IEEE COMPUTATIONAL SCIENCE & ENGINEERING 1070-9924/96/$5.00 © 1996 IEEE
Vol. 3, No. 4: WINTER 1996, pp. 38-45

Corresponding author: Jonathan Leech, Computer Science Department, Campus Box #3175, Univ. of North Carolina, Chapel Hill, NC 27599-3175.

SMD: Visual Steering of Molecular Dynamics for Protein Design

Jonathan Leech *University of North Carolina at Chapel Hill*

Jan F. Prins *University of North Carolina at Chapel Hill*

Jan Hermans *University of North Carolina at Chapel Hill*

SMD, a system for interactively steering molecular dynamics calculations of protein molecules, includes computation, visualization, and communication components. Biochemists can "tug" molecules into different shapes by specifying external forces in the graphical interface, which are added to internal forces representing atomic bonds and nonbonded interactions.

Computers have become an essential component of molecular modeling work, especially for complex biomolecules. They are commonly used for such tasks as modeling the docking between ligand and receptor molecules, structure refinement from X-ray crystallography and NMR data, and sequence analysis. Newer tools are intended to aid in designing new proteins as well as analyzing those found in nature. Designing proteins not found in nature is a rapidly developing field. In part, such designs serve to test the validity and applicability of computational molecular models. As experience with *de novo* design grows, proteins may be tailored for specific effects, such as new types of drugs.

We have built SMD, a system for interactively steering molecular dynamics calculations by adding user-specified external forces, which we call "tugs," into the computation on the fly. The purpose of introducing external forces is to help the molecular system overcome energy barriers between states. One application of this is to steer the system to a proposed new geometric conformation, which can be further studied. In other cases, the forces model physical experiments in which molecular complexes are pulled apart via atomic-force microscopy. The experiments in turn provide some validation of the computed model. The difficulty of changing the conformations in the computational model gives insight into physically realizible pathways.

SMD provides a new tool for biochemists to use in exploring the structure of proposed designs, as well as in more general applications such as exploring the molecular dynamics model itself. Its primary use is in modeling single large biomolecules in a bath of water acting as the solvent.

Previous energy minimization work

SMD was inspired by the Sculpt[1] project. Sculpt is a molecular modeling package with two main components: an *energy minimization* kernel (see glossary), and an interactive graphical interface. The user may add energy terms to the system as minimization progresses, and the system maintains a valid model by preventing close atomic contacts. Sculpt has been used as a *de novo* protein design tool by biochemistry researchers.[2]

Sculpt was the first comprehensive interface for steering molecular mechanics calculations that we know of. Earlier work, such as interactive molecular docking calculations,[3] was restricted to static molecules.

SMD adds molecular dynamics

SMD extends Sculpt by using molecular dynamics[4] instead of energy minimization, in the hope of achieving computed models that more faithfully approximate actual proteins. In our molecular dynamics code, the electrostatic and atomic bond interactions of a system of atoms are approximated by Newtonian forces which are integrated to determine the positions and velocities of atoms as the simulation progresses. This offers two new capabilities as compared to the Sculpt energy minimization model:

- A more accurate molecular *force field* (see glossary), including an explicit model of the solvent (water).
- A time-based model, which explores the conformation space—the range of possible shapes—available to a molecule and in which changes reflect a physically plausible evolution of the system along a path of low free energy. This model allows us to represent important information in the system, including temperature, pressure, and entropy.

Motivating problems for SMD

The computational side of our work has been motivated by several research problems of interest to our "client" in the SMD development effort, biochemist Jane Richardson of Duke University. The primary problem we have worked on is a specific protein system, *SScorin*, that Richardson is studying. This protein, shown in Figure 1, is a 56-residue *de novo* design originally created using Sculpt; with solvent molecules added, it comprises nearly 4,000 atoms.

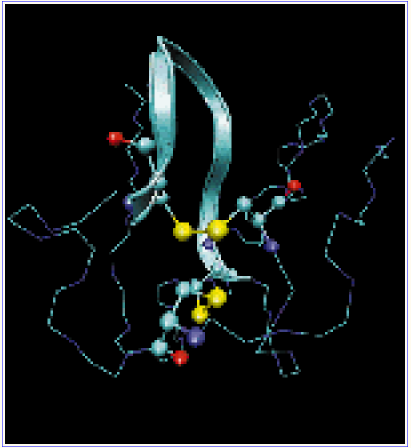


Figure 1. SScorin, a man-made protein showing cysteine residues forming disulfide bridge, loop region as ribbon, and remainder of protein backbone as bonds. Solvent present in the simulation is not shown.

Laboratory synthesis and NMR studies of SScorin showed discrepancies between the computer model determined by Sculpt and the actual synthesized protein. The two disulfide bonds which help stabilize the protein, and after which it is named, turned out to form with a connectivity different than that modeled. As presently implemented, our molecular dynamics code does not model chemical bond formation, which is usually done with much slower quantum mechanics-based models, so direct study of this part of the protein folding process is not feasible. However, it is possible to compare the stability of the two different bonding configurations.

Our hope is that the more capable molecular dynamics model in SMD, in principle an improvement over the energy minimization scheme in Sculpt, will provide insight into the causes of this discrepancy and assist in a redesign of the molecule. Preliminary investigation by Richardson suggests that length of the backbone loop between the cysteine residues forming the disulfide bond may be the dominant feature in avoiding alternative connectivities, and SMD will be used to assess whether, and why, it may be unfavorable to close the cysteine-delimited loop.

Another problem of interest to our biochemistry colleagues, which makes use of different aspects of molecular dynamics, is in an earlier stage of development: studying the parallel hydrogen bonds in a β sheet structure. The Sculpt model did not produce bond spacing corresponding to that observed in real proteins, and we hope that the solvation model will alleviate this.

The SMD system

SMD consists of two software components (Figure 2). The *computational* component is a version of the Sigma molecular dynamics package[5] (often written SIgMA) and is responsible for computing the dynamics of the system under study, including its response to user-defined external forces. The *display* component is VMD, a molecular graphics code written as part of the MDScope[6] project at the University of Illinois. (The MDScope developers have also created an interface to their NAMD dynamics code using VMD.)

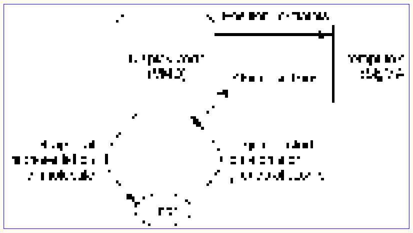


Figure 2. The SMD interactive molecular dynamics system.

We have modified VMD and Sigma to communicate with each other using a customized, lightweight protocol. Sigma sends atomic positions resulting from each molecular dynamics time step to VMD for display. When the user specifies restraints on parts of the displayed model, VMD sends them to Sigma, where they are converted into potential-well restraints added to the force field.

