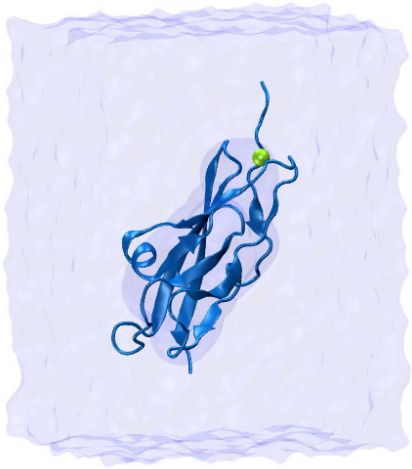
The Ohio State University
Department of Chemistry
and Biochemistry

Hands-on Introduction to Protein Simulations



NAMD Developer:
James Phillips

Tutorial Contributors:
Raúl Araya-Secchi, Michelle Gray, Marcos Sotomayor

October 2014

This tutorial is based on a NAMD tutorial version created by Timothy Isgro, James Phillips, Marcos Sotomayor, Elizabeth Villa, and Klaus Schulten at the Theoretical and Computational Biophysics Group (UIUC), which is available at
`http://www.ks.uiuc.edu/Training/Tutorials/`

Contents

Introduction

This tutorial provides a first introduction to molecular dynamics simulations of proteins using NAMD. *It is based on a version created at the Theoretical and Computational Biophysics Group at The University of Illinois at Urbana-Champaign. The tutorial assumes that you already have a working knowledge of protein visualization using VMD and that NAMD 2.10b1 or a later version has been correctly installed on your computer.*

For the accompanying tutorial on protein visualization using VMD go to

https://research.chemistry.ohio-state.edu/sotomayor/teaching/

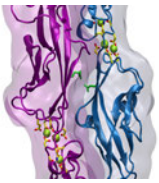
If you do not have a working knowledge of VMD, the tutorial above is a must.

For NAMD's installation instructions, please refer to the NAMD User's Guide:

http://www.ks.uiuc.edu/Research/namd/current/ug/

The tutorial covers the basic steps of a molecular dynamics simulation, i.e., system setup followed by minimization and equilibration. Brief descriptions of basic files needed for the simulations are provided in the appendices.

The examples in the tutorial will focus on the study of two tip link cadherin proteins that mediate hearing. Throughout the text, some material will be presented in separate "boxes". Some of these boxes include complementary information to the tutorial, such as information about the biological role of "tip links", and tips or shortcuts for using NAMD. These boxes are not required for understanding the tutorial and may be skipped if you are short on time.



Tip Links. This tutorial will focus on setting up and running all-atom molecular dynamics simulations of *tip link cadherins* using NAMD. Tip links are essential for hearing and balance, and are made of two proteins interacting tip to tip: cadherin-23 (Cdh23) and protocadherin-15 (Pcdh15). These are large proteins with long extracellular domains formed by extracellular cadherin (EC) repeats. The primary function of the tip link is to convey forces to the mechanosensitive channels that mediate our senses of hearing and balance. Mechanotransduciton mediated by tip links can occur at the microsecond time scale, similar to what can be achieved with molecular dynamics simulations.

Boxes with an exclamation sign are especially important and should not be skipped.

Warning!. The goal of this tutorial is to introduce you to basic molecular dynamics simulations of proteins. In some cases, simulations would take too long to run on your laptop or personal computer, so example-output files are provided for you to inspect. In addition, some examples are optimized so simulations can be done in a reasonable period of time on a common computing facility. This means that some parameters and conditions under which simulations are done in this tutorial are not suitable for scientific studies. Whenever this happens it will be pointed out and alternatives or more appropriate parameters/conditions will be provided in case you want to improve the simulations and/or you have more computer power available.

Basic command line instructions assume the tutorial is being done in a UNIX environment (Linux, Mac OS X terminal, or alike). A web page of basic UNIX commands has been made available at

http://www.ks.uiuc.edu/Training/Tutorials/Reference/unixprimer.html.

If you are using Microsoft Windows or Mac OS X, pay special attention to the following boxes:

Windows Users. If you are using a version of Microsoft Windows, you may check these boxes for alternate, Windows-specific instructions required to perform basic command line instructions. For instance, NAMD must be run from the MS-DOS command line. Therefore, we will also be using the command line to perform tasks, such as copying files, which you might otherwise do in Windows using the mouse.

OS X Users. If you are using a version of Mac OS X, you may check these boxes for alternate, OS X-specific instructions required to launch programs using OS X.

Required programs

The following programs are required for this tutorial:

- **NAMD 2.10b1:** Available at `http://www.ks.uiuc.edu/Research/namd/` (for all platforms). Make sure that your NAMD binary is in a directory that is included in the $PATH environment variable.

NAMD directory. We recommend that Windows users install NAMD in the following directory: C:\Users*username*\namd. The tutorial instructions assume you have installed it in this location.

NAMD directory. We recommend that OS X users install NAMD in the following directory: ∼/Desktop/namd/. The tutorial instructions assume you have installed it in this location.

- **VMD 1.9.1:** Available at `http://www.ks.uiuc.edu/Research/vmd/` (for all platforms)
- **Text Editor:** Nedit is a text editor which will be used throughout this tutorial to view and edit some of the files associated with the simulations. There are others such as pico, emacs, jot, and vi. Feel free to use whichever text editor you are most comfortable with.

Text Editor. Windows users will use WordPad to view and edit some of the text files associated with the NAMD simulations. You may prefer to use Notepad or another text editor of your choosing. Microsoft Word is a word processing program that should not be used as a text editor.

- **Plotting Program:** We will use the free program xmgrace, available at `http://plasma-gate.weizmann.ac.il/Grace/`, to view and analyze output data from NAMD simulations. VMD also has an internal plotting program which may be used to examine output directly from NAMD log files. Other useful graphing programs are Mathematica, `http://www.wolfram.com/` (Purchase required), Matlab, `http://www.mathworks.com/` (Purchase required), and gnuplot, `http://www.gnuplot.info/` (Free download).

Plotting Program. We will use Microsoft Excel, available at `http://office.microsoft.com/en-us/FX010858001033.aspx`, to view and analyze output data from NAMD simulations. Another graphing programs which you may find useful is scilab, available at `http://www.scilab.org/`. Scilab is a free program with capabilities nearly identical to Matlab. If Excel or any of the proprietary programs are unavailable to you, we encourage you to use scilab.

Required Files

- In addition to downloading the pdf for this tutorial, you will need to download the appropriate files required to go through it. Download, unzip, and place them in a directory that is easy to access from the command line. For instance, you may want to place them in:

  `/`*`easyaccess`*`/namd-tutorial-files`

  The files for this tutorial are available at

  `https://research.chemistry.ohio-state.edu/sotomayor/`

Namd tutorial files location. You should download the files and unzip them in a convenient location within the Windows directory structure. For instance, you may want to place them in:
C:\Users*username*\namd-tutorial-files

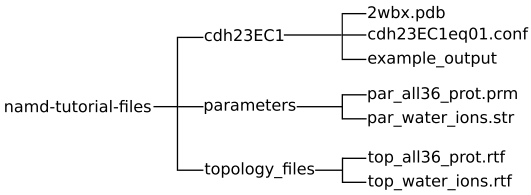


Figure 1: Directory Structure for tutorial exercises. Output for all simulations is provided in an "example-output" subdirectory within each folder.

1 Basics of Protein Simulations with NAMD

In this section you will learn how to setup a basic system for molecular dynamics (MD) simulations and how to run a NAMD simulation of it. You will learn about typical NAMD input and output files for energy minimization and equilibration of a protein fragment (Cdh23 EC1) in a water box.

1.1 What is Needed

To setup a system for MD simulation you will need at least five files:

- a Protein Data Bank (pdb) file which usually stores atomic coordinates of the system. Pdb files may be generated by hand, but they are also available via the Internet for many proteins at `http://www.pdb.org`. More in Appendix A.
- a Protein Structure File (psf) which stores structural information of the protein, such as the connectivity among atoms. More in Appendix B.
- a force field topology file. This file contains information on atom types, charges, and how the atoms are connected in a given amino acid.
- a force field parameter file. This file contains all parameters (bond strengths, equilibrium lengths, etc.) that define the potential energy which atoms in the system experience according to a given mathematical formula. CHARMM, X-PLOR, AMBER, and GROMACS are four types of force fields, with their own sets of parameters and similar mathematical formulae for the potential energy. NAMD is able to use any of them.
- a configuration file, in which the user specifies all the options that NAMD should adopt in running a simulation. The configuration file tells NAMD how the simulation is to be run.

1.2 Making a Protein Structure File (PSF) for Cdh23 EC1

Of the five files mentioned above, an initial pdb file will typically be obtained through the Protein Data Bank, and the parameter and topology files for a given class of molecule may be obtained from:

`http://mackerell.umaryland.edu/CHARMM_ff_params.html`.

The psf file must be created by the user from the initial pdb and topology files. Our first two steps are to launch VMD and to load a pdb file of our molecule of interest (Cdh23 EC1, pdb code 2WBX).

1 Open VMD by typing `vmd` in the Terminal window.

```
> vmd
```

Launch VMD. Open VMD graphically by clicking on the Start → Programs → University of Illinois → VMD menu item.

Launch VMD. Open VMD graphically by double-clicking on the VMD icon under `Applications` in the `Finder`.

2 Open the Molecule File Browser from the File → New Molecule... menu item in the VMD Main window.

3 Type `2WBX` in the Filename: text entry and then press the Load button. If a connection to the internet is not available, browse for the file `2WBX.pdb` in `namd-tutorial-files/Cdh23EC1`, then click on the Load button.

