

# The CHARMM Water-Box Tutorial

## Contents

<b>1</b>	<b>Aim of the Tutorial</b>	<b>1</b>
<b>2</b>	<b>Getting Started</b>	<b>1</b>
2.1	Interactive Mode . . . . .	1
2.2	Batch Mode . . . . .	2
2.3	CHARMM Style . . . . .	2
<b>3</b>	<b>Building a Water Molecule</b>	<b>3</b>
3.1	Residue Files . . . . .	3
3.2	Parameter Files . . . . .	5
3.3	Generating Water . . . . .	6
3.4	Write the Output Files . . . . .	8
<b>4</b>	<b>Molecular Dynamics Simulations</b>	<b>9</b>
4.1	Building a Cube of Water Molecules . . . . .	9
4.2	Adding Periodic Boundary Conditions . . . . .	12
4.3	Analysing the Simulation . . . . .	14
<b>5</b>	<b>Web Links</b>	<b>15</b>

# 1 Aim of the Tutorial

In this tutorial you will first build a simple molecular system and then combine several molecules together which will be used to run a molecular dynamics simulation.

To successfully complete the tutorial, the following steps should be carried out in order:

1. Generate a water molecule
2. Build a system of 32 water molecules
3. Run a molecular dynamics simulation

## 2 Getting Started

For this tutorial you will be using a molecular modelling program called CHARMM. This program can be run interactively or in batch mode, both require running the CHARMM executable using the command `charmm`.

### 2.1 Interactive Mode

To run something interactively means launching the CHARMM program and then enter the commands one by one, pressing the <ENTER> key at the end of each line.

CHARMM has extensive error checking, so if a command is mistyped, an error message is provided. Often, if the error is serious enough, the CHARMM session will be terminated. You will then have to run CHARMM again and retype everything up to that final mistake. This can be a very inefficient way to run CHARMM. A more efficient way is to write input files and submit them to CHARMM in batch mode (see below). But first have a go at running CHARMM interactively.

1. Launch the CHARMM program by opening up a terminal window and typing the command `charmm`. A prompt like this will appear in the terminal window:

```
1
      Chemistry at HARvard Macromolecular Mechanics
      (CHARMM) - Developmental Version 31b1      August 15, 2004
      Copyright(c) 1984-2001  President and Fellows of Harvard College
      All Rights Reserved
      Current operating system: Linux-2.6.13-15.16-smp(i686)

      Maximum number of ATOMS:      60120, and RESidues:      72000
      Current HEAP size:      10240000, and STACK size:      2000000
```

This is CHARMM running in interactive mode. Commands can be entered at the prompt one by one in order to run various tasks within CHARMM.

2. Exit the interactive mode by hitting Ctrl-C or by entering two bogus commands (any words or letters). The latter method rewards you with a nice skull, informing you about a fatal error.

## 2.2 Batch Mode

This is the usual way of using CHARMM. To run in batch mode you need to create an input file that contains all the commands, and then feed it into the CHARMM program using the following command:

```
charmm < inputfile.inp
```

To make it easier running CHARMM using the batch mode, open *two* terminals and place the windows next to each other. Use one window for editing the input file and saving it after changing things. Use the second window to run CHARMM (the arrow up and down keys let you scroll through previous commands).

It is recommended that you type in the program line by line into your input file, then save it and run CHARMM to check whether you typed it correctly. This makes error checking easier. Also read the warnings and try to trigger error messages deliberately in order to get used to CHARMM's warnings and error messages. Often you will find that the warnings refer to/ask for the next line you are going to enter (due to some information which is still missing at that point). The fatal errors most times refer to the last line you just entered incorrectly.

