

# **NAMD TUTORIAL**

## **Windows Version**

## 1. Introduction

This tutorial provides a first introduction to NAMD and its basic capabilities. It can also be used as a refresher course for the non-expert NAMD user. The tutorial assumes that you already have a working knowledge of VMD and that NAMD 2.8 or later has been installed correctly on your computer. The tutorial covers the basic steps of a molecular dynamics simulation, i.e., *preparation, minimization, and equilibration* of your system typical simulation techniques and the analysis of equilibrium properties. The examples in the tutorial will focus on the study of ubiquitin – a small protein with interesting properties.

NAMD must be run from the MS-DOS command line in Microsoft Windows.

## 2. Required programs

The following programs are required for this tutorial:

- NAMD (Provided with this tutorial)
- VMD (Provided in the VMD Tutorial)
- Text Editor: We 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 and should not be used as a text editor.*

## 3. Getting Started

In order to begin the tutorial, you should open an MS-DOS command window by clicking **Start -> Programs -> Accessories -> Command Prompt**. This is the window that you will type commands into during the tutorial. This window will be referred to as a “Terminal” window. You can create another one of these windows if needed by following the same process. In order to run the tutorial, you will need the appropriate files. Download and extract the NAMD\_Tutorial.zip file in the Desktop of your computer. You should then use the Terminal window to navigate to the directory:

```
> cd C:\Users\<account-name>\Desktop\namd
```

## 4. Basics of NAMD

In this section you will learn how to use NAMD to set up basic molecular dynamics (MD) simulations. You will learn about typical NAMD input and output files, in particular, those for protein energy minimization and equilibration in water.

### 4.1 What is Needed

In order to run any MD simulation, NAMD requires at least four things:

- A *Protein Data Bank (pdb)* file which stores atomic coordinates and/or velocities for the system. You will use 1ubq.pdb for this tutorial.
- Protein Structure File (psf)* which stores structural information of the protein, such as various types of bonding interactions.
- A *force field parameter file*. A force field is a mathematical expression of the potential which atoms in the system experience. In this tutorial you will use CHARMM27 (*Chemistry at HARvard Macromolecular Mechanics*) force field.

- D. 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.

## 4.2 Steps of MD Simulation in NAMD:

Download and Extract the **namd\_tutorial.zip** file and extract in Desktop.

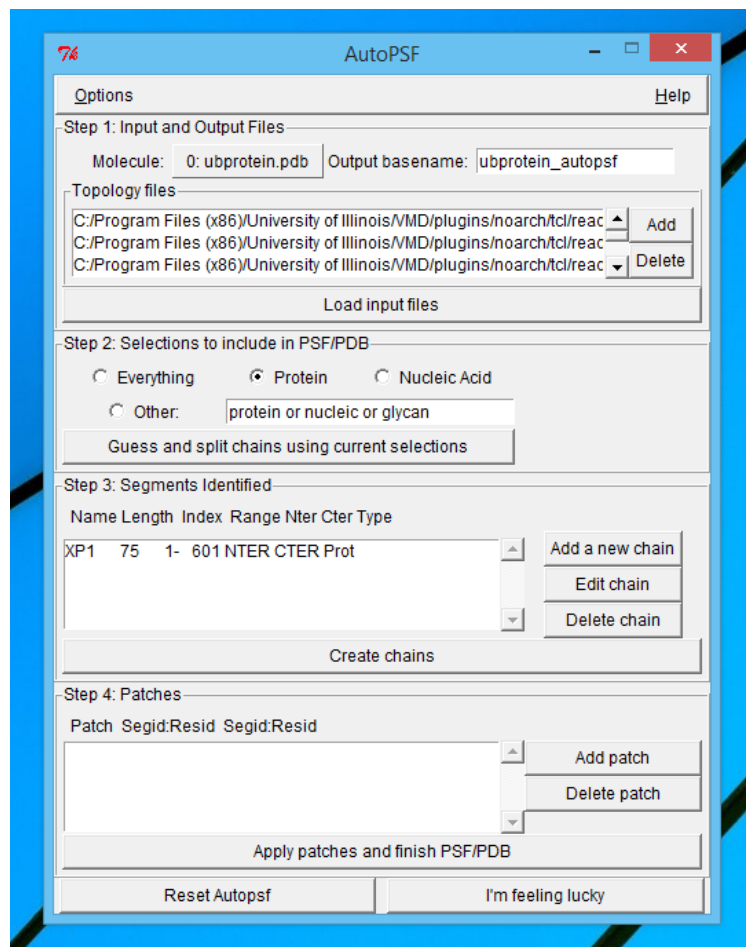
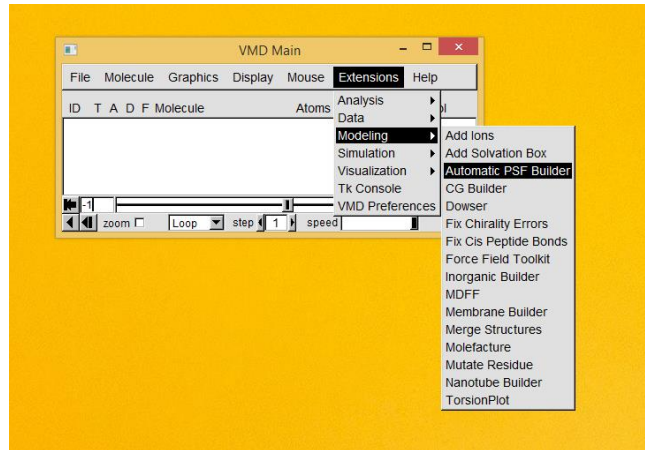
### A. PDB File Processing