Now the Cdh23 EC1 protein fragment is shown in the OpenGL Display window. You may close the Molecule File Browser window at any time.

We will also set the output directory to ensure that all your output files are stored in an appropriate place.

4 In the VMD Main window, select the Extensions → Tk Console menu item. A console window should appear with a prompt.

5 In the Tk Console window, type:

```
cd /easyaccess/namd-tutorial-files/Cdh23EC1/
ls
pwd
```

where `cd` stands for "change directory" and *easyaccess* is the name of the directory where your namd tutorial files were downloaded to (see "Required Files" section above).

After typing `ls`, you should see a list of files, including `2WBX.pdb`. After typing `pwd`, which stands for "print working directory", you should see the path you typed in after `cd`. All VMD output will now be directed to `/easyaccess/namd-tutorial-files/Cdh23EC1/`.

Setting output directory. Windows users should type:
`cd C:\Users\username\namd-tutorial-files\Cdh23EC1`
to change to the appropriate directory.

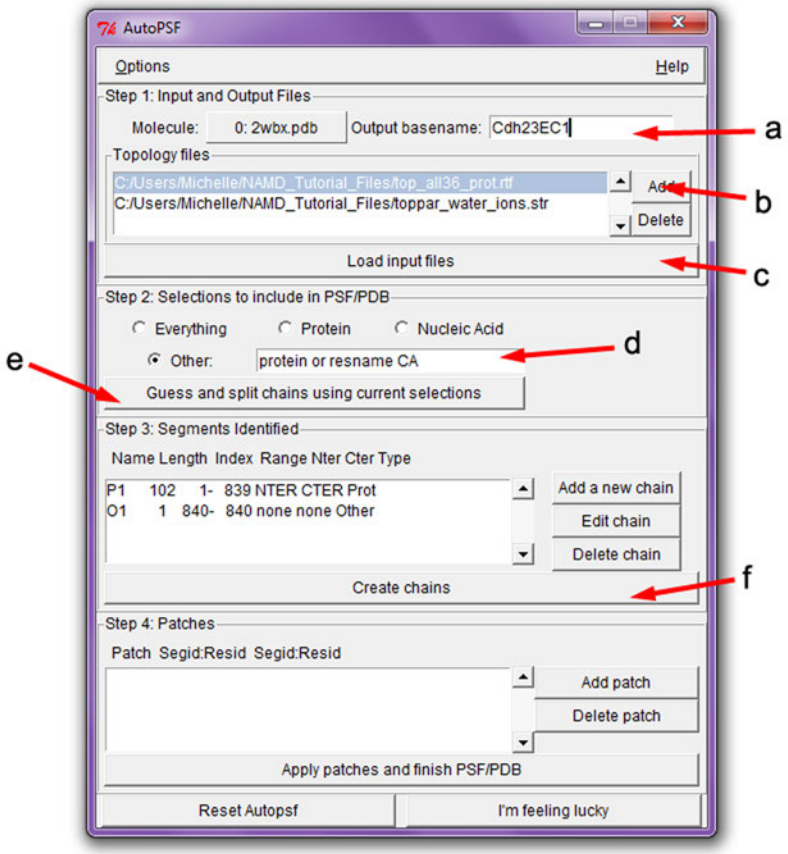


Figure 2: Automatic PSF builder.

1.2.1 AutoPSF Step 1

Let's use the VMD extension called AutoPSF to build our protein structure file:

1 In the VMD Main window, open the AutoPSF window from the Extensions → Modeling → Automatic PSF builder menu item (Fig. 2).

You will set Input and Output Files parameters in Step 1.

2 Delete the content of the Output basename: text entry and replace it by `Cdh23EC1` (Fig. 2a). All output files will use this name.

3 Click on the item displayed in the Topology files listing window. Click on the Delete next to it to eliminate this default topology file.

4 Click on the Add button next to the Topology files listing window (Fig. 2b). Browse and select the file: `top_all36_prot.rtf` located in the `topology_files` directory.

5 Click on the Add button again and browse for the file `toppar_water_ions.str` in the `topology_files` directory. You may need to enable selection of "All" files in your file browser to be able to select `toppar_water_ions.str`.

The two topology files you just added are part of the latest release of the CHARMM (Chemistry at HARvard Molecular Mechanics) forcefield, maintained by the Mackerell lab in coordination with the Karplus group. These files correspond to version 36.

Force Field Updates. Protein force fields have been developed and improved over more than three decades. Recent updates include a so-called CMAP term that is an energy correction map based on quantum mechanical calculations. It improves protein backbone behavior and thus yields more accurate dynamic properties for the protein. Make sure you use the latest release compatible with VMD and NAMD.

6 To finalize Step 1, click on Load input files (Fig. 2c). The Tk Console window should display `WORKING ON: 0` .

1.2.2 AutoPSF Steps 2, 3, and 4

In this step you will pick which selections (parts of your pdb file) will be included in the final files to be used for simulations.

1 Click on the Other: button and type `protein or resname CA` (Fig. 2d). This will select the protein and calcium ions (somehow labeled `CA` by crystallographers, do not get confused with C_α atoms!).

Selections. The PDB file that you are using might have multiple chains, crystallographic water molecules, bound ions, etc. You may want to keep all components that are relevant for comparisons between your simulation and *in vivo* or *in vitro* experimental conditions. For instance, for Cdh23 EC1 you may want to try `protein or resname CA or water` to preserve crystallographic water molecules.

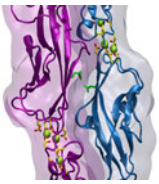
2 To finalize Step 2, click on the Guess and split chains using current selections button (Fig. 2e).

The listing window in Step 3 should get populated with two segments, `P1` and `O1`, for protein and calcium atoms present in the PDB file, respectively (Fig. 2).

3 Select each of the listed segments to see them in Van der Waals (VDW) representation in the OpenGL Display window.

Note that you can add, edit, and delete segments (chains) using the buttons on the right. In this case we will continue with the guessed chains.

Step 4 in the AutoPSF window involves applying so-called "patches" to the structure. These patches can be used to define disulfide bonds, modify N or C termini, protonate certain residues, etc. We do not require patches for the simple Cdh23 EC1 protein fragment and will ignore Step 4.



Histidine Residues. Of the 20 amino acids, histidine is the only one which ionizes within the physiological pH range (∼7.4). This effect is characterized by the pK_a of the amino acid side chain. For histidine, the value is 6.04. This leads to the possibility of different protonation states for histidine residues in a protein, and makes the consideration of the proper state important in MD simulations. The viable states are one in which the δ nitrogen of histidine is protonated (listed with residue name "HSD" in the topology file), one in which the ϵ nitrogen of histidine is protonated ("HSE"), and one in which both nitrogens are protonated ("HSP"). You can use AutoPSF and patches to select for specific protonation states of histidines and other residues.

4 To finalize all steps click on the Create chains in Step 3 of the AutoPSF window (Fig. 2f).

AutoPSF will display a warning message stating that no patches were automatically assigned and that the complete psf and pdb files will be generated at this step.

5 Click the OK button.

Also, the following message will pop-up:

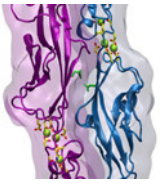
```
   Structure complete.  Your structure is in
Cdh23EC1.pdb
Cdh23EC1.psf
Please remember that while AutoPSF usually makes correct guesses about
things, its accuracy cannot be guaranteed.  You should always inspect
your structure prior to simulations; in particular, please check that
the appropriate termini were used in the appropriate places, and that
the chains have been split properly.
```

6 Click the OK button.

As indicated, AutoPSF generated a pdb/psf file pair, which will also be loaded into VMD with molecule ID 2 (see the VMD Main window). In addition, AutoPSF generates a third file (`Cdh23EC1.log`) that contains all details of what happened "behind the scenes" (also shown in the Tk Console window; Fig. 3). Warning and error messages are shown there, so you may want to inspect this file carefully.

7 Look at the loaded structure in the OpenGL Display window. Make sure to zoom in and inspect different amino acids. Do **not** close VMD after you are done looking.

The pdb/psf pair you generated contains hydrogen atoms that were not present in the X-ray crystal structure of Cdh23 EC1 downloaded from the Protein Data Bank. This is because X-ray crystallography usually cannot resolve hydrogen atoms. The coordinates for the hydrogen atoms that you see in our processed files are guessed by VMD. Later, energy minimization of the protein will ensure their positions are reasonable.



X-ray Crystallography. X-ray crystallography methods utilize the optical rule that electromagnetic radiation will interact most strongly with matter the dimensions of which are close to the wavelength of that radiation. X-rays are scattered by the electron clouds in molecules, since both the wavelength of the X-rays and the diameter of the cloud are on the order of Angstroms. The diffraction patterns formed when a group of molecules is arranged in a regular, crystalline array, may be used to reconstruct a 3-D image of the molecule. Hydrogen atoms, however, are not typically detected by X-ray crystallography since their sizes are too small to interact with the radiation and since they contain only a single electron. The best X-ray crystallography resolutions currently available are around 0.9Å.