Graphics and computation requirements

Interactions among atoms in real molecules occur extremely quickly, but molecular dynamics computations are inherently slow. A Verlet integration time step on the order of a femtosecond (10^{-15} s) of real time is required to properly model the highest-frequency oscillations of a molecule, while interesting protein dynamics such as side-chain rotations happen on time scales thousands to millions of times longer.

For SMD to be a useful interactive tool, modifications must be performed in reasonable amounts of wall-clock time—seconds to minutes at worst. Because Sigma uses a distance cutoff when computing long-range electrostatic forces, and because all other force calculations are to a bounded number of near neighbors, performance is observed to decrease roughly linearly with system size (total number of atoms). A consequence of this is that by expressing performance in terms of atomic updates per second (the product of system size and time steps per second), a figure of merit can be computed for any given computation platform (see Table 1). This can then be used to determine expected performance given the size of the protein molecule (in atoms) to be simulated.

Table 1. SMD molecular dynamics performance for some workstation CPUs.

	Performance
	(atomic
Processor and	**updates**
clock speed	**per second)**

SGI R4400-250	6,000
HP PA7100-125	11,000
HP PA7200-100	12,000
SGI R8000-90	15,000

Hardware

We usually run both the computing platform and display codes on a multiprocessor SGI R8000 Power Onyx workstation, using a RealityEngine 2 graphics accelerator. Graphics performance is usually adequate for smooth display updates of the 4,000-atom systems we use. In most cases the solvent (water), which makes up perhaps three-fourths of the atoms in the system, is not displayed because it obscures the more interesting behavior of protein atoms; however, it remains part of the dynamics calculation. Our biochemist colleagues generally prefer simple vector representations of atomic bonds, instead of the more complex graphics that appear in some of the figures accompanying this article. Space-filling, ball-stick, or solvent-accessible-surface representations may seem more attractive; however, these presentations can slow down graphical response.

When running computation and display on separate platforms, as we used to do (and may again, depending on available machines), network bandwidth is a potential concern. The upper bound of computing performance on our system is about 15,000 atomic updates per second, which corresponds to about 180 Kbytes/s of coordinate data. This is about 1/7 the peak bandwidth of Ethernet, which cannot easily be obtained using the TCP-based SMD protocol. The problem is eased when we are not displaying solvent atoms, the usual case. Initially, Sigma sends coordinates of all atoms to VMD, but if we don't want to display solvent, we can tell Sigma not to *update* the atomic coordinates of the water molecules. That is, Sigma still *computes* the positions, but it does not send updates of the positions to VMD. This process is not yet automatic, but a manually selectable option instructs Sigma to update only protein atoms. The resulting fivefold reduction in data volume makes Ethernet easily adequate to the task. Similarly, if other atoms are not displayed or not moving, as described in the next section, their coordinates need not be updated.

A more serious problem is latency. When using contention-based networks like Ethernet, unpredictable delays can arise. The result is that Sigma response to user input becomes unpredictable and bursty. The best solution would be a network which can guarantee fixed amounts of bandwidth to individual applications, such as ATM.

We initially expected to run dynamics computations on a remotely located massively parallel computing server, and moved to the SGI Onyx in part because of these networking issues. With compute and display components running on closely coupled processors of a shared-memory multiprocessor such as the Power Onyx, latency and bandwidth problems are eliminated.

"Molten zones": Trading accuracy for performance

As mentioned, Sigma can compute about 15,000 atomic updates per second on the fastest computing platform available to us, an SGI Onyx workstation. The SScorin protein design used as an application problem consists of 56 protein residues (see glossary) and roughly 4,000 atoms including solvent. This limits the computation update rate to about four time steps (8 femtoseconds of real molecular dynamics, with our 2-fs time steps) per second of wall-clock time. Tests with smaller models demonstrate that an update rate of 15-20 steps per second is needed before it is possible to steer the computation in a truly interactive fashion.

We are using two options to increase performance: using a faster computing engine, and reducing the system size by a factor of approximately 5. Parallel versions of Sigma which exhibit good scaling to moderate numbers of processors are being developed by other members of our research group, and we are preparing to run with multiple processors in the future. In the meantime, we can reduce the effective system size, albeit with some loss of fidelity in the simulation. This is done by restricting dynamics to a particular region of interest in the protein and to the solvent molecules that are within a specified distance —usually 8 angstroms, the distance cutoff for electrostatics—of this region. The remaining atoms in the system contribute to forces on the moving atoms, but do not themselves move. The moving region is called a *molten zone*.

While this reduction in system size allows studying a system of real scientific interest to our client, it is clear from our experience that much faster computing platforms will be needed in the future.

SMD's user interface

The SMD user interface is similar to that of Sculpt. As each computation step completes, the system state is sent to VMD, which updates a graphical representation of the molecule. VMD can display many possible representations of a molecule, and can manipulate the display to present different views.

We have added features to VMD to control and steer molecular dynamics. High-level commands to control the simulation, such as starting and stopping Sigma or adjusting parameters, are available through the menu system. By *steering*, we mean SMD's additional capability that allows a user to *tug* individual atoms or groups of atoms towards a desired target position at any time. This is done by entering a *tug mode* through menu or keyboard commands in the VMD interface. This changes the function of the mouse. In tug mode, an atom or group of atoms to tug is selected by pointing at the desired atoms with the mouse (Figure 3a) and clicking on them.

Once the atoms to be tugged are selected, the mouse is moved to specify a target position and released to fix the restraint (Figure 3b). As simulation progresses, the tugged atoms gradually move towards the target position (Figure 3c).

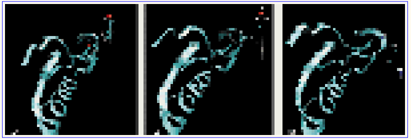


Figure 3. (a) Specifying a "tug": selected atom is highlighted as the red sphere at upper right. (b) Moving the tug: target position follows pointer. (c) System response after 300 fs of simulation.

We implemented tugging with *potential-well restraints*. These are external forces added to the molecular dynamics computation that behave like a spring with one end fixed at the target position, and the other attached to the atom being restrained, exerting an attractive force between the atom and the restraint. A scaling factor for the restraint strength may be specified when the tug is created. This parameter controls the speed at which the tug operates.

Tugs may be deleted at any time, returning the computation to unrestrained molecular dynamics. Because protein molecules are highly geometrically constrained, completing a manipulation usually requires moving atoms in several stages, creating new tugs at each stage.

Using a mouse as the pointing device limits us to two degrees of translational freedom, so the tug operates in a single plane. VMD also supports 3D pointing devices, which allow greater flexibility in moving restraints. However, using these devices requires a more elaborate hardware setup including either a stereo or immersive display system and the 3D pointer itself. For our purposes, the 2D mouse interface has proved adequate.

