B. J Comput Chem 1994, 15, 532.
347. Levy, R. M.; Perahia, D.; Karplus, M. Proceedings of the National Academy of Sciences of the United States of America-Physical Sciences 1982, 79, 1346.
348. Zheng, C.; Wong, C. F.; McCammon, J. A.; Wolynes, P. G. Nature 1988, 334, 726.
349. Berens, P. H.; Mackay, D. H. J.; White, G. M.; Wilson, K. R. J Chem Phys 1983, 79, 2375.
350. Martyna, G. J.; Tuckerman, M. E.; Tobias, D. J.; Klein, M. L. Molecular Physics 1996, 87, 1117.
351. Watanabe, M.; Karplus, M. J Phys Chem 1995, 99, 5680.
352. Crippen, G. M. J Phys Chem 1987, 91, 6341.
353. Zhou, J.; Reich, S.; Brooks, B. R. J Chem Phys 2000, 112, 7919.
354. Ryckaert, J. P.; Ciccotti, G.; Berendsen, H. J. C. J Comput Phys 1977, 23, 327.
355. Engle, R. D.; Skeel, R. D.; Drees, M. J Comput Phys 2005, 206, 432.
356. Leimkuhler, B. J.; Skeel, R. D. J Comput Phys 1994, 112, 117.
357. Tobias, D. J.; Brooks, C. L. J Chem Phys 1988, 89, 5115.
358. Hoover, W. G. Phys Rev A 1985, 31, 1695.
359. Vitkup, D.; Ringe, D.; Petsko, G. A.; Karplus, M. Nature Struct Biol 2000, 7, 34.
360. Feller, S.; Zhang, Y.; Pastor, R.; Brooks, B. J Chem Phys 1995, 103, 4613.
361. Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; Dinola, A.; Haak, J. R. J Chem Phys 1984, 81, 3684.
362. Chandrasekhar, S. Rev Mod Phys 1943, 15, 1.
363. Brünger, A.; Brooks, C. L., III; Karplus, M. Chem Phys Lett 1984, 105, 495.
364. Pastor, R. W.; Brooks, B. R.; Szabo, A. Molecular Physics 1988, 65, 1409.
365. Janezic, D.; Venable, R. M.; Brooks, B. R. J Comp Chem 1995, 16, 1554.
366. Janezic, D.; Brooks, B. R. J Comp Chem 1995, 16, 1543.
367. Brooks, B. R.; Janezic, D.; Karplus, M. J Comput Chem 1995, 16, 1522.
368. Schwarzl, S. M.; Tschopp, T. B.; Smith, J. C.; Fischer, S. J Comput Chem 2002, 23, 1143.
369. Tidor, B.; Karplus, M. J Mol Biol 1994, 238, 405.
370. Brooks, B.; Karplus, M. Proc Natl Acad Sci USA 1985, 82, 4995.
371. Cui, Q.; Li, G.; Ma, J.; Karplus, M. J Mol Biol 2004, 340, 345.
372. Ma, J. Structure 2005, 13, 373.
373. Ma, J. P.; Karplus, M. J Mol Biol 1997, 274, 114.
374. Ma, J. P.; Karplus, M. Proc Natl Acad Sci USA 1998, 95, 8502.
375. Mouawad, L.; Perahia, D. Biopolymers 1993, 33, 599.
376. Perahia, D.; Mouawad, L. Comput Chem 1995, 19, 241.
377. Tama, F.; Gadea, F. X.; Marques, O.; Sanejouand, Y.-H. Proteins Struct Funct Genet 2000, 41, 1.
378. Li, G.; Cui, Q. Biophys J 2002, 83, 2457.
379. Li, G.; van Wynsberghe, A.; Demerdash, O. N. A.; Cui, Q. In Normal Mode Analysis: Theory and Applications to Biological and Chemical Systems; Cui, Q.; Bahar, I., Eds.; Chapman & Hall/CRC Press: Boca Raton, FL, 2006; pp. 65–89.
380. Tirion, M. Phys Rev Lett 1996, 77, 1905.
381. Atilgan, A. R.; Durell, S. R.; Jernigan, R. L.; Demirel, M. C.; Keskin, O.; Bahar, I. Biophys J 2001, 80, 505.
382. van Wynsberghe, A.; Li, G.; Cui, Q. Biochemistry 2004, 43, 13083.
383. van Wynsberghe, A. W.; Cui, Q. Biophys J 2005, 89, 2939.
384. Cui, Q.; Karplus, M. J Chem Phys 2000, 112, 1133.
385. Cui, Q.; Karplus, M. J Am Chem Soc 2001, 123, 2284.
386. Corcelli, S. A.; Lawrence, C. P.; Skinner, J. L. J Chem Phys 2004, 120, 8107.
387. Woodcock, H. L.; Zheng, W. J.; Ghysels, A.; Shao, Y.; Kong, J.; Brooks, B. R. J Chem Phys 2008, 129: 214109.
388. Andricioaei, I.; Karplus, M. J Chem Phys 2001, 115, 6289.
389. Tsui, V.; Case, D. A. J Phys Chem B 2001, 105, 11314.
390. Krzanowski, W. J. Principles of Multivariate Analysis; Oxford University Press: Oxford, 2000.
391. Amadei, A.; Kinssen, A. B. M.; Berendsen, H. J. C. Proteins 1993, 17, 412.
392. Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. J Chem Phys 1953, 21, 1087.