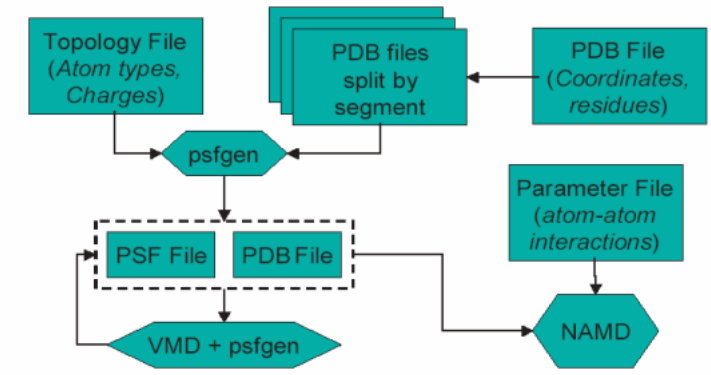


Figure 3: Flowchart indicating the role of files as used by VMD, NAMD, and AutoPSF using `psfgen`.

1.3 Centering Cdh23 EC1

The geometrical center of the coordinates of protein atoms obtained using X-ray crystallography is usually off-origin ($x = 0$, $y = 0$, $z = 0$), as it reflects

the position of one protein molecule in a periodic arrangement of molecules that form a crystal lattice. It is often convenient to center the protein being simulated on the origin, as centered systems are easier to handle and analyze (both for humans and computers!). We will use several `Tcl` commands to do this:

1 In the currently open VMD session, locate the Tk Console window and type:

```
set all [atomselect top all]
```

This will create a selection containing all the atoms of the current top molecule (Cdh23EC1).

2 To move the molecule to the origin, type:

```
$all moveby [vecinvert [measure center $all]]
```

You should be able to see Cdh23 EC1 moving to the origin.

3 To check that the Cdh23 EC1 is actually centered, type:

```
measure center $all
```

The measured center should be close to ($x = 0$, $y = 0$, $z = 0$). Note the scientific notation used in the output, where $e - 6$ stands for 10^{-6}.

4 Save the new coordinates by typing:

```
$all writepdb Cdh23EC1.pdb
```

Note that you overwrote your initial file and VMD did not provide any warning message. You might be specially careful when using the `writepdb` command.

5 Before moving on to the next section, delete all molecules by typing:

```
mol delete all
```

1.4 Solvating Cdh23 EC1 in a Water Box

Proteins are usually in a liquid environment, so to simulate Cdh23 EC1 in a realistic environment we will solvate it, i.e., put it inside a water box that closely resembles the cellular environment. To do so, we will load our newly created files and use the VMD Solvate extension:

1 Open the Molecule File Browser from the File → New Molecule... menu item in the VMD Main window. Browse for the file `Cdh23EC1.psf` in `namd-tutorial-files/Cdh23EC1`, then click on the Load button. You have just created a new molecule with a structure (connectivity) but no coordinates (nothing will appear on the screen!).

Keep the Molecule File Browser window open. Notice that the menu at the top of the window now says `0: Cdh23EC1.psf`. This ensures that the next file that you load will be added to that molecule. You **do not** need to change this menu to `New Molecule`, since you are merging data with the psf file.

2 In the Molecule File Browser window, click on the Browse button. Make sure that `Cdh23EC1.psf` is selected on the Load files for: menu. Browse for `Cdh23EC1.pdb` in `namd-tutorial-files/Cdh23EC1`, click OK, and click on the Load button again. You will be able to see the centered structure, with hydrogen atoms, appear in the OpenGL Display window.

3 In the VMD Main window, click on the Extensions → Modeling → Add Solvation Box menu item. The Solvate window should appear (Fig. 4).

The Input section should be automatically populated with names for our psf and pdb files (`Cdh23EC1.psf/pdb`).

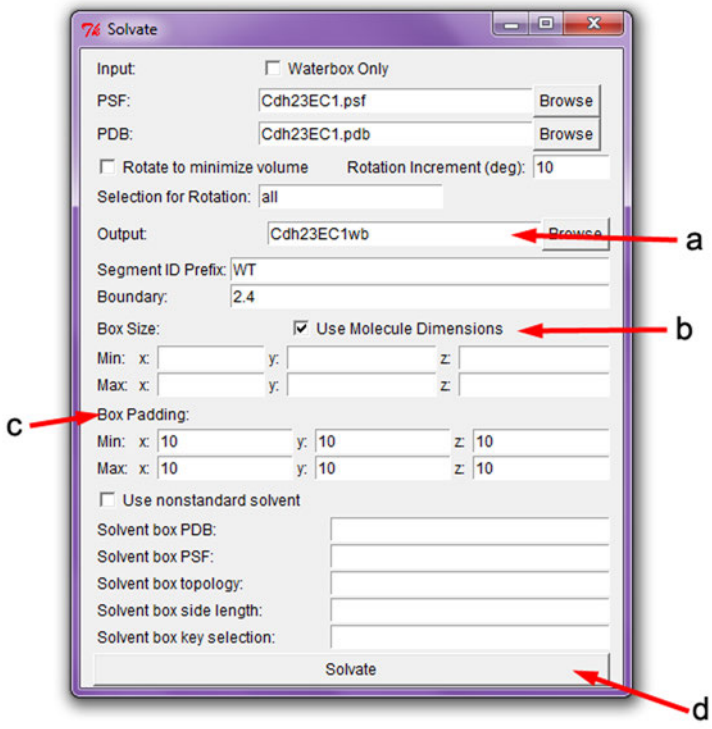


Figure 4: Solvate plugin.

4 In the Output: text entry (Fig. 4a), delete `solvate` and replace it by `Cdh23EC1wb`. This way we will store our solvated molecule with a well defined name.

5 Make sure that the Use Molecule Dimensions option stays checked (Fig. 4b).

6 In the Box Padding: section (Fig. 4c), fill all boxes with a value of 10. This will add a layer of water molecules that goes 10 Å beyond the protein in each direction.

7 Clik on the Solvate button. A solvated `Cdh23EC1wb` molecule should appear in the OpenGL Display window (Fig. 5, left panel). A `Cdh23EC1wb.log` file will be created as well, containing all details of the solvation process.

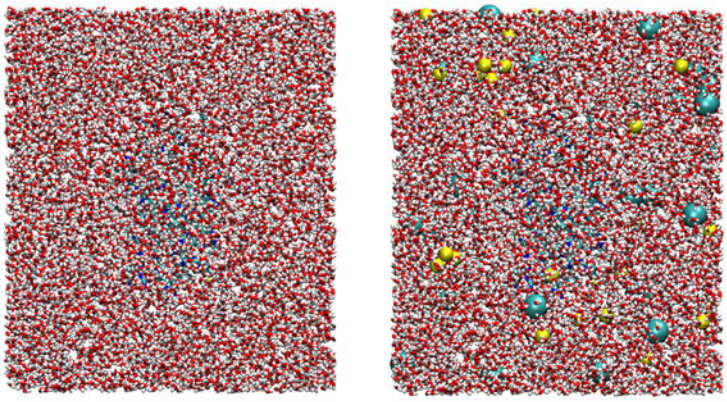


Figure 5: Solvated Cdh23EC1wb shown in licorice representation. Ionized system is shown on the right panel, with sodium ions shown in yellow, and chloride ions shown in cyan.

The water molecules that now surround Cdh23 EC1 were placed there by avoiding clashes with the protein and by trying to keep a realistic water density. Before moving on with our system preparation, we will measure the size of our solvation box:

8 Locate the Tk Console window and type:

```
set all [atomselect top all]
```

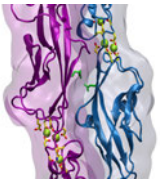
This will create a selection containing all the atoms of the current top molecule (Cdh23EC1wb).

9 To measure the size and center of the solvated system, type:

```
measure minmax $all
measure center $all
```

Note that the `minmax` command, which gives you the minimum and maximum values of x, y and z coordinates of the entire protein-water system, can be used to estimate the size of the box (approximately $48.2 \times 55.9 \times 74.2$). Check it! The measured center should still be similar to ($x = 0$, $y = 0$, $z = 0$). Make note of the values you see in your Tk Console window as they will be used in the configuration file below.

Size of Water Box. The water box created for Cdh23EC1 is somewhat small. Ideally, one should work with a box which is large enough so that the protein does not interact with its periodic image (see description of periodic boundary conditions below). On the other hand, the larger the box, the more expensive the computations required to carry out our simulation. Also note that if the protein is going to be pulled, the box should be large enough to accommodate the fully extended state of the protein.



Implicit Solvent. An implicit solvent model is a simulation technique which eliminates the need for explicit water atoms by including many of the effects of solvent in the inter-atomic force calculation. For example, polar solvent acts as a dielectric and screens (lessens) electrostatic interactions. The elimination of explicit water accelerates conformational explorations and sometimes increases simulation speed at the cost of not modeling the solvent as accurately as explicit models.

1.5 Adding Ions to the Solvated Cdh23 EC1

Ions should be placed in the water box to mimic a physiologically relevant environment. They are especially necessary if the protein being studied carries an excess charge. In that case, the number of ions should be chosen to make the system neutral as well. These ions will shield the regions of the protein which carry the charge, and make the entire system more stable. We will add ions in random positions using the VMD Autoionize extension.

1 In the VMD Main window, click on the Extensions → Modeling → Add Ions menu item. The Autoionize window should appear (Fig. 6).

The Input section should be automatically populated with names for our psf and pdb files (`Cdh23EC1wb.psf/pdb`).

2 In the Output prefix: text entry (Fig. 6a), delete `ionized` and replace it by `Cdh23EC1wbi`. This way we will store our ionized and solvated system with a well defined name.

3 In the Choose salt: menu item (next to Output prefix), keep `NaCl`.

4 In the Ion placement mode section select `Neutralize and set NaCl concentration to 0.15 mol/L` (Fig. 6b).

5 Click on the Autoionize button (Fig. 6c).

A ionized Cdh23EC1wbi molecule should appear in the OpenGL Display window (Fig. 5, right panel). Details of the process will be shown in the Tk Console window.