We should note that trying to achieve structurally significant changes by tugging on individual atoms is sometimes undesirable because of the distortion this introduces to the system. SMD also supports operations on *rigid bodies*—groups of atoms which are to be manipulated as a unit, such as an α helix which is to be realigned with respect to the protein without distorting the helix itself. The group of atoms to be affected is defined through VMD, and the tug specified by selecting any atom in the group. Rather than restraining only the atom selected, however, all atoms in the group are tugged by separate, parallel restraints, producing a uniform translational force on the group.

Molecular modeling details: Manipulation effects

In our system, interaction with the molecular dynamics simulation is done by adding external restraint forces to the system, in addition to the physically realistic covalent bonding and electrostatic forces. External forces allow the system to rapidly cross potential energy barriers between desired conformational states, but they affect the validity of the simulation. As the magnitude of external forces increases, manipulations may be accomplished more rapidly, but the predictive value of the simulation is decreased. We have begun quantifying these effects in order to determine the conditions under which manipulation may be done while retaining predictive value in the simulation.

How the system reacts to restraints

The most blatant effect of adding restraints is an increase in the overall kinetic energy of the system corresponding to the strength and duration of the restraint, which can also be quantified as an increase in temperature.

More subtle effects are observed on the structure and ordering properties of the model when manipulations operate on a time scale faster than the model can react—for example, when rapidly moving between two conformational states. Such effects include disruption of solvent structure near the protein surface.

Energy dissipation and temperature restraints

Molecular dynamics simulations are often run with a *temperature restraint*[7] which models an external heat bath and keeps the system near a desired temperature, by a rescaling of atomic velocities at each time step to account for a portion of the difference between the actual and desired system temperatures. The temperature restraint has a specified relaxation time which normally keeps the system close to the desired temperature.

When an atom is subject to a strong tug, additional energy is pumped in much faster than the temperature restraint can remove it. This appears as a significant increase in system temperature. As an extreme example, consider a simple manipulation of the alanine dipeptide molecule,8 flipping the side chain between two local minima (Figure 4). The dipeptide is often used for examples because it is among the simplest protein molecules with multiple stable conformations.

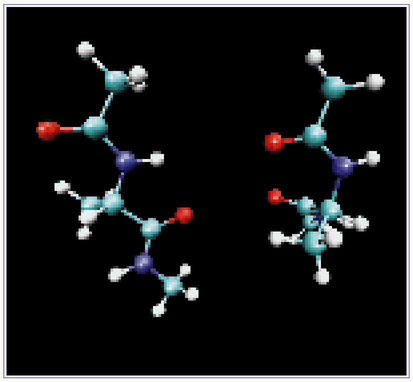


Figure 4. Before *(left)* and after *(right)* diagrams of dipeptide structure. Note position of oxygen atoms (red).

The temperature of the system before, during, and after the manipulation in SMD is charted in Figure 5. The temperature restraint operates at 300 Kelvins with a relaxation time of 100 fs (50 time steps), which keeps system temperature in a narrow range until the onset of tugging at time T_0. The oxygen atom being tugged rapidly moves through the energy barrier, driving system temperature to nearly 900 K; as the restrained atom approaches the final position, the restraint force diminishes as does overall temperature. The system rapidly returns to the 300 K temperature range after manipulation ceases at time T_1.

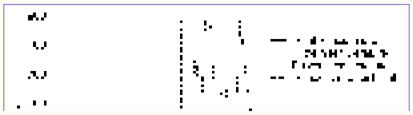
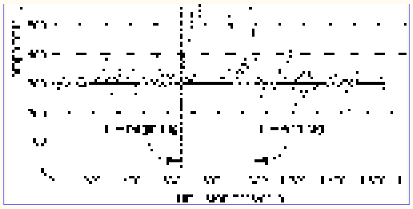


Figure 5. Graph of dipeptide system temperature during



manipulation.

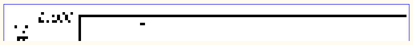
This example demonstrates how manipulation can dominate molecular dynamics in terms of effects on the system. Because protein models of interest to our clients are much larger, tugging on a single atom will have less effect on the overall system temperature, but these effects are still important and must be accounted for.

When a restraint acts on a specific atom, energy is added directly only to that atom. This energy then dissipates into the remainder of the system through the usual bonded and nonbonded interactions. The result is that a hot region is formed near the restrained atom, which is at a much higher temperature than the system as a whole. The greatest disruptive effects of manipulation can be expected to arise in this region. To alleviate them, the hot region may be treated specially by the temperature restraint. It is not unusual to restrain protein and solvent temperatures as separate ensembles, so a separate temperature restraint for the hot region is a logical extension of this practice.

Stronger tugs go faster, but...

Another way of reducing disruptive effects is to decrease the restraint forces. By reducing the strength of restraints, we reduce the energetic and disordering effects on the system. But we also increase the time to complete a desired manipulation, and in some cases may affect its feasibility. We would like to understand the tradeoffs between how fast a specified modeling task can be accomplished and how large an effect is had upon the system in the process. To begin with, we defined a repeatable manipulation task for the SScorin protein design: rotating the lysine side chain at residue 15 by approximately 90 degrees from its initial position. This was accomplished by tugging on the terminal nitrogen atom with the restraint position set at the desired endpoint. The first stages of this manipulation task were previously shown in Figure 3, demonstrating the user interface.

The task was repeated using different restraint strengths ranging from 2 to 10 kcal/(mol·Å^2). Because the potential-well depth diminishes as the target position is approached, atoms will never reach that position and stop, but only approach it. To define the time to complete the task, we chose the time at which the tugged atom approached within a specified distance *D* of the target position. Because small changes in initial conditions may lead to the simulation following different pathways, several trials were conducted at each data point, varying the restraint strength by less than 0.5 percent. The different runtimes resulting from these variations are shown by the error bars in Figure 6. We observe that the time to completion increases much faster than linearly when tug strength is decreased beyond a certain point, and that time to completion has a minimum value set by the model constraints. This indicates that tugs need to act in a limited range of strengths to be effective.



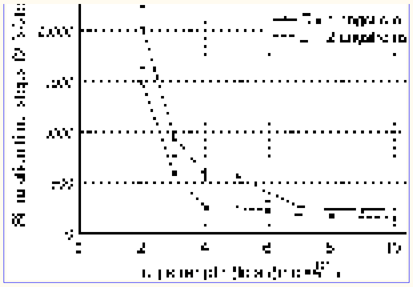


Figure 6. Time to complete a manipulation versus restraint strength.

Are tugs physically plausible?

Measurements such as those made for the manipulation task described above give a measure of how fast a system can react. But they do not provide much guidance as to how fast a system should be allowed to react, or to whether the change in the system is physically plausible. Given strong enough restraints, any desired conformation could be produced. However, our goal is not to specify a precise final state, but to guide the simulation down a faster pathway to a state it might eventually reach in any case.