393. Northrup, S. H.; McCammon, J. A. Biopolymers 1980, 19, 1001.
394. Jorgensen, W. L.; Tirado–Rives, J. J Phys Chem 1996, 100, 14508.
395. Hu, J.; Ma, A.; Dinner, A. R. J Comput Chem 2006, 27, 203.
396. Eppenga, R.; Frenkel, D. Mol Phys 1984, 52, 1303.
397. Woo, H. J.; Dinner, A. R.; Roux, B. J Chem Phys 2004, 121, 6392.
398. Deng, Y. Q.; Roux, B. J Chem Phys 2008, 128, 044106.
399. Mezei, M. Mol Phys 1980, 40, 901.
400. Mezei, M. Mol Phys 1987, 61, 565.
401. Hu, J.; Ma, A.; Dinner, A. R. J Chem Phys 2006, 125, 114101.
402. Tsallis, C. J Stat Phys 1988, 52, 479.
403. Andricioaei, I. I.; Straub, J. E. Phys Rev E 1996, 53, R3055.
404. Berg, B. A.; Neuhaus, T. Phys Lett B 1991, 267, 249.
405. Berg, B. A.; Neuhaus, T. Phys Rev Lett 1992, 68, 9.
406. Wang, F. G.; Landau, D. P. Phys Rev Lett 2001, 86, 2050.
407. Calvo, F. Mol Phys 2002, 100, 3421.
408. Bartels, C.; Karplus, M. J Comput Chem 1997, 18, 1450.
409. Ma, A.; Nag, A.; Dinner, A. R. J Chem Phys 2006, 124, 144911.
410. Dinner, A. R. Monte Carlo Simulations of Protein Folding; PhD thesis, Harvard University, 1999.
411. Go, N.; Scheraga, H. A. Macromolecules 1973, 6, 273.
412. Dodd, L. R.; Boone, T. D.; Theodorou, D. N. Mol Phys 1993, 78, 961.
413. Dinner, A. R. J Comput Chem 2000, 21, 1132.
414. Duane, S.; Kennedy, A. D.; Pendleton, B. J.; Roweth, D. Phys Lett B 1987, 195, 216.
415. Mehlig, B.; Heermann, D. W.; Forrest, B. M. Phys Rev B 1992, 45, 679.
416. Andricioaei, I.; Dinner, A. R.; Karplus, M. J Chem Phys 2003, 118, 1074.
417. Bouzida, D.; Kumar, S.; Swendsen, R. H. Phys Rev A 1992, 45, 8894.
418. Li, Z. Q.; Scheraga, H. A. Proc Natl Acad Sci USA 1987, 84, 6611.
419. Abagyan, R.; Totrov, M. J Mol Biol 1994, 253, 983.
420. Petrella, R. J.; Lazaridis, T.; Karplus, M. Fold Des 1998, 3, 353.
421. Xiang, Z.; Honig, B. J Mol Biol 2001, 311, 421.
422. Petrella, R. J.; Karplus, M. J Mol Biol 2001, 312, 1161.

423. Komazin-Meredith, G.; Petrella, R. J.; Santos, W. L.; Filman, D. J.; Hogle, J. M.; Verdine, G. L.; Karplus, M.; Coen, D. M. Structure 2008, 16, 1214.
424. Brooks, C. L., III. Computer Simulations of Biomolecular Systems: Theoretical and Experimental Applications, Princeton: New Jersey, 1988; pp. 221–234.
425. Gao, J.; Kuczera, K.; Tidor, B.; Karplus, M. Science 1989, 244, 1069.
426. McCammon, J. A. Curr Opin Struct Biol 1991, 1, 196.
427. Straatsma, T. P.; McCammon, J. A. Annu Rev Phys Chem 1992, 43, 407.
428. Tidor, B.; Karplus, M. Biochemistry 1991, 30, 3217.
429. Fleischman, S. H.; Brooks, C. L., III. J Chem Phys 1987, 87, 3029.
430. Tobias, D. J.; Brooks, C. L., III. Chem Phys Lett 1987, 142, 472.
431. Simonson, T.; Archontis, G.; Karplus, M. Acc Chem Res 2002, 35, 430.
432. Beveridge, D. L.; Dicapua, F. M. Annu Rev Biophys Biophys Chem 1989, 18, 431.
433. Jorgensen, W. L. Acc Chem Res 1989, 22, 184.
434. Kollman, P. A.; Merz, K. M. Acc Chem Res 1990, 23, 246.
435. van Gunsteren, W. F.; Berendsen, H. J. C. Angewandte Chemie (International Edition in English) 1990, 29, 992.
436. Straatsma, T. P. In Reviews in Computational Chemistry; Lipkowitz, K. B.; Boyd, D. B., Eds.; VCH Publishers: New York, 1996; pp. 81–127.
437. Kirkwood, J. G. J Chem Phys 1935, 3, 300.
438. Zwanzig, R. W. J Chem Phys 1954, 22, 1420.
439. Berendsen, H. J. C.; Johan P.M. Postma, W. F.; van Gunsteren, W. In Molecular Dynamics and Protein Structure; Hermans, J.; Ed.; Polycrystal Book service: Western Springs, IL, 1985; pp. 43–46.