1. Create a new file `water.inp` in one of the terminals and give the program a title. Titles are denoted by an asterisk starting the title line, with the last line of the title denoted by a single asterisk. You *must not* have an empty line before the title line for CHARMM reasons!

```
* CHARMM input script to build a single water molecule
* and energy minimize it
*
```

2. Save the file, switch to the other terminal and feed it into CHARMM like this:

```
charmm < water.inp
```

The output will be sent to the terminal which makes it easy for you to see any errors. CHARMM should finish with a `NORMAL TERMINATION BY THE END OF FILE` message. A skull and `ABNORMAL TERMINATION` message means that something is wrong and you need to check your input file.

## 2.3 CHARMM Style

When entering commands in CHARMM, the first word in every line specifies the actual command. Most commands have several options; these follow the command in the form of specific keywords. In most cases, only the first four letters of the keywords are required. For clarity, the commands and keywords are typed out in full in the following, however, in other tutorials you may find that the first four characters of each command are capitalised to emphasise this. CHARMM is case insensitive.

The command line is written in free format (that is, spaces don't matter). The output generated from executing each command is generally written to the standard output, which is the terminal

window where you are running CHARMM. It is possible to direct this output into a file, which will be shown later on.

Any text that appears after an exclamation mark ! is a comment and will not be executed by CHARMM. It is a good idea to include comments throughout the input file to make it clear what each set of commands are doing. For those unfamiliar with computer programming a comment is a statement that is ignored by the program, and is basically used to document the program.

```
! This is a comment
```

## 3 Building a Water Molecule

Using the file you just created, `water.inp`, you can create an input file that can build a single water molecule. This involves performing a number of steps:

1. Describing the water molecule (residue)
2. Declaring parameters for the water molecule (such as bond lengths)
3. Generating the structure

### 3.1 Residue Files

The *residue* is the basic building block in CHARMM. A *residue* could be an atom or monoatomic ion (e.g. an Ar atom or a  $K^+$  ion), a molecule (e.g. water or butane), or part of a more complex molecule (e.g. an amino acid or fatty acid chain). In this example we are building a water molecule, therefore only a single residue needs defining in order to build the system.

The description of the residue is normally saved in what is called the *Residue Topology File* or RTF as this is convenient for routine work. For simplicity, instead of writing a separate RTF, simply write the definition of the residue in your input file `water.inp`.

1. Open your `water.inp` file and add a title for the RTF so that you know what it is for. To do this, type:

```
BOMLEV -3
READ RTF CARD
* This is the water residue
*
```

The command `BOMLEV -3` tells CHARMM to be somewhat forgiving and not to abort the program for small errors. It will, however, provide warning messages that you should pay close attention to.

The next line, `READ RTF CARD` is a command that tells CHARMM to read in a description of the residue (Residue Topology File, RTF), in ASCII (human readable) format (`CARD`). The last two lines that start with asterisks form the title, as described before.

2. After the RTF title you need to add a line that tells CHARMM which version of the program you are using as this dictates the input format that will be used. This is the only place in the CHARMM script where you actually need to pay attention to the spacing and which column you are typing in. The rightmost numbers of the two numbers must be in columns 5 and 10:

```
31      1
```

The residue file needs to contain the basic description of the bonding and charges in the water molecule. To do this, you need to define a number of things:

- the atom types and their masses
- the actual residue (the water molecule)
- the actual atoms that make up the molecule; the exact number of atoms and names
- the connectivity between atoms (bonds)
- the bond angles
- the termini

Note: At this point you are not telling CHARMM what the bond lengths and angles are, you are simply declaring that certain bonds and angles exist

3. Define the atom types. For water you require hydrogen and oxygen atom types with masses of 1 and 16, respectively. To define these two atom types in the RTF, type in the lines:

```
MASS  1  H   1.0           ! Water - Hydrogen Atom
MASS  2  O  16.0           ! Water - Oxygen Atom
```

4. Define the water molecule residue by adding this line to the program:

```
RESIDUE  H2O  0.0           ! Water Model and its charge
```

This tells CHARMM to define a residue (**RESIDUE**), named H2O, and that this residue has an overall charge of 0.0.