Recent work by Grubmüller and collaborators[9] gives an indication of this problem. They simulated the force required to rupture a ligand-receptor complex, using a position restraint to literally pull the two molecules apart. The restraint target position was initially placed close to the complex, and gradually moved away at a constant velocity. For sufficiently slow velocities, the calculated rupture force varied systematically with the restraint velocity, and extrapolation to much slower speeds showed good agreement with a corresponding physical experiment[10] using an atomic force microscope. Once restraint velocity was increased beyond a threshold, however, the measured rupture force varied unpredictably, so the simulation had no predictive value.

Hybrid modeling: Combining interactivity and batch jobs

Given the speed of available computing resources, we believe that it is probably not practical to execute most SMD modeling tasks in a truly interactive fashion while retaining predictive value in the simulation. Nonetheless, the benefits of *specifying* these tasks in an interactive environment may be substantial. This suggests a hybrid modeling paradigm in which the client can define the task using high restraint strengths to create an approximate solution. Once defined, the task can then be redone over a longer time scale to collect good statistics. We are starting to implement techniques supporting this approach, but have not completed this work yet.

The information defining a modeling task consists of the atoms that are directly affected, the positions and strengths of the restraints affecting them, and the simulation times at which those restraints are created, changed, and removed. If this information is recorded, we can try to "play it back" more slowly to accomplish the same task with less effect on the system dynamics. Starting from the first point at which a restraint was specified, the idea is to reduce restraint strengths but allow more time for the weaker restraints to take effect.

There are several strategies for accomplishing this. The simplest is a proportional decrease in tug strength and increase in time such that the total external energy put into the system remains unchanged. Our expectation is that this frequently may not work. As seen in Figure 6, the relationship between tug strength and system reaction time is not linear over a wide range of strengths. Simply increasing the time scale might cause the system to end up in a quite different place than that desired due to the divergences accumulating over the course of multiple manipulations.

More sophisticated strategies will try to identify the intent of manipulations in terms of the system state, and replay restraints until that state is reached. The state most easily measured is the position of restrained atoms, which may be recorded in addition to the restraint positions and strengths. As in the SScorin manipulation, the restraint can be maintained not for a specific time related to the initial restraint duration, but rather until the tugged atom approaches the recorded end position.

CONCLUSION

The SMD system adds an interactive element to a computationally intensive modeling technique that has typically been used in batch mode. Because the time scale for molecular dynamics computations is so long, we have begun developing ways to use interactivity to *guide* more physically faithful batch computations; similar ideas may be applicable in other areas such as CFD.

Work on our clients' driving problems has progressed more slowly than the user interface and visualization elements. The next stage of SMD work will apply the techniques described for controlling restraint effects to these problems. We expect that both the control methods and the user interface will evolve in response to client needs.

Future directions for steered molecular modeling work should include a firmer theoretical foundation for studying effects of interaction, a more sophisticated user interface to the wide variety of restraints and parameters available in molecular dynamics computations, and coping with the latency and bandwidth problems that may be encountered when using large remote computing servers.◆

Acknowledgments

This work has been supported by the National Institutes of Health's National Center for Research Resources (grant RR08102 to the UNC/Duke/NYU Computational Structural Biology Resource).

References

1. M.C. Surles, *Techniques for Interactive Manipulation of Graphical Protein Models,* doctoral dissertation, Univ. of North Carolina, Chapel Hill, N.C., 1992. Also available as Tech. Report TR92-016, Dept. Computer Science, UNC.

2. M.C. Surles et al., "Sculpting Proteins Interactively: Continual Energy Minimization Embedded in a Graphical Modeling System," *Protein Science,* Vol. 3, 1994, pp. 198-210.

3. F.P. Brooks Jr. et al., "Project GROPE —Haptic Displays for Scientific Visualization," *Computer Graphics,* Vol. 24, 1990, pp. 177-185.

4. C.L. Brooks III, M. Karplus, and B.M. Pettitt, *Proteins: A Theoretical Perspective of Dynamics, Structure, and Thermodynamics,* Advances in Chemical Physics Vol. 71, John Wiley & Sons, New York, 1988, Ch. 1.

5. J. Hermans, "Sigma Documentation," Univ. of North Carolina, URL: http://femto.med.unc.edu/SIGMA/, 1994.

6. M. Nelson et al., "MDScope: A Visual Computing Environment for Structural Biology," *Computer Physics Comm.,* Vol. 91, 1995, pp. 111-134.

7. H.J.C. Berendsen et al., "Molecular Dynamics with Coupling to an External Bath," *J. Chemical Physics,* Vol. 81, 1984, pp. 3684-3690.

8. A.G. Anderson and J. Hermans, "Microfolding: Conformational Probability Map for the Alanine Dipeptide in Water from Molecular Dynamics Simulation," *Proteins,* Vol. 3, 1988, pp. 262-265.

9. H. Grubmüller, B. Heymann, and P. Tavan, "Ligand Binding: Molecular Mechanics Calculation of the Streptavidin-Biotin Rupture Force," *Science,* Vol. 271, 1996, pp. 997-999.

10. V.T. Moy, E. Florin, and H. Gaub, "Intermolecular Forces and Energies Between Ligands and Receptors," *Science,* Vol. 266, 1994, pp. 257-259.

__Jonathan Leech__ is a doctoral candidate in computer science at the University of North Carolina at Chapel Hill. His interests include interactive physically-based modeling, molecular graphics, scalable high-performance graphics rendering systems, and APIs which support programmable shading and rasterization models for high-quality, high-speed rendering. He received a BS in engineering and applied science from Caltech in 1985 and an MS in computer science from Caltech in 1989. E-mail, leech@cs.unc.edu.

__Jan F. Prins__ is associate professor of computer science at the University of North Carolina, Chapel Hill. He obtained a BS (1978) in mathematics from Syracuse University, and an MS (1983) and PhD (1987) in computer science from Cornell. He is currently a visiting professor at ETH Zurich. His research interests center on parallel computing, including algorithm design, computer architecture, and programming languages. He participates in the research component of the UNC/Duke/NYU Parallel Computing Resource for Structural Biology sponsored by NIH. Prins is a member of the IEEE and ACM. E-mail, prins@cs.unc.edu.

__Jan Hermans__ is professor of biochemistry and biophysics in the School of Medicine at the University of North Carolina at Chapel Hill. He obtained a BS in chemistry from the

University of Groningen and a PhD (1958) in physical chemistry from the University of Leiden. His research interests center on application of computer modeling to molecular structural biology. He directs the UNC/Duke/NYU Parallel Computing Resource for Structural Biology. Hermans is a member of the American Chemical Society, the Biophysical Society, the American Crystallographic Association, and AAAS. E-mail, hermans@med.unc.edu.