440. Jarzynski, C. Phys Rev Lett 1997, 78, 2690.
441. West, D. K.; Olmsted, P. D.; Paci, E. J Chem Phys 2006, 125, 204910.
442. Zacharias, M.; Straatsma, T. P.; Mccammon, J. A. J Chem Phys 1994, 100, 9025.
443. Beutler, T. C.; Mark, A. E.; van Schaik, R. C.; Gerber, P. R.; van Gunsteren, W. F. Chem Phys Lett 1994, 222, 529.
444. Steinbrecher, T.; Mobley, D. L.; Case, D. A. J Chem Phys 2007, 127, 214108.
445. Wan, S. Z.; Stote, R. H.; Karplus, M. J Chem Phys 2004, 121, 9539.
446. Norberg, J.; Nilsson, L. J Phys Chem 1995, 99, 13056.
447. Brooks, C. L., III. J Phys Chem 1986, 90, 6680.
448. Pearlman, D. A. J Phys Chem 1994, 98, 1487.
449. Boresch, S.; Karplus, M. J Phys Chem A 1999, 103, 103.
450. Boresch, S.; Karplus, M. J Phys Chem A 1999, 103, 119.
451. Yang, W.; Bitetti-Putzer, R.; Karplus, M. J Chem Phys 2004, 120, 9450.
452. Boresch, S. Mol Simulation 2002, 28, 13–37.
453. Simonson, T. Mol Phys 1993, 80, 441.
454. Banerjee, A.; Yang, W.; Karplus, M.; Verdine, G. L. Nature 2005, 434, 612.
455. Fleischman, S. H.; Brooks, C. L., III. Proteins 1990, 7, 52.
456. McDonald, J. J.; Brooks, C. L., III. J Am Chem Soc 1991, 113, 2295.
457. Sen, S.; Nilsson, L. Biophys J 1999, 77, 1801.
458. Eriksson, M. A. L.; Nilsson, L. J Mol Biol 1995, 253, 453.
459. Sneddon, S. F.; Tobias, D. J.; Brooks, C. L., III. J Mol Biol 1989, 209, 817.
460. Shobana, S.; Roux, B.; Andersen, O. S. J Phys Chem B 2000, 104, 5179.
461. van Gunsteren, W. F.; Beutler, T. C.; Fraternali, F.; King, P. M.; Mark, A.; Smith, P. In Computer Simulation of Biomolecular Systems Theoretical and experimental approaches; van Gunsteren, W. F.; Weiner, P. K.; Wilkinson, A. J., Eds.; ESCOM: Leiden, 1993; pp. 315–348.
