All output files for the minimization and equilibration of your ubiquitin in a water sphere system will be placed in this directory. The configuration file is the only input file placed here, since it is particular to this simulation. The pdb, psf, and parameter files, which may be used by many other simulations, are placed in your `common` directory and are *called* by each respective configuration file.

2 Open the configuration file, `ubq_ws_eq.conf` by typing `nedit ubq_ws_eq.conf`.

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. In the middle of a line, ";#" is used to comment out the remainder of the line.

3 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
```

4 The "Adjustable Parameters" section contains five commands:

- `structure`: calls the psf file describing the system (`../common/ubq_ws.psf`).
- `coordinates`: calls the initial coordinate data from the file listed next to it (in this case `../common/ubq_ws.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 "`ubq_ws_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.

5 The "Simulation Parameters" 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 (in this case `../common/par_all27_prot_lipid.inp`).
  - `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 4 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.

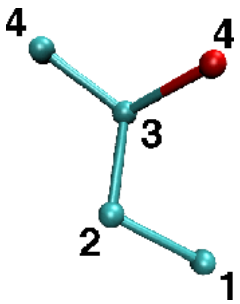


Figure 4: Number labels for atoms which are a given amount of bonds away from one another

  - `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 (Section 1.5.1) or Multilevel Summation Method (Section 1.4.1), 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 5.
- `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 5.

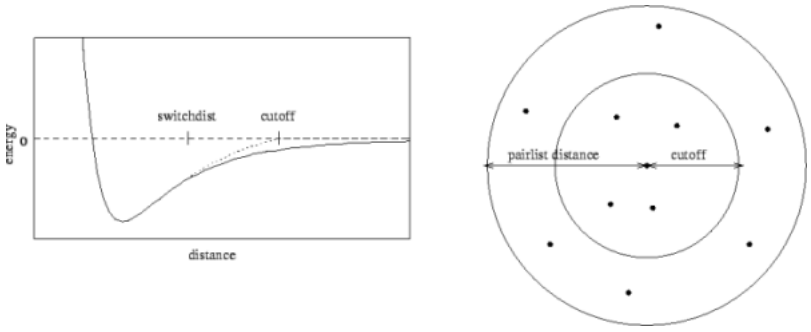


Figure 5: Cutoff and switching distances indicated on the left. Pair list distance indicated on the right.

- 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 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`.

- Electrostatics with MSM

  Multilevel Summation Method (MSM) is a useful method for dealing with electrostatic interactions in a system that does not have fully periodic boundary conditions along all three dimensions, in other words, when employing non-periodic or semi-periodic boundaries. Although MSM also supports fully periodic boundary conditions, the Particle Mesh Ewald (PME) method discussed in Section 1.5 is more efficient for smaller periodic systems, such as ubiquitin simulated in a water box. In the present simulation involving a water sphere, MSM is the best alternative for electrostatics.

  MSM uses particle meshes, similar to PME, arranged as 3-D grids for evaluating long-range electrostatic interactions. Charges are assigned to nearby grid points, and the interaction of these gridded charges result in gridded potentials that are interpolated back to the atoms to determine the long-range atomic forces. Unlike PME which uses a single grid resolution, MSM uses a nesting of grids having coarser resolution at each successive "level" in real space for resolving the long-ranged tail of the $1/r$ Coulomb potential. Rather than evaluating the Ewald sum as PME does, MSM uses a continued splitting

of the interaction potential and performs interpolation from these nested grid levels.

  - `MSM`: indicates whether or not the simulation uses the Multilevel Summation Method; uses values `yes` and `no`.
  - `MSMGridSpacing`: The grid spacing between the finest level grid points determines, in part, the accuracy and efficiency of MSM. According to an error versus cost analysis, the optimal value is for the grid spacing to be a bit larger than the average interatomic spacing; since the performance is extremely sensitive to the grid spacing, it is best to leave this parameter set to its default value of 2.5Å.
  - `MSMxmin`, `MSMxmax`, `MSMymin`, `MSMymax`, `MSMzmin`, `MSMzmax`: For non-periodic simulation, MSM needs to know the expected maximum and minimum coordinates so as to establish grids that will contain all atoms throughout the entire simulation. To keep the atoms from leaving the simulated space, it might be necessary to use a restraining potential, such as the spherical boundary restraints used for the water sphere simulation.

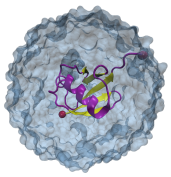
Additional MSM parameters that further control accuracy and performance are documented in the NAMD User's Guide.

- 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_ws_eq.coor` and `ubq_ws_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).

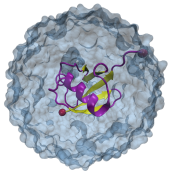


Langevin Dynamics. Langevin dynamics is a means of controlling the kinetic energy of the system, and thus, controlling the system temperature and/or pressure. The method uses the Langevin equation for a single particle:

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

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 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.**

Rigid Bonds. The time step used in any MD simulation should be dictated by the fastest process (i.e. movement of 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 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. For simulations with time step of 1 fs, one should use rigidBonds for water because water molecules have been parametrized as rigid molecules.



Electrostatics. Long-range electrostatic interactions play a crucial role in many biomolecular processes, making it important to accurately represent them in simulations. However, direct calculation of the full electrostatic interaction potential between all pairs of atoms is not feasible when simulating protein or larger systems. In order to efficiently calculate long-range electrostatics, NAMD provides the choice between two different fast electrostatic solvers: the Particle Mesh Ewald (PME) method requires using periodic boundary conditions and is preferable for constant pressure simulation; the Multilevel Summation Method (MSM) supports either periodic or non-periodic boundaries along any dimension and is preferable for simulations using non-periodic or semi-periodic boundaries.