5. Define the actual atoms to be included in the molecule. This is done by using the atom types given via the **MASS** command above. Each atom needs to be given a unique name to make them distinguishable from one another and assigned a charge.

Note: the charge on the water molecule is 0.0, therefore the charges on the individual atoms should also add up to 0.0.

To do this, type:

```
ATOM OH2 O   -0.82         ! Oxygen atom and charge
ATOM H1  H    0.41         ! Hydrogen atom and charge
ATOM H2  H    0.41         ! Hydrogen atom and charge
```

6. Define the atoms that will be bonded together by typing:

```
BOND   OH2 H1   OH2 H2           ! Define atom-atom connections
```

This tells CHARMM that the atoms OH2 and H1 form a bond, as well as OH2 and H2.

Note: More than one bond definition can go on a BOND line.

7. Define the bond angle that exists between the three atoms in water. To do this declare a dihedral angle theta by typing:

```
THETA  H1 OH2 H2           ! Define dihedral angle
```

8. Finally, we need to tell CHARMM how to treat the ends of the molecule (termini). If this residue was one part of a protein, we would need to describe how to terminate the sequence of amino acids (e.g. with an acetyl group). This is done via the `patch` command. The water molecule is complete as it is, therefore no patches need to be applied and you need to add these lines:

```
PATCH FIRST NONE LAST NONE    ! Do not patch the termini  
END
```

This command tells CHARMM that neither the first nor the last part of the molecule needs to be modified in any way (patched). The `END` statement tells CHARMM that `RESIDUE` is finished and at this point the molecule is defined.

For a complicated simulation hundreds of different residues might make up the system. Thankfully, there are `.rtf` files distributed with CHARMM that have pre-built residue definitions for amino acids, nucleic acids, fatty acids, and many other common molecules. It is important, however, to understand the `RTF` structure because it is likely that at some point you will want to simulate a molecule that has not been studied before and you will have to construct your own topology files to supplement those distributed with the program.

The definition of the water molecule atoms and bonds have now been completed. In the next section values for bond lengths, angles and non-bonding interactions will be defined as force-field parameters.

## 3.2 Parameter Files

In this section, CHARMM is given information that describes the bond lengths and angles etc. in the water molecule. These are termed force-field parameters (force constants, van der Waals radii and well-depths, etc.) so that it can carry out calculations based on the system energy (i.e. minimization and molecular dynamics). Like the residue topologies, files are distributed with CHARMM that contain force-field parameters for a wide array of atom types. If the parameters you need are not included in CHARMM you may have to consult the literature or develop your own. Since the parameters are well known for the water molecule you can enter them directly into the input file.

1. Start as before by entering a title for the parameter section of the CHARMM script:

```
READ PARAMETERS CARD  
* Bond and angle constraints for water  
*
```

This tells CHARMM it is about to read in formatted parameter information for the water molecule.

2. Enter the force constant for the H–O bonds ( $450.0 \text{ kcal}/(\text{mol} \cdot \text{Å}^2)$ ) that are defined in the residue file and the equilibrium bond lengths ( $0.9572 \text{ Å}$ ) by typing:

```
BOND
H O 450.0 0.9572          ! Force constant of the H-O bond
```

With this, CHARMM is told which atom types (defined earlier by the MASS command) are to be used with the given parameters, followed by the parameter values themselves.

3. Enter the value for the bond angle. The force constant for the bond angle ( $55.0 \text{ kcal}/(\text{mol} \cdot \text{radians}^2)$ ) and the equilibrium angle ( $104.52^\circ$ ):

```
THETA
H O H 55.0 104.52        ! Equilibrium angle in degrees
```

For a larger molecule, a set of dihedral parameters would have to be provided, but for water the bond and angle terms are sufficient to describe the intra-molecular potential.