462. Boresch, S.; Leitgeb, M.; Beselman, A.; MacKerell, A. D., Jr. J Am Chem Soc 2005, 127, 4640.
463. Leitgeb, M.; Schroder, C.; Boresch, S. J Chem Phys 2005, 122, 084109.
464. Wolfenden, R.; Andersson, L.; Cullis, P. M.; Southgate, C. C. B. Biochemistry 1981, 20, 849.
465. Roux, B.; Nina, M.; Pomes, R.; Smith, J. C. Biophys J 1996, 71, 670.
466. Boresch, S.; Tettinger, F.; Leitgeb, M.; Karplus, M. J Phys Chem B 2003, 107, 9535.
467. Woo, H.-J.; Roux, B. Proc Natl Acad Sci USA 2005, 102, 6825.
468. Wang, J.; Deng, Y.; Roux, B. Biophys J 2006, 91, 2798.
469. Deng, Y. Q.; Roux, B. J Chem Theory Comput 2006, 2, 1255.
470. Gilson, M. K.; Given, J. A.; Bush, B. L.; McCammon, J. A. Biophys J 1997, 72, 1047.
471. Hermans, J.; Shankar, S. Isr J Chem 1986, 27, 225.
472. Zhang, L.; Hermans, J. Protein Struct Funct Genet 1996, 24, 433.
473. Yang, W.; Bitetti-Putzer, R.; Karplus, M. J Chem Phys 2004, 120, 2618.
474. Li, G. H.; Zhang, X. D.; Cui, Q. J Phys Chem B 2003, 107, 8643.
475. Gao, J. L. J Phys Chem 1992, 96, 537.
476. Bennett, C. H. J Comput Phys 1976, 22, 245.
477. Shirts, M. R.; Pande, V. S. J Chem Phys 2005, 122, 144107.
478. Crooks, G. E. Phys Rev E 1999, 60, 2721.
479. Maragakis, P.; Spichty, M.; Karplus, M. Phys Rev Lett 2006, 96, 100602.
480. Ferrenberg, A. M.; Swendsen, R. H. Phys Rev Lett 1989, 63, 1195.
481. Kumar, S.; Bouzida, D.; Swendsen, R. H.; Kollman, P. A.; Rosenberg, J. M. J Comput Chem 1992, 13, 1011.
482. Bartels, C. Chem Phys Lett 2000, 331, 446.
483. Boczko, E. M.; Brooks, C. L. J Phys Chem 1993, 97, 4509.
484. Brooks, C. L.; Nilsson, L. J Am Chem Soc 1993, 115, 11034.
485. Roux, B. Comput Phys Commun 1995, 91, 275.
486. Norberg, J.; Nilsson, L. J Am Chem Soc 1995, 117, 10832.
487. Huang, N.; Banavali, N. K.; MacKerell, A. D., Jr. Proc Natl Acad Sci USA 2003, 100, 68.
488. Bartels, C.; Schaefer, M.; Karplus, M. J Chem Phys 1999, 111, 8048.
489. Souaille, M.; Roux, B. Comput Phys Commun 2001, 135, 40.
490. Bartels, C.; Widmer, A.; Ehrhardt, C. J Comput Chem 2005, 26, 1294.
491. Brunsteiner, M.; Boresch, S. J Chem Phys 2000, 112, 6953.
492. Min, D. H.; Li, H. Z.; Li, G. H.; Bitetti-Putzer, R.; Yang, W. J Chem Phys 2007, 126, 144109.
493. Weeks, J. D.; Chandler, D. C.; Andersen, H. C. J Chem Phys 1971, 54, 5237.
494. Pomes, R.; Eisenmesser, E.; Post, C. B.; Roux, B. J Chem Phys 1999, 111, 3387.
495. Guo, Z.; Brooks, C. L., III; Kong, X. J Phys Chem B 1998, 102, 2032.
496. Kong, X.; Brooks, C. L., III. J Chem Phys 1996, 105, 2414.
497. Ravimohan, W. L. J. a. C. J Chem Phys 1985, 83, 3050.
498. Liu, Z.; Berne, B. J. J Chem Phys 1993, 99, 6071.
499. Tidor, B. J Phys Chem 1993, 97, 1069.
500. Ji, J.; Cagin, T.; Pettitt, B. M. J Chem Phys 1992, 96, 1333.
501. Bitetti-Putzer, R.; Yang, W.; Karplus, M. Chem Phys Lett 2003, 377, 633.
502. Guo, Z.; Brooks, C. L., III. J Am Chem Soc 1998, 120, 1920.
503. Jarque, C.; Tidor, B. J Phys Chem B 1998, 101, 9362.

504. Valleau, J. P.; Torrie, G. M. In Statistical Mechanics, Part A; Berne, B. J., Ed.; Plenum Press: New York, 1977; pp. 169–194.
505. Shirts, M. R.; Bair, E.; Hooker, G.; Pande, V. S. Phys Rev Lett 2003; 91, 140601.
506. Li, H. Z.; Yang, W. Chem Phys Lett 2007, 440, 155.
507. Li, H. Z.; Fajer, M.; Yang, W. J Chem Phys 2007, 126, 24106.
508. Li, H. Z.; Yang, W. J Chem Phys 2007, 126, 114104.
509. Sugita, Y.; Okamoto, Y. Chem Phys Lett 1999, 314, 141.
510. Feig, M.; Karanicolas, J.; Brooks, C. L., III. J Mol Graph Model 2004, 22, 377.
511. Lahiri, A.; Sarzynska, J.; Nilsson, L.; Kulinski, T. Theor Chem Acc 2007, 117, 267.
512. Karanicolas, J.; Brooks, C. L. Protein Sci 2002, 11, 2351.
513. Karanicolas, J.; Brooks, C. L. Proc Natl Acad Sci USA 2003, 100, 3954.
514. Bernard, D.; Coop, A.; MacKerell, A. D., Jr. J Med Chem 2005, 48, 7773.
515. Gerber, R. B.; Buch, V.; Ratner, M. A. J Chem Phys 1982, 77, 3022.
516. Elber, R.; Karplus, M. J Am Chem Soc 1990, 112, 9161.
517. Miranker, A.; Karplus, M. Proteins-Structure Function and Genetics 1991, 11, 29.
518. Stultz, C. M.; Karplus, M. Proteins-Structure Function and Genetics 1999, 37, 512.
519. Stultz, C. M.; Karplus, M. In Fragment-Based Approaches in Drug Discovery; Jahnke, W.; Erlanson, D. A.; Mannhold, R.; Kubinyi, H.; Folkers, G., Eds.; Wiley-VCH: Weinheim, Germany; 2006; pp. 125–148.
520. Caflisch, A.; Miranker, A.; Karplus, M. J Med Chem 1993, 36, 2142.
521. Eisen, M. B.; Wiley, D. C.; Karplus, M.; Hubbard, R. E. Proteins-Structure Function and Genetics 1994, 19, 199.
522. Joseph-McCarthy, D.; Tsang, S. K.; Filman, D. J.; Hogle, J. M.; Karplus, M. J Am Chem Soc 2001, 123, 12758.
523. Allen, K. N.; Bellamacina, C. R.; Ding, X. C.; Jeffery, C. J.; Mattos, C.; Petsko, G. A.; Ringe, D. J Phys Chem 1996, 100, 2605.
524. Shuker, S. B.; Hajduk, P. J.; Meadows, R. P.; Fesik, S. W. Science 1996, 274, 1531.
525. Sirockin, F.; Sich, C.; Improta, S.; Schaefer, M.; Saudek, V.; Froloff, N.; Karplus, M.; Dejaegere, A. J Am Chem Soc 2002, 124, 11073.
526. Roitberg, A.; Elber, R. J Chem Phys 1991, 95, 9277.
527. Simmerling, C.; Miller, J. L.; Kollman, P. A. J Am Chem Soc 1998, 120, 7149.
528. Straub, J. E.; Karplus, M. J Chem Phys 1991, 94, 6737.
529. Ulitsky, A.; Elber, R. J Chem Phys 1993, 98, 3380.
530. Zheng, W. M.; Zheng, Q. J Chem Phys 1997, 106, 1191.
531. Hixson, C. A.; Wheeler, R. A. Phys Rev E 2001, 6402, 021406.
532. Chandler, D.; Wolynes, P. G. J Chem Phys 1981, 74, 4078.
533. Hinsen, K.; Roux, B. J Chem Phys 1997, 106, 3567.
534. Eisenmesser, E. Z.; Zabell, A. P. R.; Post, C. B. J Biomol NMR 2000, 17, 17.
535. van Schaik, R. C.; Berendsen, H. J. C.; Torda, A. E.; van Gunsteren, W. F. J Mol Biol 1993, 234, 751.
536. Nakai, T.; Kidera, A.; Nakamura, H. J Biomol NMR 1993, 3, 19.
537. Rodinger, T.; Howell, P. L.; Pomes, R. J Chem Phys 2005, 123, 034104.
538. Yu, H.; Ma, L.; Yang, Y.; Cui, Q. Plos Comput Biol 2007, 3, 214.
539. Yu, H. B.; Ma, L.; Yang, Y.; Cui, Q. Plos Comput Biol 2007, 3, 199.
540. Fischer, S.; Verma, C. S.; Hubbard, R. E. J Phys Chem B 1998, 102, 1797.
541. Elber, R.; Karplus, M. Chem Phys Lett 1987, 139, 375.
542. Czerminski, R.; Elber, R. Int J Q Chem 1990, 24, 167.
543. Jonsson, H.; Mills, G.; Jacobsen, K. W., Eds. Nudged Elastic Band Method for Finding Minimum Energy Paths of Transitions; World Scientifc: Singapore, 1998.