Now you have a pdb and psf pair for the solvated and ionized Cdh23 EC1. Let's mov on to setup your configuration file.

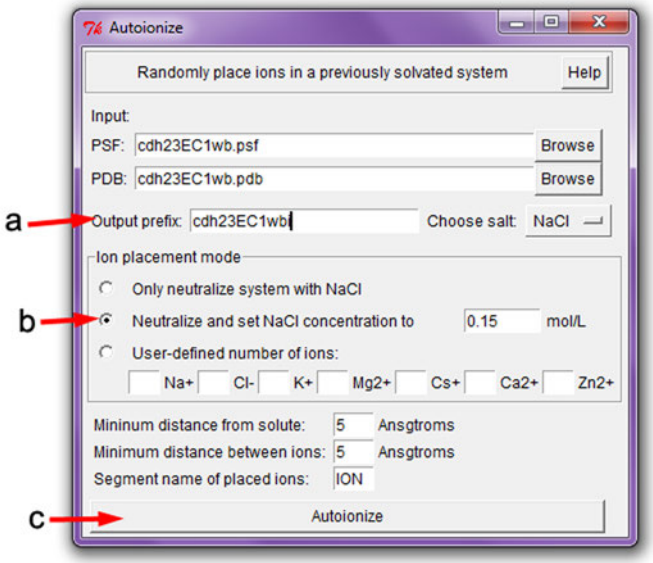
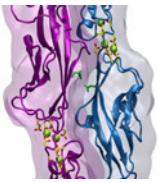


Figure 6: Autoionize plugin.

6 Close VMD by choosing File → Quit in the VMD Main window.



Physiological Ionic Concentrations. The concentration of different ions can vary widely depending on the physiological context. For instance, the extracellular environment usually has 150 mM NaCl, the concentration we use in this tutorial. However, the cochlear endolymph that baths inner-ear tip links has 150 mM KCl. You may want to pick the type and concentration that matches best the physiological environment of your protein.

Placement of ions. Autoionize places ions randomly around the water box, assuming that these ions will quickly go to regions of potential minima. However, more advanced tools for placement of ions can compute the location of these minima and place ions directly there. This approach might be particularly important for DNA and RNA molecules.

1.6 Your First Configuration File

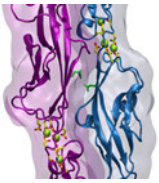
The last step before running your own simulation of Cdh23 EC1 in a water box is to set a NAMD configuration file. In the next section you will edit a NAMD configuration file provided to you to gain a beginner's understanding of its content.

Configuration Warning. Be very careful when typing commands into your configuration files. Even the slightest typographical error will cause the simulation not to run. The point is not simply to run a simulation, but to run it correctly. Be sure your file is correct before you submit a simulation to be run!

The configuration file is complete and could be submitted *as is*. However, we will examine its content line by line before submission.

1 Go to the `namd-tutorial-files/Cdh23EC1` directory and type `nedit cdh23EC1eq01.conf.txt` to open the corresponding configuration file prepared for the minimization and equilibration of Cdh23 EC1 in a water box.

Configuration File. Open WordPad by clicking Start → Programs → Accessories → WordPad. Open the configuration file, `cdh23EC1eq01.conf.txt` in your C:\Users*username*\namd-tutorial-files\Cdh23EC1 directory.



Minimization and Equilibration. Minimization and equilibration differ from each other by the nature in which they implement the molecular dynamics force field. Energy minimization involves searching the energy landscape of the molecule for a local minimum, i.e., a place in which the molecule is relaxed, by systematically varying the positions of atoms and calculating the energy of the system. Equilibration involves molecular dynamics, whereby Newton's Second Law is solved for each atom in the system to dictate its trajectory. Achievement of equilibrium is judged by how well velocities, pressure, etc. are distributed in the system over a given amount of time.

All output files for the minimization and equilibration of your Cdh23 EC1 system will be placed in the `namd-tutorial-files/Cdh23EC1` directory. The configuration file is one of the input files placed here, along with the pdb and psf pair you created. Parameter files, which may be used by many other simulations, are placed in the `parameter` directory and are *called* by each configuration file.

The configuration file may seem complex at first, but we will go over each command line to ensure that you get a basic understanding of each.

Note that when "#" appears at the beginning of a line, the entire line is treated as a comment and ignored by NAMD. In the middle of a line, ";#" is used to comment out the remainder of the line.

2 The "Job Description" section contains only comments, and its purpose is to inform those who view the configuration file about what it is meant for. Your comment should read

```
# Minimization and Equilibrium MD run of cdh23EC1 (2WBX) ...
# and neutralized with 0.15 M NaCl.
# Isothermic/isobaric ensemble NPT (1atm / 310K)
# 2fs timstep
```

3 The "Adjustable Parameters" section contains five commands and an optional block of commands:

- `structure`: calls the psf file describing the system (`Cdh23EC1wbi.psf`).
- `coordinates`: calls the initial coordinate data from the file listed next to it (in this case `Cdh23EC1wbi.pdb`).
- `outputName`: Several types of output data may be written by NAMD for any simulation. This command specifies the prefix for output filenames. NAMD always returns the following two output files for *every* simulation: a pdb file containing the final coordinates of all atoms in the system and a pdb file containing the final velocities of all atoms in the system; the extensions for these files are `.coor` and `.vel`, respectively. Thus, this configuration file will write at least two files named `Cdh23EC1wbieq01.coor` and `Cdh23EC1wbieq01.vel`.
- `set temperature`: creates a variable called "temperature" in which to store a value for the initial temperature of the system. If you place the text "`$temperature`" in the configuration file, NAMD simply reads it as a label for the number "`310`" (creating variables is useful since you can alter the value of that variable in a single place if it is listed many times in the configuration file).
- `firsttimestep`: simply sets a number value for the first time step of the simulation. It is typically useful when restarting a simulation. For instance, if a previous simulation ended at time step 552, the command `firsttimestep 553` would be used.
- Block starting with `if{0}` {: This is the beginning of the optional block of commands related to restarting simulations (not active in this case). By bracketing a set of commands inside an `if {n}` statement, NAMD will ignore the commands if `n`=0 and read them if `n`$\neq$0.

A Note on Restarting Simulations. It is often necessary or useful to run your simulation in steps. For instance, you may fully equilibrate a protein system and then desire to perform several different types of simulations using this equilibrated structure as a starting point. You can use this block of commands as a template for restarting simulations, but be aware of the following points:

- Change the first instance of "`if {0} {`" to "`if {1} {`" so that NAMD recognizes that it should take coordinates, velocities, and system dimensions from restart files.
- Make sure you do not specify a temperature if you are using a velocity restart file. You should comment out the `temperature` command (elsewhere in the configuration file, see below) by placing a # in front of it.
- For periodic boundary conditions, make sure you do not respecify the periodic cell if you are using restart files, since NAMD will read the dimensions from the restart file.
- Be certain all other simulation conditions accurately match what you would like to accomplish. For example, you may no longer need minimization if your system is already equilibrated.

4 The "Simulation Parameters" section contains many commands, commented into different categories:

- Force field parameters to use
  - `paraTypeCharmm`: indicates whether or not the parameter file is in the format used by the CHARMM force field. `on` indicates that it is; `off` indicates that it is not. If this command is not specified, NAMD assumes the file is in X-PLOR format by default.
  - `parameters`: calls the force field parameters from the file listed next to it. In this case `../parameters/par_all36_prot.prm` (`..\parameters\par_all36_prot.prm`) and `../parameters/par_water_ions.str` (`..\parameters\par_water_ions.str`).

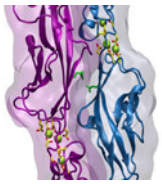
Introduce \. Make sure that directory paths use \ as needed, e.g., in `..\parameters\par_water_ions.str`. If the configuration file does not use \, modify accordingly and save it.

5 Continuation of "Simulation Parameters"

- Global variables
  - `temperature`: sets the initial temperature of the system in Kelvin (K), with the value listed next to it (in this case `$temperature`,

or 310). This is done by assigning random velocities to atoms picked from a Maxwell distribution such that their average kinetic energy accurately represents the given temperature.

- Periodic Boundary Conditions
  - Three periodic cell basis vectors are to be specified to give the periodic cell its shape and size. They are `cellBasisVector1`, `cellBasisVector2`, and `cellBasisVector3`. In this file, each vector is perpendicular to the other two, as indicated by a single x, y, or z value being specified by each. For instance, `cellBasisVector1` is $x = 48.2$Å, $y = 0$Å, $z = 0$Å. With each vector perpendicular, a rectangular 3-D box is formed. Check that what is in the configuration file actually matches the size of your simulation system. This is why we asked you to take note of the size and center of your water box after you solvated your system!
  - `cellOrigin`: specifies the coordinates of the center of the periodic cell in Å.
  - `wrapAll`: This command may be used with periodic boundary conditions. If an atom crosses the periodic boundary, leaving the cell, setting this command to `on` will translate its coordinates to the mirror point on the opposite side of the cell. Nothing can escape. The command may be set to `on` or `off`.



Periodic Boundary Conditions. The use of periodic boundary conditions involves surrounding the system under study with identical virtual unit cells. The atoms in the surrounding virtual systems interact with atoms in the real system. These modeling conditions are effective in eliminating surface interaction of the water molecules and creating a more faithful representation of the *in vivo* environment than a water sphere surrounded by vacuum provides.