4. The last set of parameters that needs to be provided are for the Lennard-Jones (or van der Waals) potential that describes the non-bonding intermolecular interactions. The ‘well depth’, in  $\text{kcal}/\text{mol}$ , and the van der Waals radius, in Angstrom, at the bottom of the well are needed. The following parameters are suitable for a water molecule (note that the first zero in each column is a parameter which is not used):

```
NONBONDED
O 0.0 -0.1521 1.7682 ! Well-depth and van der Waals radius
H 0.0 -0.04598 0.2245 ! Well-depth and van der Waals radius
END
```

This is the last set of parameters that is needed for the water molecule, therefore the END command is added.

The addition of force-field parameters to the previously defined water molecule is now complete. In the next section this information is used to build a system containing a single water molecule.

### 3.3 Generating Water

At this point, the RTF and parameters you have entered could be used to build any system containing water molecules. This could be a system with just a single molecule (as will be done now) or an entire “box” of water molecules (as will be shown later). The RTF and parameters are the building blocks of the system; in this section these building blocks are used to generate an actual molecular system that can be studied using CHARMM.

1. Continue adding commands to the file by giving a title to this section of the input file:

```

READ SEQUENCE CARD
* This is the water molecule
*
1                      ! Number of residues
H2O                    ! Name of the residue

```

CHARMM is told to read a formatted sequence, and add a title. Next, the number of residues is defined, which will be used to build the system, in this case one, and the name of the residue, which was defined earlier as H2O.

- Next, a name needs to be given to the system of molecules and the system be generated:

```

GENERATE WAT           ! Generate structure with name WAT

```

This tells CHARMM to generate the structure and give the name WAT to it. After entering this command, CHARMM should give you a summary description of the system including the number of atoms, bonds, angles, residues, etc.

- Having defined the system, CHARMM needs to be provided initial coordinates to define the position of the water molecule. This is done through the COORDINATES card:

```

READ COORDINATES CARD FREE
* The coordinates for the water molecule
*
1
1 1 H2O OH2 0.0 0.0 0.0

```

This set of commands has placed the OH2 (oxygen) atom of the residue at the origin. The first statement in this group tells CHARMM to read formatted coordinates (READ COORDINATES CARD), and that the spacing does not matter (FREE). Next comes the title for this section, and the number of residues, in this case one. The following line tells CHARMM in residue number 1, to place the number 1 atom in the H2O residue, which will be an OH2, at the origin. The three values of 0.0 are the x, y and z coordinates.

Note that only one of the three atoms (the oxygen) was placed in the water residue. The hydrogens can be automatically added to the oxygen using the HBUILD command. This is possible because we previously defined the bonding between the hydrogens and the oxygen (see section 3.1).

- Add the HBUILD command to your input file:

```

HBUILD                ! Add the hydrogen atoms

```

- Print out the coordinates of the water molecule to make sure that the hydrogens were added:

```

PRINT COORDINATES    ! Print out the coordinates

```

- The water molecule is now built and defined at an initial position and it is possible to perform energy-based calculations, such as a minimization:

```
MINIMIZE NRAP NSTEP 10      ! Perform an energy minimization
PRINT COORDINATES          ! Print out the coordinates
```

These commands tell CHARMM to perform an energy minimization on the molecule using the Newton-Raphson method with 10 steps, and then to print the coordinates. There are numerous algorithms for energy minimization available in CHARMM, for example to use the steepest descent method the `SD` keyword is substituted for `NRAP`.

The water molecule is too small to show any big change of the coordinates, however, basically this is your first CHARMM script which builds and optimizes a molecule from scratch.