Glossary

equation **helix:** a major structural element found in proteins, in which 3.6 residues per turn are wound around each turn of a helix with hydrogen bonds between adjacent turns of the helix.

equation **sheet:** another major structural element in proteins, in which strands of protein backbone are aligned adjacent to each other so that hydrogen bonds between the strands can form.

de novo **design:** a protein which does not exist in nature, but can be created with molecular modeling and synthesized in the laboratory.

energy minimization: systematically changing molecular structure to continually lower the total energy of a system, finding local energy minima. Total energy is the sum of intra- and intermolecular energies, which contain terms for local geometric deformation and long-range attraction and repulsion. These terms approximate a more expensive quantum-mechanical model.

force field: a parameterized model that approximates atomic interactions in terms of classical mechanics and electrostatics. Terms in the model correspond to atomic bonds, bond angles and dihedral angles, repulsive and attractive forces, and electrostatic forces.

residue: biochemical term for one amino acid unit in a protein.

restraint: a term added to the molecular dynamics force field which constrains a term in the computation towards a specified value. Common types of restraints include atom position, bond angle, and protein temperature.

[END OF ARTICLE]

CS home | All titles | This title | Expand this view | Collapse this view | Previous hit | Next hit | Clear search
Show frames

□ Expand Search Search

Table of contents for IEEE Computational Science & Engineering 1996

About the Articles

Articles

Articles

Interview

Book News & Reviews

Conferences & Workshops

Product News

Interfaces

Summer 1996 (Vol. 3, No. 2)

From the New Editor-in-Chief

CSE at Work

Computing Prescriptions

Theme Articles

Theme Articles

Features

Features

Conferences & Workshops

Book News & Reviews

Product News

News Analysis

Interfaces

Fall 1996 (Vol. 3, No. 3)

From the New Editor-in-Chief

Site Report

Computing Prescriptions

CSE Education

About the Articles

Features

Features

Theme Articles

News Analysis

CSE at Work

Conferences & Workshops

Technology News

Interfaces

▼ **Winter 1996 (Vol. 3, No. 4)**

From the Editor-in-Chief

Project Report

CSE at Work

Theme Articles

Theme Articles

Feature Articles

Feature Articles

1996 Annual Index

Conference & Workshops

Book News & Reviews

In Memoriam

CSE Education

Technology News

Letters

Interfaces

1. M.C. Surles, *Techniques for Interactive Manipulation of Graphical Protein Models,* doctoral dissertation, Univ. of North Carolina, Chapel Hill, N.C., 1992. Also available as Tech. Report TR92-016, Dept. Computer Science, UNC.

Glossary

equation **helix:** a major structural element found in proteins, in which 3.6 residues per turn are wound around each turn of a helix with hydrogen bonds between adjacent turns of the helix.

equation **sheet:** another major structural element in proteins, in which strands of protein backbone are aligned adjacent to each other so that hydrogen bonds between the strands can form.

de novo **design:** a protein which does not exist in nature, but can be created with molecular modeling and synthesized in the laboratory.

energy minimization: systematically changing molecular structure to continually lower the total energy of a system, finding local energy minima. Total energy is the sum of intra- and intermolecular energies, which contain terms for local geometric deformation and long-range attraction and repulsion. These terms approximate a more expensive quantum-mechanical model.

force field: a parameterized model that approximates atomic interactions in terms of classical mechanics and electrostatics. Terms in the model correspond to atomic bonds, bond angles and dihedral angles, repulsive and attractive forces, and electrostatic forces.

residue: biochemical term for one amino acid unit in a protein.

restraint: a term added to the molecular dynamics force field which constrains a term in the computation towards a specified value. Common types of restraints include atom position, bond angle, and protein temperature.[END OF ARTICLE]

2. M.C. Surles et al., "Sculpting Proteins Interactively: Continual Energy Minimization Embedded in a Graphical Modeling System," *Protein Science*, Vol. 3, 1994, pp. 198-210.

3. F.P. Brooks Jr. et al., "Project GROPE —Haptic Displays for Scientific Visualization," *Computer Graphics*, Vol. 24, 1990, pp. 177-185.

4. C.L. Brooks III, M. Karplus, and B.M. Pettitt, *Proteins: A Theoretical Perspective of Dynamics, Structure, and Thermodynamics,* Advances in Chemical Physics Vol. 71, John Wiley & Sons, New York, 1988, Ch. 1.

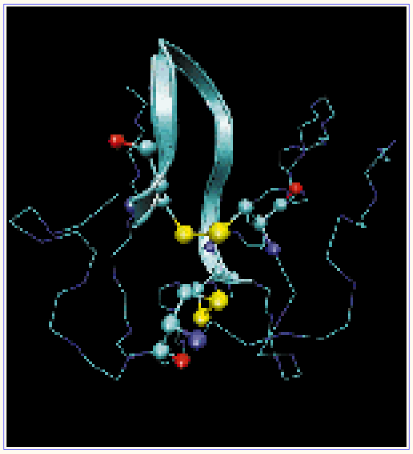
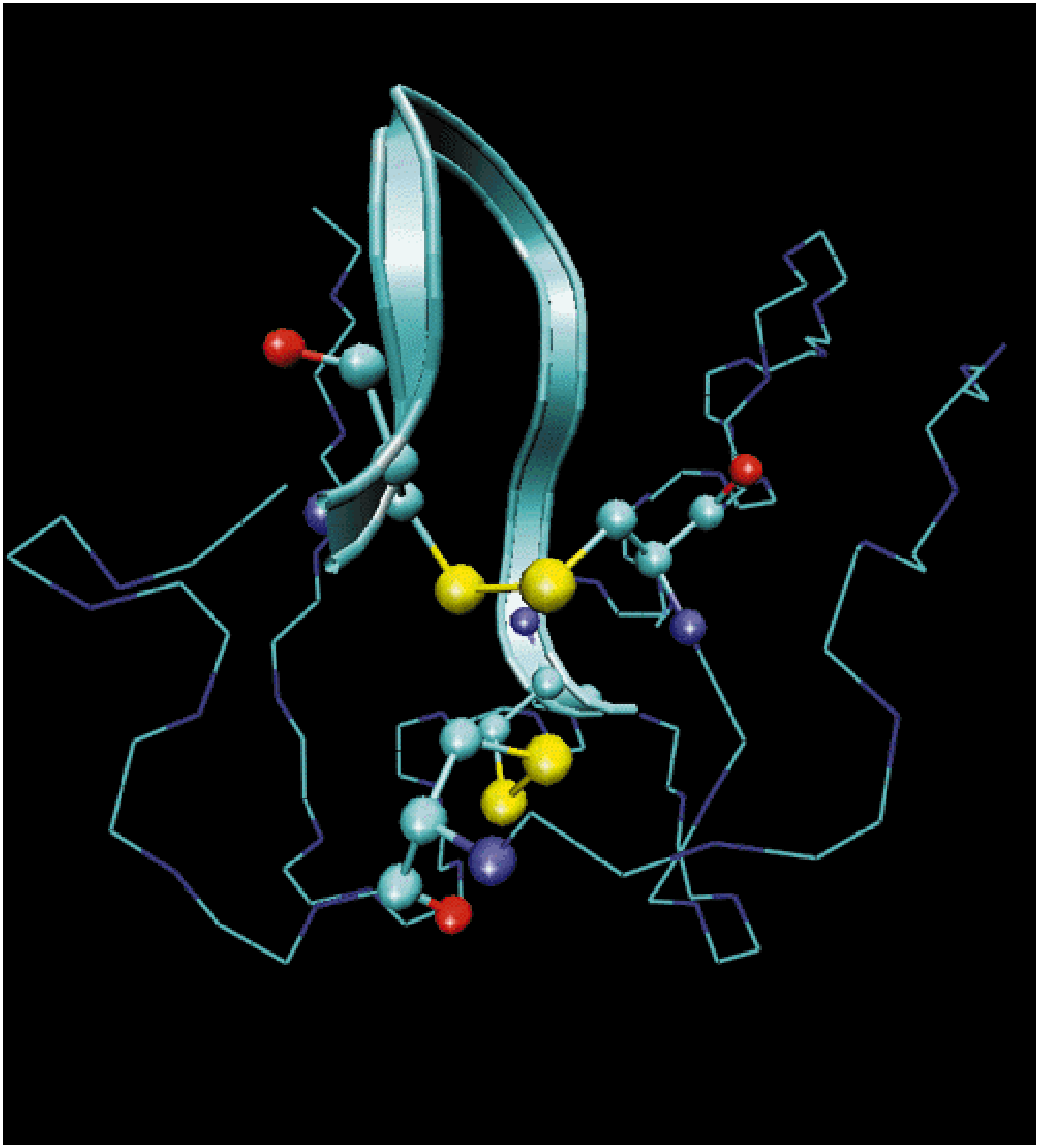