6 Continuation of "Simulation Parameters":

- Force-field parameters specified by CHARMM
  - `exclude`: specifies which atomic interactions are to be excluded from consideration. See Fig. 7 for general atom labels. The `scaled1-4` value indicates that interactions between any such atoms 1 and 2 and 1 and 3 are neglected and interactions between atoms 1 and 4 are weakened. The van der Waals interaction for "1-4" atoms are modified using special 1-4 parameters defined in the parameter files, and electrostatic interaction is modified as shown in the next command.
  - `1-4scaling`: specifies the degree to which the electrostatic interaction between 1-4 atoms is to be taken into account. It may be a decimal between 0 and 1 (here, it is 1) and indicates how much the interaction is "turned off" or "on", respectively.

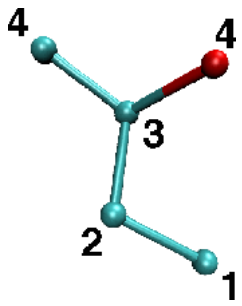


Figure 7: Number labels for atoms which are a given amount of bonds away from one another

- `switching`: indicates whether or not switching functions are used to smoothly take electrostatic and van der Waals interactions to zero at the `cutoff` distance. You may list "on" or "off" next to it as a yes/no answer.
- `cutoff`: indicates the distance in Å beyond which electrostatic and van der Waals interactions are cut-off. Otherwise, those interactions are considered over the entire volume of your system. This can be computationally too costly. The definition changes when the Particle Mesh Ewald Sum is invoked (see below).
- `switchdist`: indicates the distance in Å at which the functional form of electrostatic and van der Waals interactions is modified to allow their values to approach zero at the `cutoff` distance. A visual explanation of this is useful and may be found in Fig. 8.
- `pairlistdist`: is designed to make computation faster. It specifies a distance in Å. NAMD will only search within this distance for atoms which may interact by electrostatic or van der Waals interactions. This way, NAMD does not have to search the entire system. The distance must be greater than the `cutoff` distance, and the list must be updated during the simulation. See Fig. 8.
- `stepspercycle`: atoms are reassigned pair list identities (as explained above) every so often in a cycle. This command specifies how long one cycle lasts, i.e. the number of time steps in one cycle.
- `pairlistsPerCycle`: number of times pairlists are regenerated in a cycle
- `seed`: number used to seed the random number generator if `temperature` or `langevin` is selected. This can be used so that consecutive simulations produce the same results. If no value is specified, NAMD will choose a pseudo-random value based on the current UNIX clock time.
- `vdwForceSwitching`: indicates whether or not a force-based switch-

ing function to the van der Waals term is used. This is required for simulations involving lipids and the CHARMM36 force field.

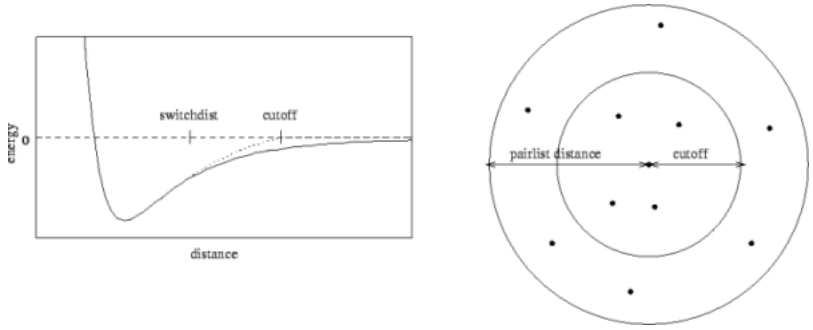


Figure 8: Cutoff and switching distances indicated on the left. Pair list distance indicated on the right.

- Integrator Parameters
  - `timestep`: indicates the value of the time step size used in the simulation. MD simulations solve Newton's equations of motion in a discrete approximation to determine the trajectories of atoms. The time step tells NAMD how to discretize the particle dynamics. It is specified in femtoseconds (here, `2` fs).
  - `rigidBonds`: specifies which bonds involving hydrogen are considered to be rigid (non-vibrating). The value `all` specifies all linear bonds involving hydrogen and any other atoms.
  - `nonbondedFreq`: specifies in number of time steps how often nonbonded interactions should be calculated. It is useful for saving computational time.
  - `fullElectFrequency`: specifies in number of time steps how often full electrostatic interactions should be calculated.
  - `longSplitting`: specifies the method used to split electrostatic forces between long and short range potentials. The `C2` option produces demonstrably better long time stability.

Rigid Bonds. The time step used in any MD simulation should be dictated by the fastest process (i.e. movement of fastest atoms) taking place in the system. Among the various interactions, bond stretching and angle bending are the fastest, with typical bond stretching vibrations occurring on the order of once every 10-100 femtoseconds. Using a time step of 2 fs, which is close to the vibrational period (10 fs) of linear bonds involving hydrogen (fastest vibrations, since hydrogen has a small mass), requires that these bonds be fixed, and only slower vibrations may be free to move, such as dihedral angle bending. For large molecules, these slower vibrations typically govern the behavior of the molecule more than the quicker ones, so bond fixing is somewhat acceptable, but should be avoided for accurate simulations. One prefers to use an MD timestep which is $\sim 1/10$ of the fastest interactions in the simulation. The `rigidBonds` option should always be used for water (TIP3P) because water molecules in this model were parametrized as rigid molecules.

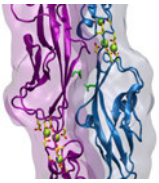
7 Continuation of "Simulation Parameters":

- PME (for full-system periodic electrostatics) PME stands for Particle Mesh Ewald. It is a useful method for dealing with electrostatic interactions in the system when periodic boundary conditions are present. The Ewald sum is an efficient way of calculating long range forces in the periodic system. The particle mesh is a 3-D grid created in the system over which the system charge is distributed. From this charge, potentials and forces on atoms in the system are determined. As a result, your grid size should be chosen such that it is fine enough to accurately represent the configuration of your system.
  - `PME`: indicates whether or not the simulation uses the Particle Mesh Ewald Sum method; uses values `yes` and `no`.
  - `PMEGridSpacing`: sets the minimum ratio between the number of PME grid points along each `cellBasisVector` and the physical dimensions. Since the grid will replicate the charge distribution in your system, `PMEGridSpacing` should be chosen to be large enough so that the grid spacing accurately reproduces your charge distribution. However, it should not be so large that it will slow down your simulation to provide unnecessary precision. Typically, a grid density of slightly more than 1/Å is a good choice to reproduce charge distribution in biological systems, where the closest atoms have a bond separation on the order of 1 Å. This corresponds to a `PMEGridSpacing` of 1.0. NAMD will then automatically set the PME grid sizes such that there is always less than 1.0 Å between grid points and the sizes contain only small prime factors (e.g. 2, 3, and 5) for computational efficiency.

  Note also that when using the PME method, the command `cutoff`

(see above) dictates the separation between long and short range forces for the method; it does not simply turn off electrostatic interactions.

- Constant Temperature Control
  - `langevin`: indicates whether or not the simulation uses Langevin dynamics; uses values `on` and `off`. See the science box below for more on Langevin dynamics.
  - `langevinDamping`: sets the value of the Langevin coupling coefficient, which quantifies the friction applied to the system, removing energy from the system, slowing atoms down, etc. It is specified in ps^{-1}.
  - `langevinTemp`: Langevin dynamics may be applied to all atoms or only non-hydrogen atoms in the system. This command specifies the temperature at which to keep those atoms, even though friction and random forces will be acting on them (remember `$temperature` is a variable for the value `310`).
  - `langevinHydrogen`: indicates whether or not Langevin dynamics will be applied to hydrogen atoms in the simulation; uses values `on` and `off`.



Langevin Dynamics. Langevin dynamics is a means of controlling the kinetic energy of the system, and thus, controlling the system temperature and/or pressure. The method uses the Langevin equation for a single particle:

$$m_i \frac{d^2\mathbf{x}_i(t)}{dt^2} = \mathbf{F}_i\{\mathbf{x}_i(t)\} - \gamma_i \frac{d\mathbf{x}_i(t)}{dt} m_i + \mathbf{R}_i(t)$$

Here, two additional terms on the right hand side accompany the ordinary force the particle experiences. The second term represents a frictional damping that is applied to the particle with frictional coefficient $\gamma_i m_i$. The third term represents random forces which act on the particle (as a result of solvent interaction). These two terms are used to maintain particle kinetic energy to keep system temperature, for instance, at a constant average value. Because an unnecessarily high damping constant can significantly slow the system's dynamics, one should always find the minimum `langevinDamping` coefficient sufficient to maintain the temperature. **A value of 1.0 is often a good starting point.**

8 Continuation of "Simulation Parameters":

- Constant Pressure Control (variable volume)
  - `useGroupPressure`: NAMD calculates system pressure based on the forces between atoms and their kinetic energies. This command specifies whether interactions involving hydrogen atoms should be counted for all hydrogens or simply between groups of hydrogen atoms; uses values `yes` and `no` and must be set to `yes` if `rigidBonds` are set.