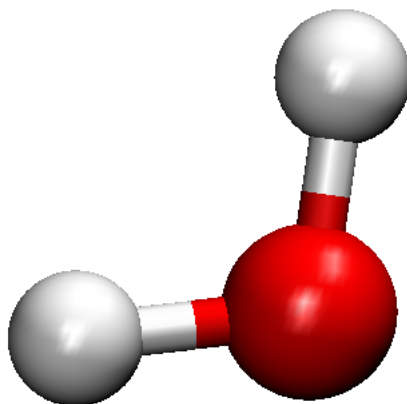
### 3.4 Write the Output Files

To write all atoms of the system that has just been generated, add the following commands to the end of your input file:

```
! The psf file contains all the information on the makeup of the system
OPEN UNIT 1 WRITE CARD NAME water.psf
WRITE PSF CARD UNIT 1
```

```
! The CRD file will contain the coordinates of all the atoms in the system
! written in CHARMM format
OPEN UNIT 1 WRITE CARD NAME water.crd
WRITE COORDINATES CARD UNIT 1
```

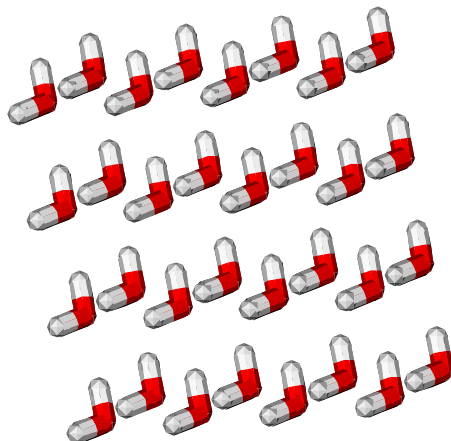
Re-run your input file and you will see that two files `water.psf` and `water.crd` have been created. These files can be opened using a software program such as VMD in order that you can view the molecule you have just built. If viewed in CPK mode the molecule should appear as shown below. (CPK is the acronym of Corey, Pauling, Kolthun display style which represent each atom by a sphere of the appropriate radius.)



## 4 Molecular Dynamics Simulations

### 4.1 Building a Cube of Water Molecules

For this section of the tutorial the `water.inp` input file will be extended in order to build a larger system containing a number of water molecules as shown below.



1. Copy the old file to a new filename:

```
cp water.inp water32.inp
```

The `water32.inp` will produce more output than before. This is because every intermediate step of the simulation is printed. For this reason, it is better to redirect the output into a file that can be saved and reviewed for errors.

2. Redirect the output to a file like this:

```
charmm < water32.inp > water32.out
```

CHARMM will read in the file ‘`water32.inp`’, carry out all the commands, and save the output to the file ‘`water32.out`’ which contains all the simulation results as well as any error messages. *Be careful* not to accidentally direct output into the `.inp` file as this would overwrite your script!

If you run the command above you will see that your `water32.out` file contains the same output as you got for your earlier work generating the water molecule.

3. The topology and parameter definitions are identical to the single water molecule case so these do not need altering. You just need to change the title as shown below:

```
* CHARMM input script to run an MD simulation on a  
* system of 32 water molecules
```

```

BOMLEV -3
READ RTF CARD
* This is the water residue
*
  31    1
MASS  1  H  1.0          ! Water - Hydrogen Atom
MASS  2  O 16.0          ! Water - Oxygen Atom

RESIDUE H2O  0.0          ! Water Model and its charge

ATOM OH2 O  -0.82        ! Oxygen atom and charge
ATOM H1  H   0.41        ! Hydrogen atom and charge
ATOM H2  H   0.41        ! Hydrogen atom and charge

BOND   OH2 H1  OH2 H2    ! Define atom-atom connections
THETA  H1 OH2 H2        ! Define dihedral angle

PATCH FIRST NONE LAST NONE ! Do not patch the termini
END

! =====

READ PARAMETERS CARD
* Bond and angle constraints for water
*

BOND
H O  450.0  0.9572      ! Force constant of the H-O bond

THETA
H O H  55.0  104.52     ! Equilibrium angle in degrees

NONBONDED
O  0.0 -0.1521  1.7682  ! Well-depth and van der Waals radius
H  0.0 -0.04598 0.2245  ! Well-depth and van der Waals radius
END

! =====

```

- At this point in your file you originally generated a single water molecule. You need to modify it now to define a system of 32 water molecules by adding the following set of commands to it:

```

READ SEQUENCE H2O 32

GENERATE WAT SETUP
READ COORDINATES CARD FREE
* Oxygen Coordinates
*
  32