- 1) First, you will remove the water molecules from **1ubq.pdb**, and create a pdb file of the protein alone.
- 2) Open VMD and load the **1ubq.pdb** file.
- 3) From **Graphics -> Representations** create a new representation for **'protein'** selection.
- 4) Select the molecule in VMD Main window. Go to **File -> Save Coordinates**.
- 5) In the Selected Atoms of the Save Trajectory window select **'protein'**
- 6) Click on **'Save'**
- 7) Save the file with the name **'ubprotein.pdb'** in the **namd** folder in your Desktop.

Note that the X-ray structure from the Protein Data Bank does not contain the hydrogen atoms of ubiquitin. This is because X-ray crystallography usually cannot resolve hydrogen atoms. The pdb file you will generate with psfgen along with the psf will contain guessed coordinates for hydrogen atoms of the structure. Later, energy minimization of the protein will ensure their positions are reasonable.

## B. Generating a Protein Structure File (PSF)

- 1) Now, you will create the psf file of ubiquitin. First load the **ubprotein.pdb** in the VMD Main. VMD offers an automatic psf file builder via the VMD Main menu by clicking **Extensions -> Modeling -> Automatic PSF Builder**.



2) Now in Automatic PSF Builder Click on **Load Input Files** and **Select the Protein** for PSF preparation. Click on **Guess and Split Chains using current selections**. Finally generate the psf file by clicking on **I am feeling lucky**.

3) This process will generate two files **ubprotein\_autopsf.psf** and **ubprotein\_autopsf.pdb** which can be found in **C:\-> Users -> <User Name>-> AppData -> Local -> VirtualStore -> Program Files (x86) -> University of Illinois -> VMD**

4) Copy these files to 'namd' directory in Desktop

So, you have created the psf! You may want to inspect **ubprotein\_autopsf.psf** and **ubprotein\_autopsf.pdb** using WordPad.

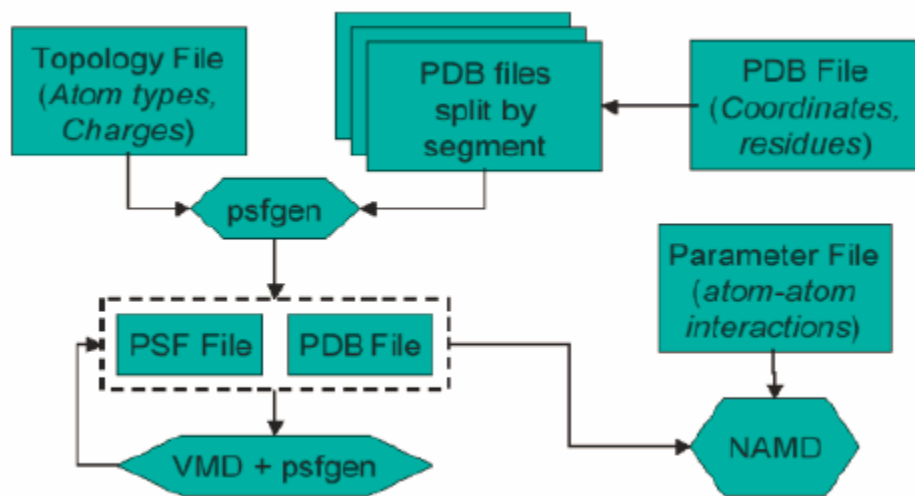
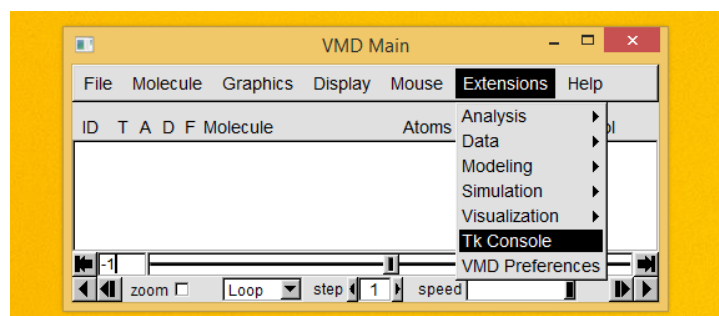