  - `useFlexibleCell`: specifies whether or not you want to allow the three dimensions of the periodic cell to vary *independently*; uses values `yes` or `no`.
  - `useConstantArea`: NAMD allows you to keep the $x - y$ cross sectional area constant while varying the z dimension; uses values `yes` and `no` (useful for membrane protein simulations).
- The following are specifications for the Langevin piston.
  - `langevinPiston`: indicates whether or not the simulation uses a Langevin piston to control the system pressure; uses values `on` and `off`.
  - `langevinPistonTarget`: specifies, in units of bar, the pressure which the Langevin piston tries to maintain (1 atm = 1.013 bar).
  - `langevinPistonPeriod`: sets the oscillation time constant in fs for the Langevin piston.
  - `langevinPistonDecay`: sets the damping time constant in fs for the Langevin piston.
  - `langevinPistonTemp`: sets the "noise" temperature in K for the Langevin piston; should be set equal to the target temperature for the temperature control method (here, set by `$temperature`).
- Output Frecuencies
  - `restartfreq`: During the simulation, NAMD can also create restart files, one of which is a pdb file which stores atomic coordinates, and the other of which stores atomic velocities. This command specifies the amount of time steps between writing to the restart file (here, every 5000 steps, equivalent to 10000fs or 10 ps). If this command is not set, NAMD will not create a restart file. Furthermore, NAMD will store the file from the previous cycle each time it writes a new file. The filename is appended with a `.old` extension; it is created in case NAMD fails in writing the new restart file.
  - `dcdfreq`: The dcd file contains only atomic coordinates, and they are written to the file several times over the course of a simulation. Thus, it provides a *trajectory* of the system over the runtime. This command specifies the number of time steps between writing new coordinates to the dcd output file. If this command is not set, NAMD will not create a dcd file.
  - `xstFreq`: The *extended system trajectory* file contains a record of the periodic cell parameters, essentially recording the trajectory of the cell boundaries over the run time. This command specifies how often, in time steps, the configuration will be recorded. If this command is set, three xst files will be output: 1 final and 2 restarts.
- In addition to the output files described, NAMD also prints a log of the simulation (which we will redirect into a `.log` file). This output

is explained further below. Here are some instructions that control the content of this file.

- `outputEnergies`: specifies the number of time steps between each output of system energies (for various force field interactions) into the `.log` file (here, every 100 steps, or 200 fs).
- `outputPressure`: specifies the number of time steps between each output of system pressure into the `.log` file.

9 The "Extra Parameters" section contains commands which are applicable to more specific simulations, not needed for our simple minimization and equilibration of Cdh23 EC1.

10 The "Execution Script" section contains three commands, the first two of which apply to minimization and the last one of which applies to equilibration.

- Minimization
  - `minimize`: sets the number of iterations over which to vary atom positions to search for a local minimum in the potential (in this case `1000`).
  - `reinitvels`: Minimization is performed on the system after all atomic velocities have been set to zero. This command resets atomic velocities such that the system starts at the temperature specified (in this case, `$temperature`, or 310K).
- Equilibration
  - `run`: sets the number of time steps over which to run the MD equilibration (in this case `50000`, which corresponds to 10,000 fs or 10 ps, since a 2 fs time step has been used).

11 Now, close the configuration file by clicking File → Exit.

Minimization and Equilibration. Typical MD minimization and equilibration simulations involve more than one minimization-equilibration cycle, often fixing and releasing different parts of molecules in the system. For instance, one typically minimizes the system and then equilibrates with the atoms in the protein fixed in space, and then minimizes the system again and equilibrates again, this time with the protein free to move. Fixing the protein allows the water, which typically responds much faster to forces than the protein, to do the relaxing in the first step. This saves computational effort and prevents the introduction of artifacts from an unstable starting structure.

NAMD User's Guide. The explanations of configuration file commands in the previous sections is far from complete. For a more in-depth explanation of configuration file commands and for acceptable values which may be supplied with those commands, see the NAMD User's Guide at `http://www.ks.uiuc.edu/Research/namd/current/ug/`.

Required Configuration File Parameters. The following parameters are required by NAMD to appear in every configuration file: `coordinates`, `structure`, `parameters`, `exclude`, `outputname`, `numsteps` and exactly one of the following: `temperature`, `velocities`, or `binvelocities`.

1.7 Run your Simulation

Running. Run your simulation by typing in a Terminal Window:

```
namd2 Cdh23EC1wbieq01.conf > Cdh23EC1wbieq01.log &
```

Setting the Path. You will either need to provide the full path to namd2 (e.g., `C:\Users\`*username*`\namd\namd2`) or add the namd2 directory to your path in order for Windows to locate it. This can be accomplished by right-clicking Computer on the Desktop and selecting Properties → Advanced system settings → Advanced → Environment Variables (the precise procedure may vary depending on your version of Windows). Under System variables, scroll down and select Path and then Edit. At the end of the long line in Variable Value, add a semicolon ; then the full path to the directory containing namd2 (but do NOT add the executable "namd2" at the end). Click OK. Now open a new command prompt. Regardless of the directory you are in, you should be able to type `namd2` and run it.

Your simulation may take several hours to complete, but in the meantime NAMD will be producing the output files described in Section 1.8. You can check these files while the simulation runs, and also look at an example in the `example-output` directory.

1.8 Exploring Output Files

1 The output of your minimization-equilibration simulation will be returned to the `Cdh23EC1` directory. Go there using the graphical interface and check the files that are being generated.

Based on the descriptions in the Output part of the "Simulation Parameters" section, your present simulation should yield these eleven output files:

- `Cdh23EC1wbieq01.log`
- `Cdh23EC1wbieq01.coor`
- `Cdh23EC1wbieq01.vel`

- `Cdh23EC1wbieq01.xsc`
- `Cdh23EC1wbieq01.dcd`
- `Cdh23EC1wbieq01.restart.coor`
- `Cdh23EC1wbieq01.restart.vel`
- `Cdh23EC1wbieq01.restart.xsc`
- `Cdh23EC1wbieq01.coor.old`
- `Cdh23EC1wbieq01.vel.old`
- `Cdh23EC1wbieq01.xsc.old`

Your original configuration file should also still be in the directory.

Seven of the output files contain binary data so that coordinate and velocity information may be kept to high precision. The files which are not in binary code are:

* `Cdh23EC1wbieq01.xsc`, `Cdh23EC1wbieq01.restart.xsc`, and `Cdh23EC1wbieq01.restart.xsc.old`, which are *extended system configuration* files. They store the periodic cell dimensions of the system and the time step at which this information was taken. They are not useful when periodic boundary conditions are not used.

* `Cdh23EC1wbieq01.log`, the log file. It is here that the energy of the interactions among atoms, governed by the force field, is stored. The log file stores a lot of information, let's explore it!

2 The first section of the log file, indicated by the `Info` tag, contains the parameters used to run the simulation, and other useful information about the system, such as the number of atoms, the type and number of bonds, the total mass, and the total charge.

3 After the first section, information about the minimization of the system is displayed beginning with the line labeled `TCL`. This line states how long the minimization lasts.

4 After that, the pressure of the system is given.

5 The next part gives a listing of energies of interaction, first defining which are listed with `ETITLE`, then listing the actual values in kcal/mol for every iteration (next to `ENERGY`). The time step (TS) value is a count of the number of iterations NAMD has performed and is *not* in units of time, while the temperature (TEMP) value is in K.

```
TCL: Minimizing for 1000 steps
PRESSURE: 0 1.56281e+07 1.88446e+07 -2.53392e+07 1.88446e+07 5.52088e+07 -4.07396e+07 -2.53392e+07 -4.07396e+07 4.77935e+07
GPRESSURE: 0 1.5634e+07 1.88438e+07 -2.534e+07 1.88453e+07 5.52135e+07 -4.07393e+07 -2.53396e+07 -4.07391e+07 4.77969e+07
ETITLE:      TS           BOND          ANGLE          DIHED          IMPRP               ELECT          VDW       BOUNDARY          MISC
     KINETIC          TOTAL           TEMP      POTENTIAL         TOTAL3        TEMPAVG      PRESSURE     GPRESSURE        VOLUME
       PRESSAVG      GPRESSAVG

ENERGY:       0      5374.3972      1473.3166       963.0433         5.6441        -58251.1521  28440502.2967       0.0000       0.0000
       0.0000     28390067.5458         0.0000  28390067.5458  28390067.5458         0.0000      39543481.6253  39548158.0082    199922.9960
 39543481.6253  39548158.0082


MINIMIZER RESTARTING CONJUGATE GRADIENT ALGORITHM DUE TO POOR PROGRESS
LINE MINIMIZER REDUCING GRADIENT FROM 1.6718e+07 TO 16718
PRESSURE: 10 375.227 54.0611 -144.61 54.0611 117.99 -461.79 -144.61 -461.79 -811.285
GPRESSURE: 10 2022.6 215.968 -14.4594 66.2977 2011.07 -220.878 59.7151 -350.176 991.102
ETITLE:      TS           BOND          ANGLE          DIHED          IMPRP               ELECT          VDW       BOUNDARY          MISC
     KINETIC          TOTAL           TEMP      POTENTIAL         TOTAL3        TEMPAVG      PRESSURE     GPRESSURE        VOLUME
       PRESSAVG      GPRESSAVG

ENERGY:      10      2401.3788      1279.2637       957.2361         4.7007        -59275.6996   6455.0804       0.0000       0.0000
       0.0000       -48178.0399         0.0000    -48178.0399    -48178.0399         0.0000       -106.0226      1674.9234    199922.9960
    -106.0226      1674.9234
```

Figure 9: Typical minimization output in the log file.

6 There are several parameters listed that correspond to the numerical search for a minimum in the potential. `GRADIENT` should decrease over the course of the minimization and may be used to roughly judge the extent of minimization. See Fig. 9.