```

```

1  1  H2O  OH2  -4.5  -4.5  -4.5
1  2  H2O  OH2  -1.5  -4.5  -4.5
1  3  H2O  OH2  -4.5  -1.5  -4.5
1  4  H2O  OH2  -4.5  -4.5  -1.5
1  5  H2O  OH2  -1.5  -1.5  -4.5
1  6  H2O  OH2  -1.5  -4.5  -1.5
1  7  H2O  OH2  -4.5  -1.5  -1.5
1  8  H2O  OH2  -1.5  -1.5  -1.5
1  9  H2O  OH2  -4.5  -4.5   1.5
1 10  H2O  OH2  -1.5  -4.5   1.5
1 11  H2O  OH2  -4.5  -1.5   1.5
1 12  H2O  OH2  -4.5  -4.5   4.5
1 13  H2O  OH2  -1.5  -1.5   1.5
1 14  H2O  OH2  -1.5  -4.5   4.5
1 15  H2O  OH2  -4.5  -1.5   4.5
1 16  H2O  OH2  -1.5  -1.5   4.5
1 17  H2O  OH2  -4.5   1.5  -4.5
1 18  H2O  OH2  -1.5   1.5  -4.5
1 19  H2O  OH2  -4.5   4.5  -4.5
1 20  H2O  OH2  -4.5   1.5  -1.5
1 21  H2O  OH2  -1.5   4.5  -4.5
1 22  H2O  OH2  -1.5   1.5  -1.5
1 23  H2O  OH2  -4.5   4.5  -1.5
1 24  H2O  OH2  -1.5   4.5  -1.5
1 25  H2O  OH2  -4.5   1.5   1.5
1 26  H2O  OH2  -1.5   1.5   1.5
1 27  H2O  OH2  -4.5   4.5   1.5
1 28  H2O  OH2  -4.5   1.5   4.5
1 29  H2O  OH2  -1.5   4.5   1.5
1 30  H2O  OH2  -1.5   1.5   4.5
1 31  H2O  OH2  -4.5   4.5   4.5
1 32  H2O  OH2  -1.5   4.5   4.5
! format above is (segment #) (residue #) (residue name) (atom type)
! (x coor) (y coor) (z coor)
HBUILD
PRINT COORDINATES

```

This sequence of commands you just entered included a compact read sequence used to generate the system of 32 molecules. Then the coordinates of the oxygen atoms of all 32 molecules are entered. These initial coordinates were made by placing the oxygen atoms on a simple grid. Not a good way to make realistic water, but it will serve as a starting point for a simulation, which will lead to an accurate structure for liquid water. In many simulations the initial coordinates will be based on a previous simulation so that you start with a more random configuration. As before, this was followed by the `HBUILD` and `PRINT COORDINATES` commands.

5. Before adding the commands needed to run molecular dynamics it is recommended that you write some output files in order to keep track of what has been done. These files can be opened in the future and make it unnecessary to repeat the structure generation and initial coordinate building procedures. The files are written out in `CARD` format which means that it will be a simple text file.

Add the following commands to your input file to create the output files:

```
! The psf file contains all the information on the makeup of the system
OPEN UNIT 1 WRITE CARD NAME water.psf
WRITE PSF CARD UNIT 1

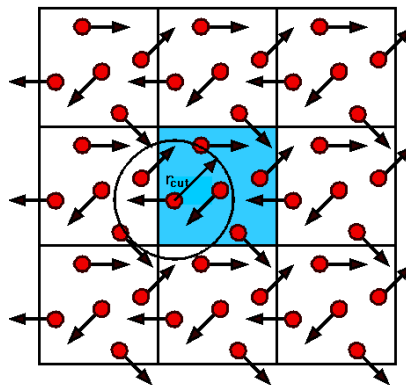
* PSF for 32 water molecule system
*
! The CRD file will contain the coordinates of all the atoms in the system
! written in CHARMM format
OPEN UNIT 1 WRITE CARD NAME water.crd
WRITE COORDINATE CARD UNIT 1

* Initial coordinates for starting point of simulation
*
! The PDB file will contain the coordinates of all the atoms in the system
! written in Protein Data Bank format
OPEN UNIT 1 WRITE CARD NAME water.pdb
WRITE COORDINATES PDB UNIT 1

* Initial coordinates for starting point of simulation
*
```