Fig1. Flowchart indicating the role of files as used by VMD, NAMD, and psfgen.

### C. Solvating the Protein

Now, the protein needs to be solvated, i.e., put inside water, to more closely resemble the cellular environment. You will do so in two ways, placing ubiquitin in a water box, in preparation for minimization and equilibration with periodic boundary conditions.

1) Form VMD Main window go to **Extension -> TkConsole** window.

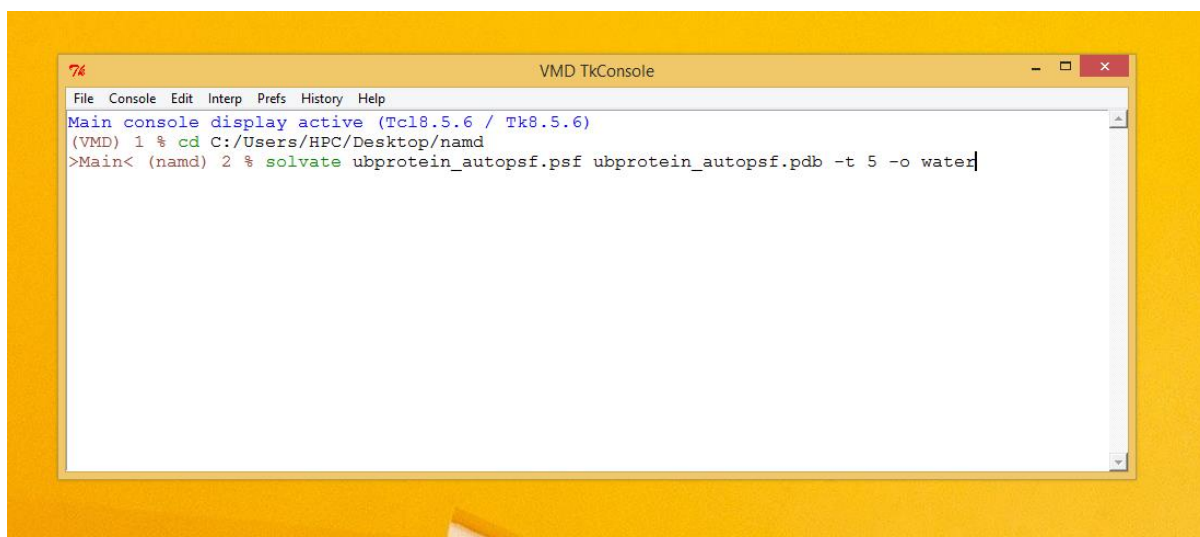


- 2) Change your Working directory to the **namd** directory in your Desktop using the following command.

```
% cd C:/Users/<user name>/Desktop/namd/
```

- 3) Now solvate the protein with the following command:

```
% solvate ubprotein_autopsf.psf ubprotein_autopsf.pdb -t 5 -o water
```



The **solvate** command loads the **solvate package**, so that VMD will be able to call it. The solvate package will put your protein (described in **ubprotein\_autopsf.psf** and **ubprotein\_autopsf.pdb**) in a box of water. The **-t** option creates the water box dimensions such that there is a layer of water **5Å** in each direction from the atom with the largest coordinate in that direction. The **-o** option creates the output files **water.pdb** and **water.psf** for ubiquitin with the water box.

Now, you can load the **water.psf** and **water.pdb** to visualize the water box surrounding the protein.