Figure 1. SScorin, a man-made protein showing cysteine residues forming disulfide bridge, loop region as ribbon, and remainder of protein backbone as bonds. Solvent present in the simulation is not shown.



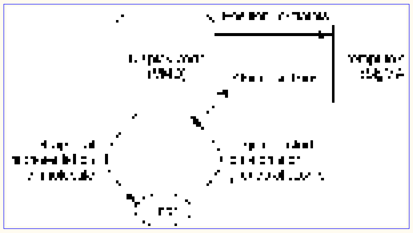


Figure 2. The SMD interactive molecular dynamics system.

5. J. Hermans, "Sigma Documentation," Univ. of North Carolina, URL: http://femto.med.unc.edu/SIGMA/, 1994.

6. M. Nelson et al., "MDScope: A Visual Computing Environment for Structural Biology," *Computer Physics Comm.*, Vol. 91, 1995, pp. 111-134.

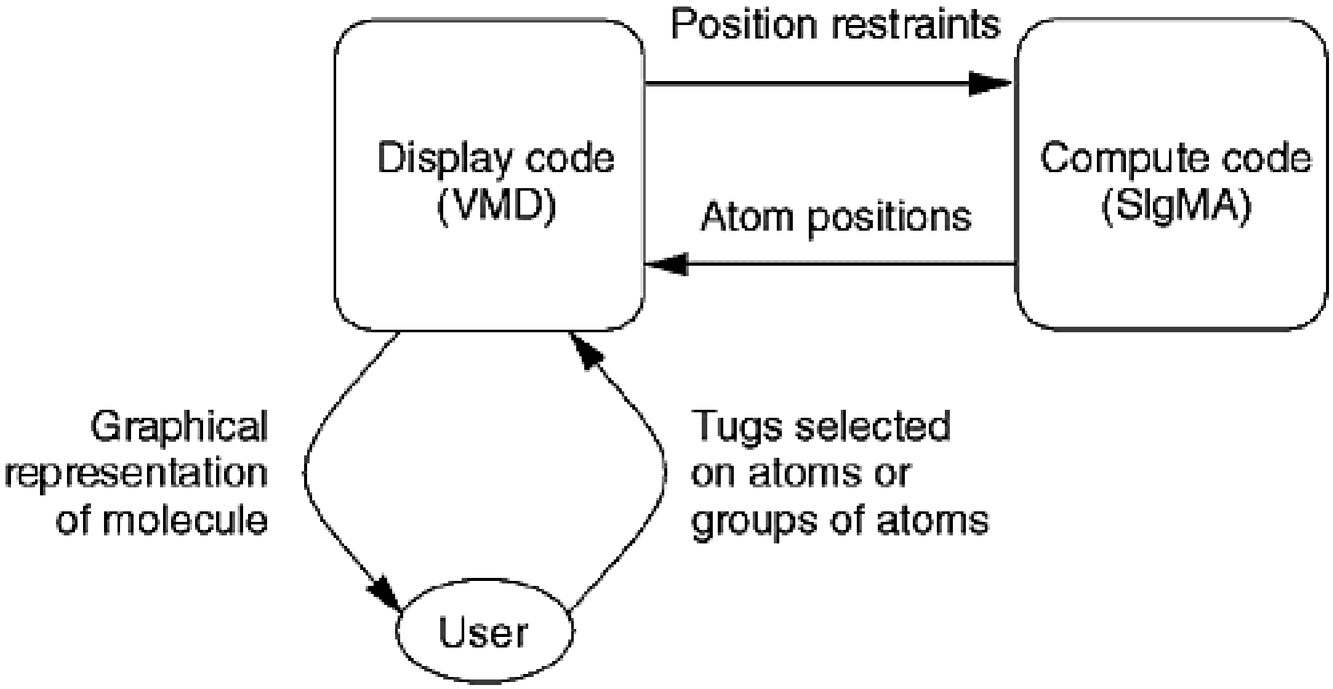

Position restraints
Display code
(VMD)
Compute code
(SIgMA)
Atom positions
Graphical
representation
of molecule
Tugs selected
on atoms or
groups of atoms
User

Table 1. SMD molecular dynamics performance for some workstation CPUs.

		Performance
		(atomic
	Processor and	**updates**
	clock speed	**per second)**
	SGI R4400-250	6,000
	HP PA7100-125	11,000
	HP PA7200-100	12,000
	SGI R8000-90	15,000

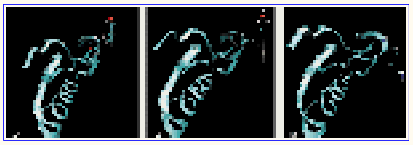


Figure 3. (a) Specifying a "tug": selected atom is highlighted as the red sphere at upper right. (b) Moving the tug: target position follows pointer. (c) System response after 300 fs of simulation.

(a)
(b)
(c)

7. H.J.C. Berendsen et al., "Molecular Dynamics with Coupling to an External Bath," *J. Chemical Physics*, Vol. 81, 1984, pp. 3684-3690.