## 4.2 Adding Periodic Boundary Conditions

A problem with the system so far is that 32 molecules are not likely to provide a realistic representation of a macroscopic sample. The artifacts due to the small size of the system can be minimized by applying periodic boundary conditions. Essentially copies of the system are created and placed all around the central simulation cell. To invoke PBC, the size and shape of the central cell needs to be defined using the CHARMM crystal facility. There are many box shapes available, of which the example will use a simple cube. (Note that after constructing the crystal the results are saved into a file for use in later calculations, just as before with the psf and crd.)



1. Add the following commands to define a cube with sides of 13.0 Angstroms.

```
CRYSTAL DEFINE CUBIC 13.0 13.0 13.0 90.0 90.0 90.0
CRYSTAL BUILD NOOPERATIONS 0
```

```
OPEN UNIT 1 WRITE CARD NAME cubic.cry
WRITE CRYSTAL CARD UNIT 1
```

```
* PBC for simple cubic geometry
*
```

The first three numbers after the keyword **CUBIC** are the edge lengths of the cube in Å and the next set of 3 numbers define the angles of the box. You will want to double check and make sure that these are the right sizes for the box. This can be done by checking the output file. If the average pressure is negative, the box is too big and should be reduced, if it is positive, the opposite is true. The optimal pressure should be somewhere around 0. It does not have to be exactly zero, but it should be close.

A difficulty with MD simulations of hydrogen atoms is that the small mass of hydrogen leads to very rapid bond vibrations that require short time steps to ensure accurate integration of the equations of motion.

2. To allow longer time scales, it is common to fix the bond lengths to H atoms using the SHAKE procedure:

```
SHAKE BONH PARAMETERS SELECT ALL END ! Constrain all bonds to H atoms
                                         ! to their parameter values
```

Before you run the dynamics command you need to setup files that will capture the output from the simulation. The positions of the atoms during the simulations will be periodically written out for subsequent viewing and analysis. This trajectory will take up a lot of disk space, therefore it is written in the more compact (but only machine-readable) binary format which in CHARMM is called the **FILE** format.

3. Set up your output files by adding:

```
OPEN UNIT 11 WRITE FILE NAME water.dcd
```

There are many decisions to be made before carrying out an MD simulation. The following is a relatively simple dynamics protocol for carrying out the equilibration of our water box.

Note: the `'-'` is used to continue a line in a CHARMM script, the commands are grouped roughly by subject. Look up the definitions of each command, in the **dynamc** documentation (the last line of commands can be found in the **nbonds** documentation). It is important to understand each of these terms!

4. Once you understand what these commands are doing add them to your input file:

```
DYNAMICS STRT -
  NSTEP 20000 TIMESTEP 0.001 -
  NPRINT 100 IPRFRQ 1000 NTRFRQ 5000 -
  NSAVC 100 IUNCRD 11 IUNWRI 1 -
  FIRSTT 100.0 FINALT 300.0 TEMINC 50.0 IHTFRQ 500 -
  IASORS 1 IASVEL 1 ISCVEL 0 IEQFRQ 1000 ICHECW 0 -
  TWINDH 10.0 TWINDL -10.0 -
  INBFRQ 25 IMGFRQ 25 CUTNB 12.0 CUTIM 12.0 CTOFNB 10.0 -
  CTONNB 8.0 SHIFT VSWITCH
```

5. Run the simulation by typing:

```
charmm < water32.inp > water32.out
```

This will run the simulation, print basic results in the output file (`water32.inp`) and write out a trajectory file (`water.dcd`).