544. E, W.; Ren, W. Q.; Vanden-Eijnden, E. J Phys Chem B 2005, 109, 6688.
545. Maragliano, L.; Fischer, A.; Vanden-Eijnden, E.; Ciccotti, G. J Chem Phys 2006, 125, 024106.
546. E, W.; Ren, W. Q.; Vanden-Eijnden, E. Phys Rev B 2002, 66, 052301.
547. Noe, F.; Krachtus, D.; Smith, J. C.; Fischer, S. J Chem Theory Comput 2006, 2, 840.
548. Schlitter, J.; Engels, M.; Kruger, P.; Jacoby, E.; Wollmer, A. Mol Simulation 1993, 10, 291.
549. Grubmüller, H.; Heymann, B.; Tavan, P. Science 1996, 271, 997.
550. Leech, J.; Prins, J.; Harmans, J. IEEE Comp Sci Eng 1996, 3, 38.
551. Izrailev, S.; Stepaniants, S.; Balsera, M.; Oono, Y.; Schulten, K. Biophys J 1997, 72, 1568.
552. Apostolakis, J.; Ferrara, P.; Caflisch, A. J Chem Phys 1999, 110, 2099.
553. van der Vaart, A.; Karplus, M. J Chem Phys 2005, 122, 1149031.
554. Wu, X.; Brooks, B. R. Chem Phys Lett 2003, 381, 512.
555. Chandler, D. J Chem Phys 1978, 68, 2959.
556. Torrie, G. M.; Valleau, J. P. Chem Phys Lett 1974, 28, 578.
557. Berne, B. J. In Mutiple Time Scales; Brackbill, J. U.; Cohen, B. I; Eds.; Academic Press, New York, NY, 1985; pp. 419–436.
558. Berne, B. J.; Borkovec, M.; Straub, J. E. J Phys Chem 1988, 92, 3711.
559. Crouzy, S.; Woolf, T.; Roux, B. Biophys J 1994, 67, 1370.
560. Pomes, R.; Roux, B. Biophys J 2002, 82, 2304.
561. Dellago, C.; Bolhuis, P. G.; Csajka, F. S.; Chandler, D. J Chem Phys 1998, 108, 1964.
562. Bolhuis, P. G.; Chandler, D.; Dellago, C.; Geissler, P. L. Annu Rev Phys Chem 2002, 53, 291.
563. Hagan, M. F.; Dinner, A. R.; Chandler, D.; Chakraborty, A. K. Proc Natl Acad Sci USA 2003, 100, 13922.
564. Noe, F.; Ille, F.; Smith, J. C.; Fischer, S. Proteins 2005, 59, 534.
565. Fischer, S.; Dunbrack, R. L.; Karplus, M. J Am Chem Soc 1994, 116, 11931.
566. Czerminski, R.; Elber, R. J Chem Phys 1990, 92, 5580.
567. Gonzalez, C.; Schlegel, H. B. J Phys Chem 1990, 94, 5523.
568. Wu, X.; Wang, S. J Phys Chem B 1998, 102, 7238.
569. Woodcock, H. L.; Hodoscek, M.; Sherwood, P.; Lee, Y. S.; Schaefer, H. F., III. Brooks, B. R. Theor Chem Acc 2003, 109, 140.
570. Guest, M. F.; Bush, I. J.; Van Dam, H. J. J.; Sherwood, P.; Thomas, J. M. H.; Van Lenthe, J. H.; Havenith, R. W. A.; Kendrick, J. Mol Physics 2005, 103, 719.
571. Woodcock, H. L.; Hodoscek, M.; Brooks, B. R. J Phys Chem A 2007, 111, 5720.
572. Henkelman, G.; Jonsson, H. J Chem Phys 2000, 113, 9978.
573. Henkelman, G.; Uberuaga, B. P.; Jónsson, H. J Chem Phys 2000, 113, 9901.
574. Pan, A. C.; Sezer, D.; Roux, B. J Phys Chem B 2008, 112, 3432.
575. Maragliano, L.; Vanden-Eijnden, E. Chem Phys Lett 2007, 446, 182.
576. Fischer, S.; Karplus, M. Chem Phys Lett 1992, 194, 252.
577. Fischer, S.; Michnick, S.; Karplus, M. Biochemistry 1993, 32, 13830.
578. Dutzler, R.; Schirmer, T.; Karplus, M.; Fischer, S. Structure 2002, 10, 1273.
579. Fischer, S.; Windshugel, B.; Horak, D.; Holmes, K. C.; Smith, J. C. Proc Natl Acad Sci USA 2005, 102, 6873.
580. Park, S.; K.Schulten. J Chem Phys 2004, 120, 5946.
581. Ma, J.; Karplus, M. Proc Natl Acad Sci USA 1997, 94, 11905.
582. Barbacid, M. Annu Rev Biochem 1987, 56, 779.
583. Ford, B.; Hornak, V.; Kleinman, H.; Nassar, N. Structure 2006, 14, 427.

584. Hall, B. E.; Bar-Sagi, D.; Nassar, N. Proc Natl Acad Sci USA 2002, 99, 12138.
585. Ma, J. P.; Sigler, P. B.; Xu, Z. H.; Karplus, M. J Mol Biol 2000, 302, 303.
586. van der Vaart, A.; Ma, J. P.; Karplus, M. Biophys J 2004, 87, 562.