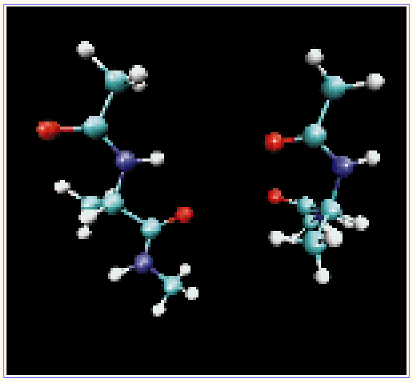
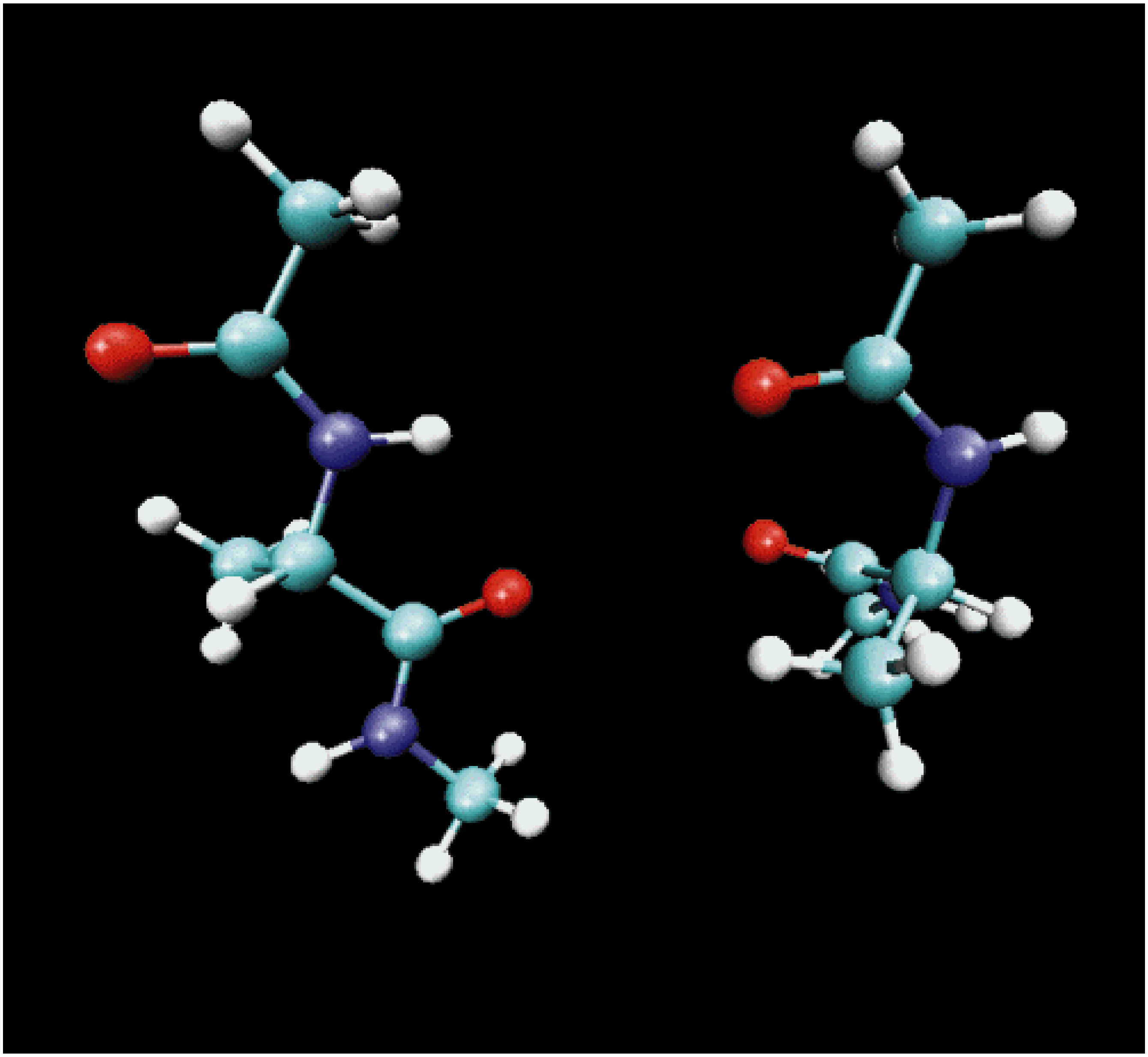


Figure 4. Before *(left)* and after *(right)* diagrams of dipeptide structure. Note position of oxygen atoms (red).



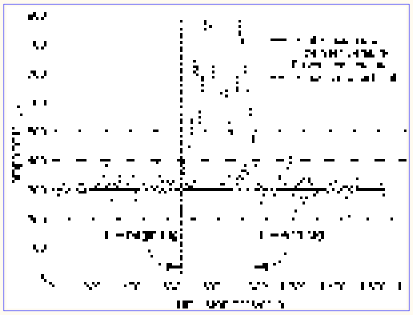
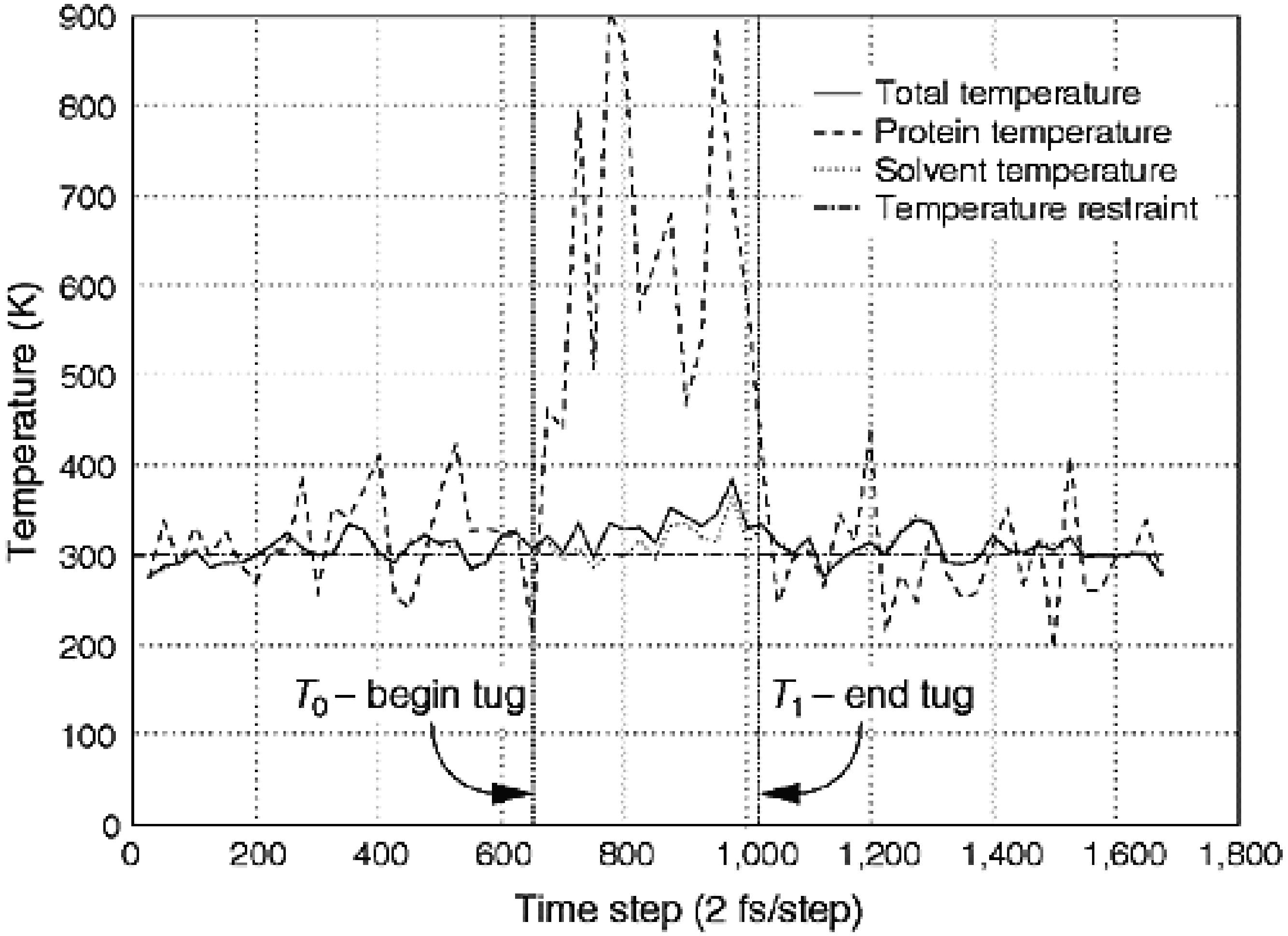