7 After minimization, equilibration data are presented in the log file. See Fig. 10. The first lines shown in the figure confirm that NAMD is writing the proper information at the proper time step, which you have specified in the configuration file. The following space is where pressures will be written. Energies at that time step are then fully reported. In this case, they are output every 100 time steps, as specified in the configuration file.

```
REINITIALIZING VELOCITIES AT STEP 1000 TO 310 KELVIN.
TCL: Running for 250000 steps
PRESSURE: 1000 -917.455 -68.4486 51.4632 -68.4486 -1055.23 81.0371 51.4632 81.0371 -969.715
GPRESSURE: 1000 -870.624 -58.5767 63.7563 -67.6355 -1074.72 67.974 54.0155 61.9085 -991.12
ETITLE:      TS           BOND          ANGLE          DIHED          IMPRP               ELECT
             VDW       BOUNDARY          MISC       KINETIC           TOTAL           TEMP
       POTENTIAL         TOTAL3       TEMPAVG      PRESSURE     GPRESSURE        VOLUME
         PRESSAVG      GPRESSAVG

ENERGY:     1000       105.8088       277.9583       912.2703        11.0449        -80282.2922
       9273.5280         0.0000         0.0000     11573.7879       -58127.8940       309.2548
    -69701.6820    -58135.8099       309.2548          -980.7992      -978.8223    199922.9960
       -980.7992      -978.8223


PRESSURE: 1500 301.443 52.2335 67.1192 52.2335 -35.997 -70.6228 67.1192 -70.6228 515.665
GPRESSURE: 1500 255.71 43.1202 6.77833 44.8786 -50.6454 -79.6304 139.641 30.8404 570.494
PRESSAVG: 1500 -53.4251 -38.0654 55.1454 -38.0654 9.76447 -103.44 55.1454 -103.44 41.628
GPRESSAVG: 1500 -52.9006 -41.1915 55.2651 -41.3719 9.16481 -108.413 60.5953 -102.682 42.009
Info: Benchmark time: 8 CPUs 0.0404139 s/step 0.233877 days/ns 626.055 MB memory
ENERGY:     1500       228.5763       621.8203       984.7185        34.4298        -74531.3095
       7678.2785         0.0000         0.0000      7174.4068       -57809.0793       191.7021
    -64983.4861    -57787.6439       188.0709           260.3703      258.5196    177473.2104
         -0.6775        -0.5756
```

Figure 10: Typical equilibration output of the log file.

8 The end of the file confirms that the final frame of the simulation is being recorded, and provides computational data which may be used to bench-

mark your run should you wish to change parameters, run on another computer, etc.

1.9 Analysis of Equilibration

In this section, you will analyze the extent to which your system has equilibrated using what is known as the *Root Mean Square Deviation*, or RMSD. The RMSD characterizes the amount by which a given selection of your molecule deviates from a defined position in space. You will use the output files from your minimization and equilibration of Cdh23 EC1 in a water box to calculate RMSD values and analyze the extent of equilibration of the simulation. RMSD values will be calculated for all atoms of the protein backbone (without hydrogens) for the entire protein.

1.9.1 Checking Thermodynamic Quantities

Your minimization-equilibration simulation generated a trajectory for the system called `Cdh23EC1wbieq01.dcd`. That trajectory is key to calculating the RMSD of the protein over the course of the equilibration. The simulation was performed in a so-called *NPT ensemble* in which the number of particles, N, the pressure, P, and the temperature, T, are kept constant.

VMD includes many analysis tools. One of these, the NAMD Plot extension, may be used to plot output from NAMD log files. This VMD plugin uses the program `namdplot`, which gets the relevant data from the log files for values of energies, temperature, etc. over time. VMD then uses an internal plotting program to plot the data for you.

namdplot. The program `namdplot` is also available separately at www.ks.uiuc.edu/Research/namd/utilities/. This stand-alone version will retrieve the relevant data and then use the package `xmgrace` for plotting.

1 Open VMD by typing `vmd` in a Terminal window.

Launch VMD. Open VMD graphically by double-clicking on the VMD icon under `Applications` in the `Finder`.

Launch VMD. Open VMD by clicking Start → Programs → VMD.

2 Navigate to the `Cdh23EC1` directory using the `cd` command in the VMD Tk Console

3 Launch the NAMD Plot extension by clicking Extensions → Analysis → NAMD Plot in the VMD Main window.

4 In the NAMD Plot window, select File → Select NAMD Log File, pick the file `Cdh23EC1wbieq01.log`, and click Open.

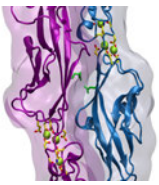
5 All the thermodynamic variables from the NAMD log file are available for plotting as shown in the NAMD Plot window. TS stands for time step, BOND for bond energy, and so on. Click the box marked TEMP and then select File → Plot Selected Data. This will get all the temperature data from the log file and plot it over time.

6 Try using NAMD Plot for `TOTAL` (total energy). You should see the variable converging to an average value for an equilibrated system.

7 When you are finished examining the plots, close the Multiplot window.

The thermodynamic variables that you checked above tell you about the thermodynamic state of the whole system. But you would also like to know if your protein is conformationally stable. For this, you will calculate the RMSD of the protein backbone. This will give you an idea of the stability of the protein. If the RMSD is still increasing at the end of the run, it means your protein is still searching for a lower energy state, and thus is not yet equilibrated!

1.9.2 RMSD



Root Mean Square Deviation. The Root Mean Squared Deviation (RMSD) is a numerical measure of the difference between two structures. It is defined as:

$$RMSD_\alpha(t_j) = \sqrt{\frac{\sum_{\alpha=1}^{N_\alpha} (\vec{r}_\alpha(t_j) - \langle \vec{r}_\alpha \rangle)^2}{N_\alpha}}$$

$$\langle \vec{r}_\alpha \rangle = \frac{1}{N_t} \sum_{j=1}^{N_t} \vec{r}_\alpha(t_j)$$

where N_α is the number of atoms whose positions are being compared, N_t is the number of time steps over which atomic positions are being compared, $\vec{r}_\alpha(t_j)$ is the position of atom α at time t_j, and $\langle \vec{r}_\alpha \rangle$ is the average value of the position of atom α to which the positions $\vec{r}_\alpha(t_j)$ are being compared. Note that the RMSD for your protein will change over time. Also, note that the script you are using takes the first time step of the simulation as a reference, rather than using an average value.

You will use VMD to calculate RMSD values:

1 Load your psf file for the Cdh23 EC1 system, `Cdh23EC1wbi.psf`, which is in your `Cdh23EC1` directory, by clicking File → New Molecule... in the

VMD Main window. Click the Browse button in the Molecule File Browser window. Find `Cdh23EC1wbi.psf` and double click on it. Then click Load.

2 Click the Browse button in the Molecule File Browser window. Browse for `Cdh23EC1wbieq01.dcd` and select it by double clicking on it. Then click Load. You should see water molecules moving around. You have loaded the trajectory of your equilibration.

3 The script we are going to use is called `rmsd.tcl`. This is the script content:

```
set outfile [open rmsd.dat w]
set nf [molinfo top get numframes]
set frame0 [atomselect top "protein and backbone and noh" frame 0]
set sel [atomselect top "protein and backbone and noh"]
# rmsd calculation loop
for { set i 1 } { $i ≤ $nf } { incr i } {
 $sel frame $i
 $sel move [measure fit $sel $frame0]
 puts $outfile "[measure rmsd $sel $frame0]"
}
close $outfile
```

4 The script does the following:

- Opens the file `rmsd.dat` for writing.
  `set outfile [open rmsd.dat w]`
- Gets the number of frames in the trajectory and assigns this value to the variable `nf`
  `set nf [molinfo top get numframes]`
- Selects the first frame of the molecule to be the one other frames will compare to. The selection contains the atoms in the backbone of the protein, excluding hydrogens:
  `set frame0 [atomselect top "protein and backbone and noh" frame 0]`
- Makes the same selection as before for an undetermined frame.
  `set sel [atomselect top "protein and backbone and noh"]`
- Text after `#` denotes comment.
  `# rmsd calculation loop`
- Loops over all frames in the trajectory:
  `for { set i 1 } { $i ≤ $nf } { incr i } {`
- Changes the selection `$sel` to the frame to be compared.
  `$sel frame $i`

- Calculates the matrix that will fit both selections. Apply this matrix to the second selection to align the molecules:
  `$sel move [measure fit $sel $frame0]`
- Calculates the RMSD value between these two selections, and write it to file:
  `puts $outfile "[measure rmsd $sel $frame0]"`

You can use the script for the system to test for equilibration.

5 Open the Tk Console window of VMD by clicking Extensions → Tk Console in the VMD Main window. In the Tk Console window, type `source rmsd.tcl`. This will perform all the commands in the script. The script will write a file (`rmsd.dat`) that will contain the value of the RMSD of the protein backbone against time.

Outside of VMD, you can use a plotting program to see this data. Examples of these are gnuplot, xmgrace, Microsoft Excel, Mathematica, and scilab.

6 Use xmgrace to plot the file `rmsd.dat`. In the Terminal window, type `xmgrace rmsd.dat`.

Plotting. We will use Microsoft Excel to plot the values in the file `rmsd.dat`.

7 Open Excel by clicking Start → Programs and finding your version of Excel.

8 In Excel, click File → Open..., navigate to the directory `Cdh23EC1`, and choose the file `rmsd.dat`. Make sure the drop-down menu Files of type is set to All Files.

9 If the Text Import Wizard appears, choose Delimited for your file type and click Finish. The RMSD values will appear in Column A

10 Make a plot of the data by clicking Insert → Chart..., choosing XY (Scatter) as your Chart Type, selecting the lower-right Chart sub-type, and following appropriate steps.