#### **D. Adding Ions to the system**

Ions should be placed in the water to represent a more typical biological 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. The ions present will shield the regions of the protein which carry the charge, and make the entire system more stable. They should be placed in regions of potential minima, since they will be forced to those regions during the simulation anyway. The psf file contains the charge of each atom and may be used to determine the charge of the total system or parts of it.

In this tutorial we will add *150mM of NaCl* to the system. In the VMD TkConsole Window enter the following command:

```
% autoionize -psf water.psf -pdb water.pdb -sc 0.15 -cation SOD -anion CLA -o ionized
```

This command will call **autoionize plugin**. It will neutralize the system first and add an additional 150mM NaCl. You will find two newly generated files i.e. **ionized.psf** and **ionized.pdb**

## E. Ubiquitin in a Water Box: Simulation with Periodic Boundary Conditions

Open the configuration file, **ub\_min\_eq.conf** in your **namd** directory. The configuration file may seem complex at first, but it will be examined line by line to determine its function in your simulation. Note that when “#” appears at the beginning of a line, the entire line is treated as a **comment** and ignored by NAMD.

A detailed information of Configuration File Formats are given in the [“Introduction to File Formats.pdf”](#) provided with this tutorial. Though some basic features are explained here.

**1.** 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 Equilibration of  
# Ubiquitin in a Water Sphere
```

**2.** The **“Adjustable Parameters”** section contains five commands:

- **structure**: calls the psf file describing the system (**ionized.psf**)
- **coordinates**: calls the initial coordinate data from the file listed next to it (**ionized.pdb**).
- **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.)
- **set outputname**: creates a variable called **“outputname”** in which to store a generic name for output files. If you place the text **“\$outputname”** in the configuration file, NAMD simply reads it as a label for **“ubqw\_min\_eq”**.
- **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.

**3.** The **“Simulation Parameter”** section contains many commands, commented into different categories:

- **Input**

- **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.
- **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.

- **Force-Field Parameters**

– **exclude**: specifies which atomic interactions are to be excluded from consideration. See Figure 2 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.

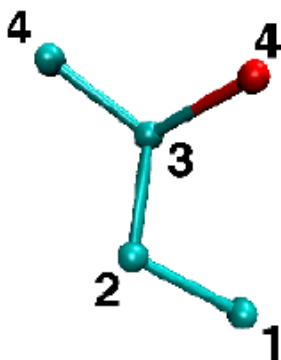


Fig. 2 Number labels for atoms which are a given amount of bonds away from one another

- **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.
- **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. When also employing a fast solver for electrostatics such as Particle Mesh Ewald or Multilevel Summation Method, the cutoff parameter instead defines the splitting distance for the  $1/r$  interaction potential, in which a short-ranged part is evaluated exactly between atoms within the cutoff distance, and the remaining long-ranged part is approximated by the solver.
- **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.
- **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 Figure 3.
- **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 Figure 3 .



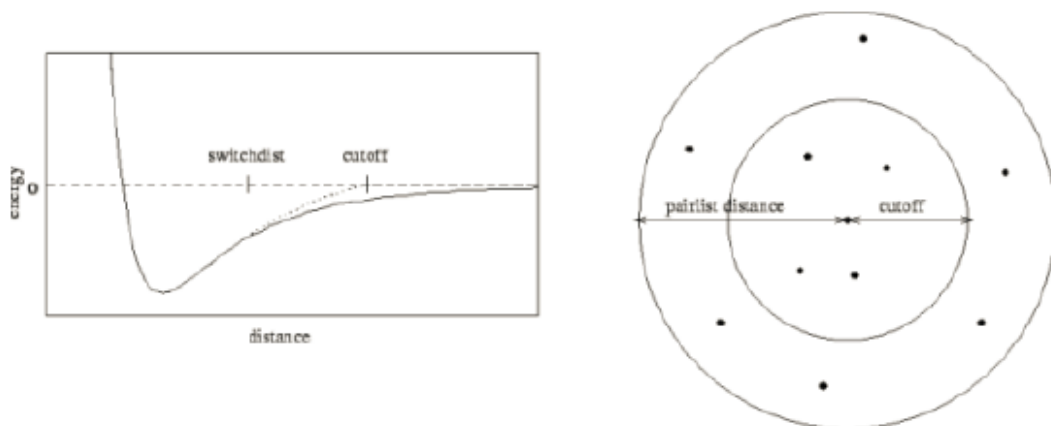


Fig. 3 Cutoff and switching distances indicated on the left. Pair list distance indicated on the right.

- **Integrator Parameters**

- **timestep**: indicates the value of the time step size used in the simulation. MD simulations solve *Newton's laws* 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.