587. Harvey, S. C.; Gabb, H. A. Biopolymers 1993, 33, 1167.
588. Marchi, M.; Ballone, P. J Chem Phys 1999, 110, 3697.
589. Paci, E.; Smith, L. J.; Dobson, C. M.; Karplus, M. J Mol Biol 2001, 306, 329.
590. Daggett, V.; Fersht, A. R. In Mechanisms of Protein Folding; Pain, R. H., Ed.; Oxford university press: Oxford, 2000.
591. Paci, E.; Vendruscolo, M.; Dobson, C. M.; Karplus, M. J Mol Biol 2002, 324, 151.
592. Vendruscolo, M.; Paci, E.; Dobson, C. M.; Karplus, M. J Am Chem Soc 2003, 125, 15686.
593. Astumian, R. D. Science 1997, 276, 917.
594. Balsera, M.; Stepaniants, S.; Izrailev, S.; Oono, Y.; Schulten, K. Biophys J 1997, 73, 1281.
595. Neelov, I. M.; Adolf, D. B.; McLeish, T. C. B.; Paci, E. Biophys J 2006, 91, 3579.
596. Xie, X. S.; Choi, P. J.; Li, G.-W.; Lee, N. K.; Lia, G. Ann Rev Biophys 2008, 37, 417.
597. Lahiri, A.; Nilsson, L.; Laaksonen, A. J Chem Phys 2001, 114, 5993.
598. Wen, E. Z.; Hsieh, M. J.; Kollman, P. A.; Luo, R. J Mol Graph Model 2004, 22, 415.
599. MacFadyen, J.; Andricioaei, I. J Chem Phys 2005, 123, 074107.
600. Kottalam, J.; Case, D. A. J Am Chem Soc 1988, 110, 7690.
601. Bartels, C.; Karplus, M. J Phys Chem B 1998, 102, 865.
602. Sheinerman, F. B.; Brooks, C. L., III. J Mol Biol 1998, 278, 439.
603. Banavali, N. K.; Roux, B. Structure (Camb) 2005, 13), 1715.
604. Banavali, N. K.; Roux, B. J Am Chem Soc 2005, 127, 6866.
605. Rajamani, R.; Gao, J. L. J Am Chem Soc 2003, 125, 12768.
606. Rajamani, R.; Naidoo, K. J.; Gao, J. L. J Comput Chem 2003, 24, 1775.
607. Gao, J. J Am Chem Soc 1991, 113, 7796.
608. Brooks, C. L., III. Acc Chem Res 2002, 35, 447.
609. Mezei, M.; Beveridge, D. L. Ann N Y Acad Sci 1986, 482, 1.
610. Kuczera, K. J Comput Chem 1996, 17, 1726.
611. Wang, Y.; Kuczera, K. J Phys Chem B 1997, 101, 5205.
612. Mahadevan, J.; Lee, K. H.; Kuczera, K. J Phys Chem B 2001, 105, 1863.
613. Depaepe, J. M.; Ryckaert, J. P.; Paci, E.; Ciccotti, G. Mol Phys 1993, 79, 515.
614. Darve, E.; Pohorille, A. J Chem Phys 2001, 115, 9169.
615. Tobias, D. J.; Brooks, C. L. J Chem Phys 1990, 92, 2582.
616. Carter, E. A.; Ciccotti, G.; Hynes, J. T.; Kapral, R. Chem Phys Lett 1989, 156, 472.
617. Gunther, R.; Hofmann, H. J.; Kuczera, K. J Phys Chem B 2001, 105, 5559.
618. Wang, Y.; Kuczera, K. Theor Chem Acc 1999, 101, 274.
619. Bolhuis, P. G. J Phys Condens Mat 2003, 15, S113.
620. Dellago, C.; Bolhuis, P. G.; Chandler, D. J Chem Phys 1998, 108, 9236.
621. Ma, A.; Dinner, A. R. J Phys Chem B 2005, 109, 6769.
622. So, S. S.; Karplus, M. J Med Chem 1996, 39, 1521.
623. So, S. S.; Karplus, M. J Med Chem 1996, 39, 5246.
624. Hu, J.; Ma, A.; Dinner, A. R. Proc Natl Acad Sci USA 2008, 105, 4615.
625. Haliloglu, T.; Bahar, I.; Erman, B. Phys Rev Lett 1997, 79, 3090.
626. Zheng, W. J. Biophys J 2008, 94, 3853.
627. Bahar, I.; Atilgan, A. R.; Erman, B. Fold Des 1997, 2, 173.
628. Ikeguchi, M.; Ueno, J.; Sato, M.; Kidera, A. Phys Rev Lett 2005, 94, 078102.
629. Ming, D.; Kong, Y. F.; Lambert, M. A.; Huang, Z.; Ma, J. P. Proc Natl Acad Sci USA 2002, 99, 8620.
630. Ming, D. M.; Kong, Y. F.; Wakil, S. J.; Brink, J.; Ma, J. P. Proc Natl Acad Sci USA 2002, 99, 7895.
631. Tama, F.; Wriggers, W.; Brooks, C. L. J Mol Biol 2002, 321, 297.
632. Stan, G.; Lorimer, G. H.; Thirumalai, D.; Brooks, B. R. Proc Natl Acad Sci USA 2007, 104, 8803.
633. Maragakis, P.; Karplus, M. J Mol Biol 2005, 352, 807.
634. Heitler, W.; London, F. Zeitschrift f Physik 1927, 44, 455.
635. Warshel, A. Computer Modeling of Chemical Reactions in Enzymes and Solutions; Wiley: New York, 1991.