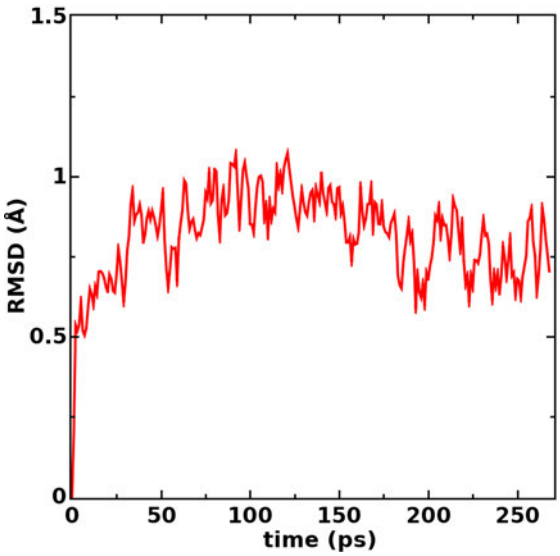


Figure 11: RMSD versus simulation time for a system equilibrating for several picoseconds.

The curve does not reveal very much information because of the short runtime and infrequent writing of the dcd file. In a longer MD simulation you would see the RMSD curve flattening. This means the system is equilibrated.

Congratulations! This ends the "Hands-on Introduction to Protein Simulations" tutorial. We hope that you learned a lot with it, and that you will make a great use of all the capabilities NAMD has to offer. Please visit
https://research.chemistry.ohio-state.edu/sotomayor/teaching/
for updated versions of this and other tutorials.

A PDB Files

The term PDB can refer to the Protein Data Bank (http://www.rcsb.org/pdb/), to a data file provided there, or to any file following the PDB format. Files in the PDB include information such as the name of the compound, the species and tissue from which is was obtained, authorship, revision history, journal citation, references, amino acid sequence, stoichiometry, secondary structure locations, crystal lattice and symmetry group, and finally the ATOM and HETATM records containing the coordinates of the protein and any waters, ions, or other heterogeneous atoms in the crystal. Some PDB files include multiple sets of coordinates for some or all atoms. Due to the limits of x-ray crystallography and NMR structure analysis, the coordinates of hydrogen atoms are not included in the PDB.

NAMD and VMD ignore everything in a PDB file except for the ATOM and HETATM records, and when writing PDB files the ATOM record type is used for all atoms in the system, including solvent and ions. Here are the ATOM records for the first two residues of ubiquitin from the 1UBQ entry in the PDB:

```
ATOM      1  N   MET     1      27.340  24.430   2.614  1.00  9.67      1UBQ  71
ATOM      2  CA  MET     1      26.266  25.413   2.842  1.00 10.38      1UBQ  72
ATOM      3  C   MET     1      26.913  26.639   3.531  1.00  9.62      1UBQ  73
ATOM      4  O   MET     1      27.886  26.463   4.263  1.00  9.62      1UBQ  74
ATOM      5  CB  MET     1      25.112  24.880   3.649  1.00 13.77      1UBQ  75
ATOM      6  CG  MET     1      25.353  24.860   5.134  1.00 16.29      1UBQ  76
ATOM      7  SD  MET     1      23.930  23.959   5.904  1.00 17.17      1UBQ  77
ATOM      8  CE  MET     1      24.447  23.984   7.620  1.00 16.11      1UBQ  78
ATOM      9  N   GLN     2      26.335  27.770   3.258  1.00  9.27      1UBQ  79
ATOM     10  CA  GLN     2      26.850  29.021   3.898  1.00  9.07      1UBQ  80
ATOM     11  C   GLN     2      26.100  29.253   5.202  1.00  8.72      1UBQ  81
ATOM     12  O   GLN     2      24.865  29.024   5.330  1.00  8.22      1UBQ  82
ATOM     13  CB  GLN     2      26.733  30.148   2.905  1.00 14.46      1UBQ  83
ATOM     14  CG  GLN     2      26.882  31.546   3.409  1.00 17.01      1UBQ  84
ATOM     15  CD  GLN     2      26.786  32.562   2.270  1.00 20.10      1UBQ  85
ATOM     16  OE1 GLN     2      27.783  33.160   1.870  1.00 21.89      1UBQ  86
ATOM     17  NE2 GLN     2      25.562  32.733   1.806  1.00 19.49      1UBQ  87
```

The fields seen here in order from left to right are the record type, atom ID, atom name, residue name, residue ID, x, y, and z coordinates, occupancy, temperature factor (called beta), segment name, and line number.

If this file is loaded into VMD and then written out as a new file, most of the extra information will be removed, the HETATM records will become ATOM records, and the previously empty chain ID field (between residue name and residue ID) will be set to X (unless present in the original file), and the line number will be omitted, as seen here:

```
ATOM      1  N   MET X   1      27.340  24.430   2.614  1.00  9.67      1UBQ
ATOM      2  CA  MET X   1      26.266  25.413   2.842  1.00 10.38      1UBQ
ATOM      3  C   MET X   1      26.913  26.639   3.531  1.00  9.62      1UBQ
```

B PSF Files

A PSF file, also called a protein structure file, contains all of the molecule-specific information needed to apply a particular force field to a molecular system. The CHARMM force field is divided into a topology file, which is needed to generate the PSF file, and a parameter file, which supplies specific numerical values for the generic CHARMM potential function. The topology file defines the atom types used in the force field; the atom names, types, bonds, and partial charges of each residue type; and any patches necessary to link or otherwise mutate these basic residues. The parameter file provides a mapping between bonded and nonbonded interactions involving the various combinations of atom types found in the topology file and specific spring constants and similar parameters for all of the bond, angle, dihedral, improper, and van der Waals terms in the CHARMM potential function.

The PSF file contains six main sections of interest: atoms, bonds, angles, dihedrals, impropers (dihedral force terms used to maintain planarity), and cross-terms. The following is taken from a PSF file for ubiquitin. First is the title and atom records:

```
PSF CMAP

       6 !NTITLE
 REMARKS original generated structure x-plor psf file
 REMARKS 2 patches were applied to the molecule.
 REMARKS topology top_all27_prot_lipid.inp
 REMARKS segment U { first NTER; last CTER; auto angles dihedrals }
 REMARKS defaultpatch NTER U:1
 REMARKS defaultpatch CTER U:76

    1231 !NATOM
       1 U    1    MET  N    NH3   -0.300000       14.0070           0
       2 U    1    MET  HT1  HC     0.330000        1.0080           0
       3 U    1    MET  HT2  HC     0.330000        1.0080           0
       4 U    1    MET  HT3  HC     0.330000        1.0080           0
       5 U    1    MET  CA   CT1    0.210000       12.0110           0
       6 U    1    MET  HA   HB     0.100000        1.0080           0
       7 U    1    MET  CB   CT2   -0.180000       12.0110           0
```

The fields in the atom section are atom ID, segment name, residue ID, residue name, atom name, atom type, charge, mass, and an unused 0. PSF files may be in either CHARMM or X-PLOR format, with the CHARMM format using an integer rather than a name for the atom type. NAMD requires the X-PLOR format, which is also more flexible since it is not tied to the specific order of atom types in a single parameter file. NAMD and VMD require that the order of atoms in any PDB, DCD, or other atomic data file exactly matches the order found in the PSF file.

Notice that hydrogen atoms are included, that multiple atoms in a residue may share the same type (e.g., HT1, HT2, and HT3 are of type HC), and that

atoms of the same element may have different types (e.g., CA and CB, HA, and HT1). Much of the information in the PDB is also included in the PSF file. Often, the only information from the PDB file used by NAMD is the atomic coordinates. In some cases the contents of the occupancy and beta fields of the PDB file are used as parameters to optional simulation methods.

The covalent bond section lists four pairs of atoms per line:

```
 1237 !NBOND: bonds
    1       5       2       1       3       1       4       1
    5       6       7       5       7       8       7       9
   10       7      10      11      10      12      13      10
```

The angle section lists three triples of atoms per line:

```
 2257 !NTHETA: angles
    1       5       6       1       5      18       2       1       5
    2       1       4       2       1       3       3       1       5
    3       1       4       4       1       5       5      18      19
```

The dihedral and improper sections list two quadruples of atoms per line:

```
 3293 !NPHI: dihedrals
    1       5       7      10       1       5       7       8
    1       5       7       9       1       5      18      20
    1       5      18      19       2       1       5       7

  204 !NIMPHI: impropers
   18       5      20      19      20      18      22      21
   30      32      27      31      30      27      32      31
   32      30      33      34      32      30      34      33
```

The cross-term sections list two quadruples of atoms per line:

```
74 !NCRTERM: cross-terms
     18      20      22      35      20      22      35      37
     35      37      39      54      37      39      54      56
     54      56      58      74      56      58      74      76
```

The order of the atoms within each bond is not significant. The order of atoms within angles, dihedrals, and impropers is significant, but the order can be reversed without affecting the resulting potential. In no case is the relative order of different bonds etc. significant. Since every bond and improper is explicitly listed in the topology file, the atom order within these terms tend to match the original order given. Angles and dihedrals are typically generated automatically and therefore appear in a sorted order.

There is a difference between X-PLOR formatted CHARMM PSF files, which VMD generates with the psfgen module and NAMD uses in conjunction with CHARMM parameter files, and PSF files generated by X-PLOR, which NAMD uses in conjunction with X-PLOR parameter files. Dihedral terms sometimes require multiple sinusoids to represent a torsional potential and therefore

multiple parameter sets appear in the parameter file. In PSF files generated by X-PLOR multiple dihedrals would be indicated by duplicate dihedrals in the PSF file. When using CHARMM PSF and parameter files NAMD extracts any multiple dihedral terms directly from the parameter file and each dihedral appears only once in the PSF file.