- **stepspercycle**: Atoms are reassigned pair list identities (as explained above) once every cycle. This command specifies how long one cycle lasts, i.e. the number of time steps in one cycle.

- **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  $\text{ps}^{-1}$ .
  - **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.

- **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  $x = 42\text{\AA}$ ,  $y = 0\text{\AA}$ ,  $z = 0\text{\AA}$ . With each vector perpendicular, a rectangular 3-D box is formed.
- **cellOrigin**: specifies the coordinates of the center of the periodic cell in  $\text{\AA}$ . This is calculated after the solvation step!
- **wrapWater**: This command may be used with periodic boundary conditions. If a water molecule 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.
- **wrapAll**: same as **wrapWater**, except this applies to all molecules.

- **Electrostatics with PME**

*Particle Mesh Ewald (PME)* is a useful method for dealing with electrostatic interactions in a system when periodic boundary conditions are present. The Ewald sum is an efficient way of calculating long range forces in a 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  $1/\text{\AA}$  is a good choice to reproduce charge distribution in biological systems, where the closest atoms have a bond separation on the order of  $1\text{\AA}$ . This corresponds to a **PMEGridSpacing** of 1.0. NAMD will then automatically set the PME grid sizes (see below) such that there is always less than  $1.0\text{\AA}$  between grid points and the sizes contain only small prime factors (e.g. 2, 3, and 5). Alternatively, one can define the PME grid sizes manually, using **PMEGridSizeX**, **PMEGridSizeY**, and **PMEGridSizeZ**. These set the size of the PME grid along **cellBasisVector1**, 2 and 3, respectively (not the x, y, and z directions as implied). For speed in computing Fast Fourier Transforms, **PMEGridSizeX** should be chosen so that it can be factorized by 2, 3, or 5. If your **cellBasisVector1** = (60, 0, 0), a good choice for **PMEGridSizeX** might be 64, since  $60\text{\AA} / 64 = 0.9375\text{\AA}$  and  $64 = 2^6$ . Note that since **cellBasisVector** is defined with slightly different values in each direction, the size of the mesh spacing (in length) will be different in each direction. Note also that when using the PME method, the command cutoff dictates the separation between long and short range forces for the method; it does not simply turn off interactions.

- **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 should be counted for all hydrogen atoms 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.
  - **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)

The following are specifications for the Langevin piston which NAMD allows you to specify.

- **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 langevinTemp).

- **Output**

- **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 two files named **ubq\_min\_eq.coor** and **ubq\_min\_eq.vel**
- **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 500 steps, or 1000fs, or 1 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. 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.
- **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).
- **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.

– **outputPressure**: specifies the number of time steps between each output of system pressure into the .log file.

**4.** 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 100).

– **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).

- **Run**: sets the number of time steps over which to run the MD equilibration (in this case 2500, which corresponds to 5,000 fs or 5 ps, since a 2 fs time step has been used).

### **Running the Simulation:**

1) Now, in windows search type **cmd** for Command Prompt. By default it will open