6 The "Extra Parameters" section contains commands which are applicable to more specific simulations. Included here are commands which characterize the spherical boundary conditions on the water sphere. These conditions prevent the sphere from undergoing evaporation or diffusion.

- Spherical Boundary Conditions
  - `sphericalBC`: indicates whether or not the simulation uses spherical boundary conditions; uses values `on` and `off`.
  - `sphericalBCcenter`: sets the x, y, z coordinates of the center of the sphere to which spherical boundary conditions are applied. This is why you need to record the center of your water sphere after you solvate!

  The next three commands are parameters which must be specified for spherical boundary conditions to work properly.

  - `sphericalBCr1`: sets the distance in Å at which the first boundary potential begins to act.
  - `sphericalBCk1`: The first potential applied by spherical boundary conditions to keep the sphere together (or pull it apart) is a harmonic one. This command sets the value of the force constant for that potential. Its value is specified in kcal/mol·Å^2.
  - `sphericalBCexp1`: sets the value of the exponent used to formulate the potential; must be a positive, even integer.

7 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).

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

NAMD User's Guide. The explanations of configuration file commands in the previous sections is far from complete. For an explanation of a more general configuration file, see Appendix E. 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`.

1.4.2 Run your Simulation

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

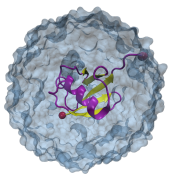
```
namd2 ubq_ws_eq.conf > ubq_ws_eq.log &
```

Setting the Path. In order to run namd2, you can provide the full path to the binary executable (e.g., `/home/me/installed/namd2`) or you can add the directory to your `PATH` environment variable (e.g., for bash you would `export PATH=/home/me/installed:$PATH`). An even better approach might be to create a namd2 alias that specifies the number of cores you plan to use for your simulations. For example, setting up a namd2 alias (again using bash) to run NAMD on four cores of your workstation, you would specify `alias namd2="/home/me/installed/namd2 +p4"`. The most common Linux distributions as well as Mac OS X default to using bash (Bourne again shell) for command line access.

Your simulation should take about 20 minutes to complete, and NAMD will produce the output files described in Section 1.7.

1.5 Ubiquitin in a Water Box: Simulation with Periodic Boundary Conditions

In this section, you will examine the minimization and equilibration of ubiquitin in a water box with periodic boundary conditions.



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.

1.5.1 Configuration File

1 Go to your `1-3-box` directory by typing `cd ../1-3-box` . Here, you will find a configuration file for the minimization and equilibration of ubiquitin in a water box. All output files for the minimization and equilibration of your ubiquitin in a water box system will be placed in this directory.

2 Open the configuration file, `ubq_wb_eq.conf` by typing `nedit ubq_wb_eq.conf`.

The configuration file contains some commands which are different than the water sphere configuration file. Here, these differences are pointed out and explained.

3 The only differences in this file lie in the "Simulation Parameters" section, where three new categories of parameters have been added. The "Output" section has also been modified. The new commands are listed:

- 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 $x = 42$Å, $y = 0$Å, $z = 0$Å. With each vector perpendicular, a rectangular 3-D box is formed.
  - `cellOrigin`: specifies the coordinates of the center of the periodic cell in Å. This is why you need to calculate the center of your water box after you solvate!
  - `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/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 (see below) 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).

  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 Å / 64 = 0.9375 Å 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

- `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.

Note that the commands which specify spherical boundary conditions have been completely removed, since this simulation is using periodic boundary conditions.

Minimization and Equilibration. Typically, MD minimization and equilibration simulations involve more than one minimization-equilibration cycle, often fixing and releasing 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.

1.5.2 Run your Simulation

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

```
namd2 ubq_wb_eq.conf > ubq_wb_eq.log &
```

A Note on Restarting Simulations. Often, it is 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. In that case, you wish to restart simulating from the equilibrated structure. You can use the configuration file `namd-tutorial-files/common/sample.conf` as a template for restarting simulations if you note 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. 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.
- Make sure you do not specify a temperature if you are using a velocity restart file. You should comment out the `temperature` command 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.