636. Pu, J. Z.; Karplus, M. Proc Natl Acad Sci USA 2008, 105, 1192.
637. Dinner, A. R.; Blackburn, G. M.; Karplus, M. Nature 2001, 413, 752.
638. Garcia-Viloca, M.; Truhlar, D. G.; Gao, J. Biochemistry 2003, 42, 13558.
639. Gao, J. Acc Chem Res 1996, 29, 298.
640. Mo, Y. R.; Gao, J. J Comput Chem 2000, 21, 1458.
641. Neria, E.; Karplus, M. J Chem Phys 1996, 105, 10812.
642. Neria, E.; Karplus, M. Chem Phys Lett 1997, 267, 23.
643. Nam, K.; Prat-Resina, X.; Garcia-Viloca, M.; Devi-Kesavan, L. S.; Gao, J. J Am Chem Soc 2004, 126, 1369.
644. Pu, J.; Gao, J.; Truhlar, D. G. Chem Rev 2006, 106, 3140.
645. Quaytman, S. L.; Schwartz, S. D. Proc Natl Acad Sci USA 2007, 104, 12253.
646. Basner, J. E.; Schwartz, S. D. J Am Chem Soc 2005, 127, 13822.
647. Ellingson, B. A.; Truhlar, D. G. J Am Chem Soc 2007, 129, 12765.
648. Fernandez-Ramos, A.; Miller, J. A.; Klippenstein, S. J.; Truhlar, D. G. Chem Rev 2006, 106, 4518.
649. Corchado, J. C.; Chuang, Y.-Y.; Fast, P. L.; Hu, W.-P.; Liu, Y.-P.; Lynch, G. C.; Nguyen, K. A.; Jackels, C. F.; Fernandez-Ramos, A.; Ellingson, B. A.; Lynch, B. J.; Zheng, J.; Melissas, V. S.; Villà, J.; Rossi, I.; Coitino, E. L.; Pu, J.; Albu, T. V.; Steckler, R.; Garrett, B. C.; Isaacson, A. D.; Truhlar, D. G. POLYRATE, Version 9.7, Department of Chemistry, University of Minnesota, Minneapolis, 2007.
650. Major, D. T.; Gao, J. J Chem Theory Comput 2007, 3, 949.
651. Major, D. T.; Garcia-Viloca, M.; Gao, J. J Chem Theory Comput 2006, 2, 236.
652. Major, D. T.; Gao, J. J Am Chem Soc 2006, 128, 16345.
653. Kabsch, W.; Sander, C. Biopolymers 1983, 22, 2577.
654. Brünger, A. T.; Brooks, C. L., III; Karplus, M. Proc Natl Acad Sci USA 1985, 82, 8458.
655. Radkiewicz, J. L.; Brooks, C. L., III. J Am Chem Soc 2000, 122, 225.
656. Lahiri, A.; Nilsson, L. Biophys J 2000, 79, 2276.
657. Karpen, M. E.; Tobias, D. J.; Brooks, C. L., III. Biochemistry 1993, 32, 412.
658. Levitt, M. J Mol Biol 1983, 168, 621.
659. Woolf, T. B.; Roux, B. Biophys J 1997, 72, 1930.
660. Allen, T. W.; Andersen, O. S.; Roux, B. J Am Chem Soc 2003, 125, 9868.
661. Yang, D.; Kay, L. E. J Mol Biol 1996, 263, 369.
662. Rudas, T.; Schroder, C.; Boresch, S.; Steinhauser, O. J Chem Phys 2006, 124, 234908.
663. Schroder, C.; Rudas, T.; Boresch, S.; Steinhauser, O. The Journal of Chemical Physics 2006, 124, 234907.
664. Eriksson, M. A. L.; Laaksonen, A. Biopolymers 1992, 32, 1035.
665. Brünger, A. T.; Karplus, M. Proteins 1988, 4, 148.
666. Vieth, M.; Hirst, J. D.; Dominy, B. N.; Daigler, H.; Brooks, C. L. J Comput Chem 1998, 19, 1623.
667. Frank, J. Three-Dimensional Electron Micrsoscopy of Macromolecular Assemblies; Academic Press: London, 1996.

668. Holmes, K. C. Structure 1994, 2, 589.
669. Holmes, K. C.; Angert, I.; Kull, F. J.; Jahn, W.; Schroder, R. R. Nature 2003, 425, 423.
670. Andrusier, N.; Nussinov, R.; Wolfson, H. J. Proteins 2007, 69, 139.
671. Jackson, R. M.; Gabb, H. A.; Sternberg, M. J. E. J Mol Biol 1998, 276, 265.
672. Wang, C.; Bradley, P.; Baker, D. J Mol Biol 2007, 373, 503.
673. Wu, X.; Milne, J. L.; Borgnia, M. J.; Rostapshov, A. V.; Subramaniam, S.; Brooks, B. R. J Struct Biol 2003, 141, 63.
674. Milne, J. L.; Shi, D.; Rosenthal, P. B.; Sunshine, J. S.; Domingo, G. J.; Wu, X.; Brooks, B. R.; Perham, R. N.; Henderson, R.; Subramaniam, S. EMBO J 2002, 21, 5587.
675. Milne, J. L. S.; Wu, X. W.; Borgnia, M. J.; Lengyel, J. S.; Brooks, B. R.; Shi, D.; Perham, R. N.; Subramaniam, S. J Biol Chem 2006, 281, 4364.
676. Lee, D. Y.; Park, S. J.; Jeong, W.; Sung, H. J.; Oho, T.; Wu, X. W.; Rhee, S. G.; Gruschus, J. M. Biochemistry 2006, 45, 15301.