**C:\Users\<user name>>**

Go to the ‘**namd**’ directory in the desktop.

**C:\Users\<user name>> cd Desktop\namd**

2) Now, Run the simulation with the ub\_min\_eq.conf file provided by the command below:

**C:\Users\<user name>\ Desktop\namd>NAMD\_2.12\namd2 ub\_min\_eq.conf**

You will see the process running in the **cmd** window. You can save the output of this process to a log file with the command below:

**C:\Users\<user name>\ Desktop\namd>NAMD\_2.12\namd2 ub\_min\_eq.conf > ub\_min\_eq.log**

### 4.3 Analysis of the Simulation:

So, we have performed the energy minimization and equilibration of the ubiquitin in water. Load the **water.psf** and the **ubq\_min\_eq.dcd** in VMD to visualize the movements of the protein throughout the process. Now we will use the VMD **RMSD Trajectory tool** and **HBonds Plugin** (from **VMD Main -> Extension -> Analysis**) for some basic analysis.

#### A. RMSD TT: RMSD Trajectory Tool

RMSD measures the deviation of a target set of coordinates (i.e. a structure) to a reference set of coordinates, with RMSD=0.0 indicating a perfect overlap. RMSD is defined as:

$$RMSD = \sqrt{\frac{\sum_{i=0}^N [m_i * (X_i - Y_i)^2]}{M}}$$

Where N is the number of atoms,  $m_i$  is the mass of atom i,  $X_i$  is the coordinate vector for target atom i,  $Y_i$  is the coordinate vector for reference atom i, and M is the total mass. If the RMSD is not mass-weighted, all  $m_i = 1$  and  $M = N$ .

The basic features of the tool are:

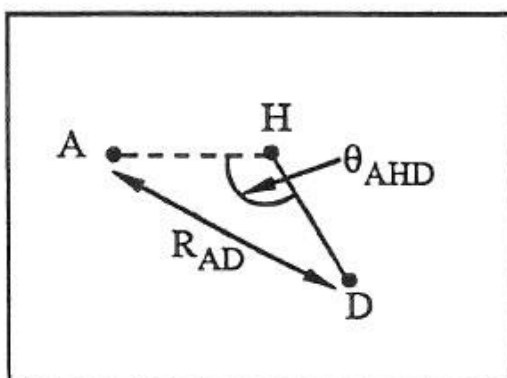
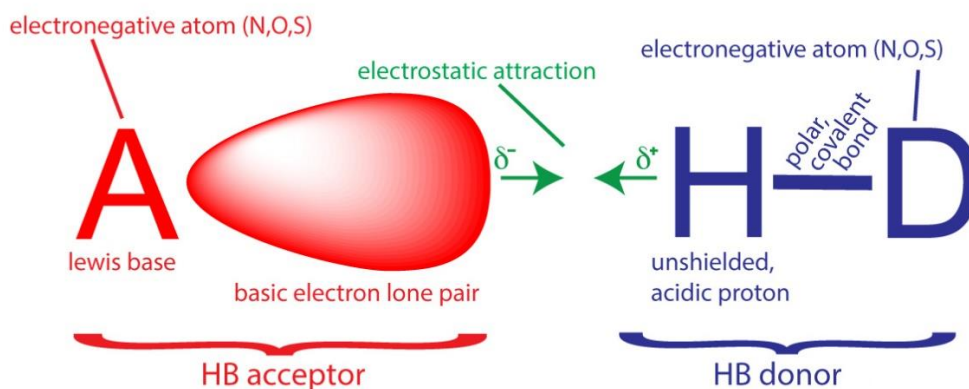
- Calculate rmsd in trajectories, skip frames and do an all-to-all rmsd (all molecules/frames against each other).
- Calculate weighted rmsd. Align frames in trajectories. Iterative fitting.
- Plot results via the Multiplot VMD plugin (in all systems), Xmgrace (on unix systems) or MS Excel (on windows). Save results to file.
- Swap equivalent atoms (i.e. Glu OE1 and OE2) to reduce rmsd (if Swap plugin is available).
- Generate basic statistics (average, standard deviation, minima and maxima).
- Multi-line atoms selection with support for comments.
- Highlight equivalent atoms of the rmsd selection.

#### B. HBonds Plugin

The Hbonds plugin counts the number of hydrogen bonds formed throughout a trajectory. The search can be restricted to a single selection or between two distinct selections, as well as a frame range given by the user.

*Criteria for the formation of a hydrogen bond:* A hydrogen bond is formed between an atom with a hydrogen bonded to it (the donor, D) and another atom (the acceptor, A) provided that the distance D-A is less than the cut-off distance (default 3.0 Angstroms) and the angle D-H-A is less than the cut-off angle (default 20 degrees).

## Anatomy of a Hydrogen Bond



The plugin provides options for up to two selections (which should not overlap) as well as the frames over which to calculate. The selections can be updated every frame, but at the cost of speed; details about selections that may require this option can be found on the Salt Bridges plugin page. The number of hydrogen bonds vs. time can be plotted immediately within VMD, and/or saved to a file (default hbonds.dat). Additionally, all messages can be output to a log file. Further options are provided to control the atoms used in the selections (only polar atoms vs. all atoms as normally done in VMD), whether to limit the first selection to be the donor, the acceptor, or both (default: both), and whether to calculate detailed information about the hydrogen bonds. The detailed output includes all hydrogen bonds formed in the trajectory (according to some basic criteria) and their frequency. Note that when using the option "all" for detailed output, the frequency of interaction may be greater than 100%, because a given residue pair may contain more than one hydrogen bond, each of which is counted separately.