6. Read the output file carefully to see if there were any mistakes. Search for the word 'WARN', as searching for this word will find all of the places in the output file where CHARMM had an error. By tracing these errors back to what commands caused them, you can debug your CHARMM scripts.

### 4.3 Analysing the Simulation

After successfully running the `water32.inp` script without error, you should also check the properties of the system, such as temperature, pressure, and potential energy to make sure that the simulation ran without problems and to decide if your system has reached equilibrium yet. Equilibrium is when the energies and properties have reached a stable state.

You should start by checking the temperature of the simulation, which is listed periodically in the output file, to make sure that it stays near the limit that you have set for it in the dynamics protocol.

To view what is happening, you can use VMD. Click on **File => New Molecule => Browse** and select the `.psf` file. After clicking **Load** the file will show up in the main VMD window. Select it (left mouse click), right click and select **Load Data Into Molecule...** Click **Browse**, select the `.dcd` file and load it. You will then see the MD simulations of your 32 water molecules and play the video with the controls in the main VMD window. Under **Graphics => Representations** you can play about with the **Drawing Method** to get a 3D representation of the molecules.

A small shell script is very helpful and will speed up tasks which are needed repeatedly. For this, you need programs such as `grep`, which will search a text file for a certain phrase and print out all of the lines in the file that contain the phrase, and `cut` and `paste` which cut certain characters out of each line and add lines of text together, respectively.

Here is an example of a bash script to make three files, one that contains the potential energy of the simulation over time, one for temperature and another for pressure.

1. Create a new, empty file using `touch` and make it executable:

```
touch makeplot
chmod +x makeplot
```

2. Open the file and enter the following lines then save it.

```
grep 'DYNA>' water32.out | cut -c19-28 > time.tmp
grep 'DYNA>' water32.out | cut -c69-79 > temp.tmp
grep 'DYNA>' water32.out | cut -c55-68 > pe.tmp
grep 'DYNA PRESS>' water32.out | cut -c54-67 > press.tmp

paste time.tmp temp.tmp > plot_tempcp.dat
paste time.tmp pe.tmp > plot_pecp.dat
paste time.tmp press.tmp > plot_presscp.dat
```

The bar (|) symbol is called a pipe and ‘feeds’ the output of one command into another. The first command in the first line (**grep**) searches for all lines in water.out which contain ‘DYNA>’. The found lines are then piped into the **cut**-command, which cuts out columns 19 to 28 (the times) and this output is directed into the file time.tmp. You should try entering just the **grep** command (everything before the first bar) to see the output and then everything till the second bar to understand what is happening there.

3. Run the script from the command prompt with the following command (the ./ means that it is in the current directory, which needs to be said explicitly under Linux):

```
./makeplot
```

The data files can then be opened in Excel as .csv and plotted to make sure that the simulation went as planned.

4. View the changes to the system by examining it before and after the simulation. This can be done by opening the .psf and .crd and .trj files in VMD. You should see that the water molecules have moved from their initial starting positions, where they were unrealistically ordered, to a more realistic distribution.

Now you know the basics of working with CHARMM scripts. Be sure to check the CHARMM documentation if you have trouble with CHARMM. Many of the pages contain examples you can try for yourself if you have any problems. The documentation can be found at <http://www.charmm.org/document/Charmm/chmdoc.shtml>.

## 5 Web Links

- This tutorial was adapted from [http://persweb.wabash.edu/facstaff/fellers/tutorial/water\\_tutorial/page\\_1.html](http://persweb.wabash.edu/facstaff/fellers/tutorial/water_tutorial/page_1.html)
- The CHARMM homepage with documentation <http://www.charmm.org/document/Charmm/chmdoc.shtml>
- Molecular Dynamics Tutorial [http://www.ch.embnet.org/MD\\_tutorial](http://www.ch.embnet.org/MD_tutorial)