677. Humphrey, W.; Dalke, A.; Schulten, K. J Mol Graph 1996, 14, 33.
678. Backer, A. Comput Sci Eng 2007, 9, 30.
679. Scherer, D.; Dubois, P.; Sherwood, B. Comput Sci Eng 2000, 2, 56.
680. Chabay, R.; Sherwood, B. Am J Phys 2008, 76, 307.
681. Marrink, S. J.; Tieleman, D. P.; Mark, A. E. J Phys Chem B 2000, 104, 12165.
682. Bogusz, S.; Venable, R. M.; Pastor, R. W. J Phys Chem B 2001, 105, 8312.
683. Dixon, A. M.; Venable, R.; Widmalm, G.; Bull, T. E.; Pastor, R. W. Biopolymers 2003, 69, 448.
684. Pastor, R. W.; Venable, R. M.; Feller, S. E. Accounts Chem Res 2002, 35, 438.
685. Skibinsky, A.; Venable, R. M.; Pastor, R. W. Biophys J 2005, 89, 4111.
686. Momany, F. A.; Rone, R. J Comput Chem 1992, 13, 888.
687. Lindahl, E.; Hess, B.; van der Spoel, D. J Mol Mod 2001, 7, 306.
688. Nilsson, L. J Comput Chem 2008, 9999; DOI 10.1002/jcc.21169.
689. Plimpton, S.; Hendrickson, B. J Comput Chem 1996, 17, 326.
690. Shu, J. W.; Wang, B.; Chen, M.; Wang, J. Z.; Zheng, W. M. Comput Phys Commun 2003, 154, 121.
691. Shaw, D. E. J Comp Chem 2005, 26(13), 1318.
692. Bowers, K. J.; Dror, R. O.; Shaw, D. E. J Chem Phys 2006, 124, 184109.
693. Kale, L.; Skeel, R.; Bhandarkar, M.; Brunner, R.; Gursoy, A.; Krawetz, N.; Phillips, J.; Shinozaki, A.; Varadarajan, K.; Schulten, K. J Comput Phys 1999, 151, 283.
694. van de Geijn, R. A. Technical Report CS-91–129; University of Tennessee, 1991.
695. Brooks, B. R.; Hodoscek, M. Chem Design Automation News 1992, 7, 16.
696. Hwang, Y.; Das, R.; Saltz, J.; Hodoscek, M.; Brooks, B. R. IEEE Computational Science and Engineering 1995, 2, 18.
697. Clark, T. W.; Hanxleden, R. V.; McCammon, J. A.; Scott, L. R. Scalable High-performance Computing Conference, 1994; pp. 95–102.
698. Nelson, M. T.; Humphrey, W.; Gursoy, A.; Dalke, A.; Kale, L. V.; Skeel, R. D.; Schulten, K. Int J Supercomputer Appl High Perform Comput 1996, 10, 251.
699. Eichinger, M.; Grubmuller, H.; Heller, H.; Tavan, P. J Comput Chem 1997, 18, 1729.
700. Hess, B.; Kutzner, C.; vanderSpoel, D.; Lindahl, E. J Chem Theory Comput 2008, 4, 435.
701. Mertz, J. E.; Tobias, D. J.; Brooks, C. L.; Singh, U. C. J Comput Chem 1991, 12, 1270.
702. Narumi, T.; Susukita, R.; Ebisuzaki, T.; McNiven, G.; Elmegreen, B. Molecular Simulation 1999, 21, 401.
703. Borstnik, U.; Hodoscek, M.; Janezic, D. J Chem Inf Comput Sci 2004, 44, 359.
704. Becker, O.; Karplus, M. A Guide to Biomolecular Simulations; Springer, 2006.
705. Gelin, B. R. Ph.D. Thesis, Chemistry; Harvard University: Cambridge, 1976.
706. Szabo, A.; Karplus, M. J Mol Biol 1972, 72, 163.
707. Gelin, B. R.; Karplus, M. Proc Natl Acad Sci USA 1977, 74, 801.
708. Case, D. A.; Karplus, M. J Mol Biol 1979, 132, 343.
709. Levitt, M.; Lifson, S. J Mol Biol 1969, 46 269.
710. Scheraga, H. A. Adv Phys Org Chem 1968, 6, 103.
711. Brünger, A. T.; Kuriyan, J.; Karplus, M. Science 1987, 235, 458.
712. Polanyi, J.; Zewail, A. H. Acc Chem Res 1995, 28, 119.
713. Lim, M.; Jackson, T. A.; Anfinrud, P. A. Science 1995, 269, 962.
714. de Groot, B. L.; Grubmuller, H. Science 2001, 294, 2353.
715. Tajkhorshid, E.; Nollert, P.; Jensen, M. O.; Miercke, L. J. W.; O'Connell, J.; Stroud, R. M.; Schulten, K. Science 2002, 296, 525.
716. de Vries, A. H.; Mark, A. E.; Marrink, S. J. J Am Chem Soc 2004, 126, 4488.
717. Bockmann, R. A.; Grubmuller, H. Nat Struct Biol 2002, 9, 198.
718. Cowan, W. M.; Sudhof, T. C.; Stevens, C. F., Eds. Synapses; Johns Hopkins University Press: Baltimore, Maryland, 2000.
719. Lippincott-Schwartz, J. Nature 2002, 416, 31.
720. Tama, F.; Brooks, C. L., 3rd. J Mol Biol 2002, 318, 733.
721. Tama, F.; Valle, M.; Frank, J.; Brooks, C. L., III. Proc Natl Acad Sci USA 2003, 100, 93193.
722. Young, M. A.; Gonfloni, S.; Superti-Furga, G.; Roux, B.; Kuriyan, J. Cell 2001, 105, 115.
723. Noskov, S. Y.; Berneche, S.; Roux, B. Nature 2004, 431, 830.
724. Zhang, P.; Yu, M. Y. W.; Venable, R.; Alter, H. J.; Shih, J. W. K. Proc Natl Acad Sci USA 2006, 103, 9214.