Figure 5. Graph of dipeptide system temperature during manipulation.


900
800
700
600
500
400
300
200
100
0
Temperature (K)
0
200
400
600
800
1,000
1,200
1,400
1,600
1,800
Time step (2 fs/step)
Total temperature
Protein temperature
Solvent temperature
Temperature restraint
T_0 – begin tug
T_1 – end tug

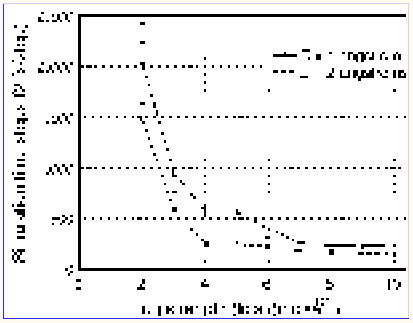
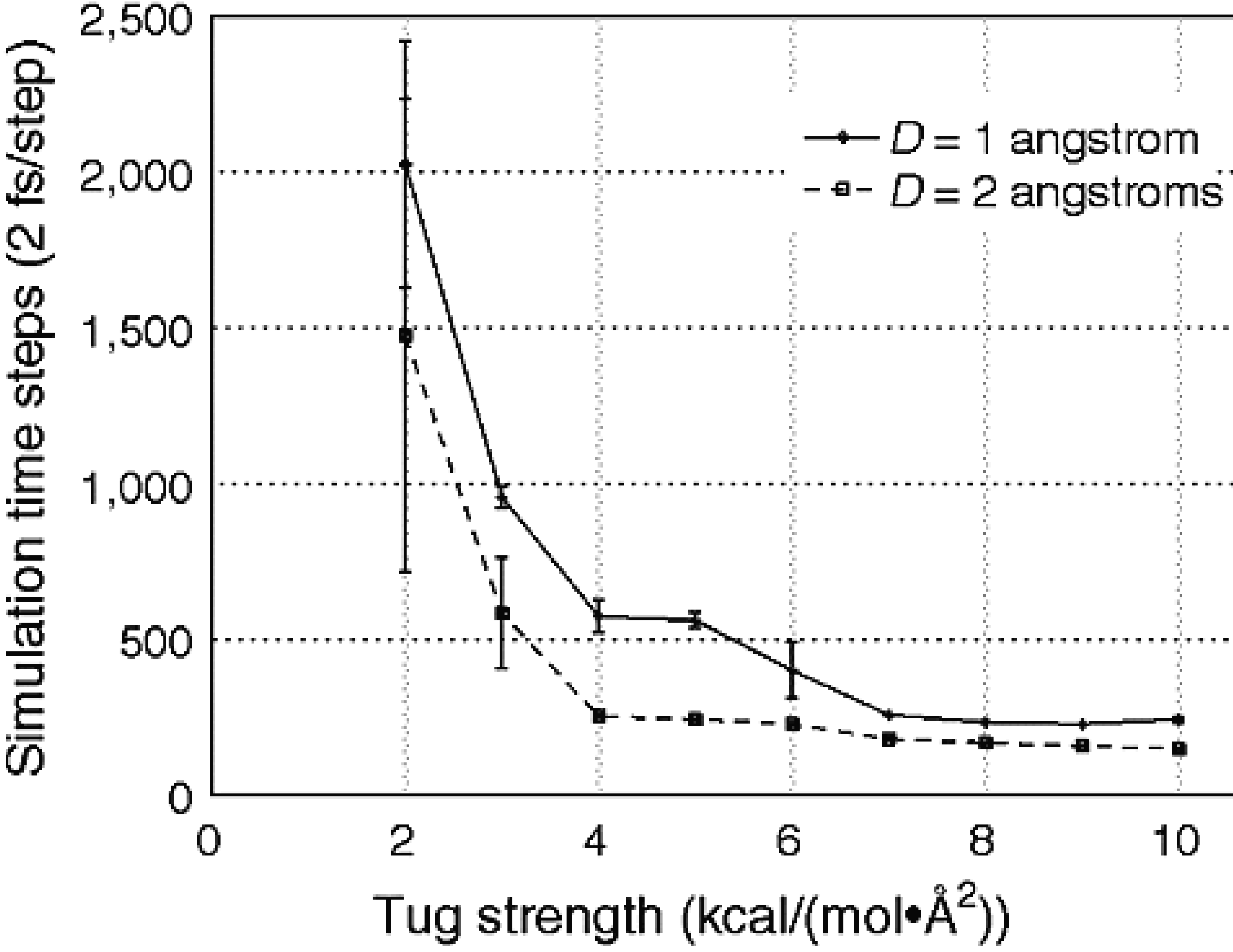


Figure 6. Time to complete a manipulation versus restraint strength.


Simulation time steps (2 fs/step)
2,500
2,000
1,500
1,000
500
0
0
2
4
6
8
10
Tug strength (kcal/(mol•Å^2))
D = 1 angstrom
D = 2 angstroms

9. H. Grubmüller, B. Heymann, and P. Tavan, "Ligand Binding: Molecular Mechanics Calculation of the Streptavidin-Biotin Rupture Force," *Science*, Vol. 271, 1996, pp. 997-999.

10. V.T. Moy, E. Florin, and H. Gaub, "Intermolecular Forces and Energies Between Ligands and Receptors," *Science*, Vol. 266, 1994, pp. 257-259.