(Rev 10/7/08)

Lab 9 Molecular Dynamics of DNA using Amber

Contents

Acknowledgments: This lab is based on the amber tutorial for Poly(A)-Poly(T). The web address for this is http://amber.scripps.edu/polyA-polyT/index.html. The material at this web site was used for a tutorial at the Pittsburgh Supercomputer Center. What is not clear is that the original material was largely written by Thomas Cheatham. The lab has been modified so that (1) it works at the WVU Computational Chemistry and Molecular Modeling lab and (2) so that it works with Amber 9.0.

Objectives

What's a trajectory, what's in a trajectory and how to get at it...

Molecular mechanics (MM) was developed on relatively small molecules. It works reasonably well for small molecules. However, to insure that the global minimum energy structure has been located the user must start from several quite different conformations and have some sense of what structures are likely candidates for the global minima. As molecules increase in size starting from multiple conformations, etc. becomes impossible and the chances of locating the global minima decrease rapidly.

Molecular dynamics (MD) has been developed to facilitate finding global minima. It is not a alternative to MM, rather they are used in combination. Thus, after structures are built (DNA, protein) they are first minimized with MM, mainly to remove particularly bad interactions, and then subjected to MD. The purpose of this lab is to provide you an example of this process.

The final part of this lab is to provides an example of how one goes about analyzing the results from a molecular dynamics run using some of the tools supplied within **Amber** and some available through other sources. Particular emphasis will be placed on analyzing molecular dynamics trajectories that need to be "imaged" after running a particle mesh Ewald calculation; as you work through the lab, you will see what the problem is and why this is necessary.

Outline

This lab has several the following sections:

Build and solvate a ds DNA, $d(A)_{10}{\cdot}d(T)_{10}$ in the A DNA conformation
Equilibrate the water molecules/ions (MM)
Equilibrate the water molecules/ions (MD)
Relax the DNA (MM)
Minimize the whole system (MM)
Minimize the whole system (MD)

Imaging and the fix.
Generating movies of the dynamics with **moilview** and **vmd**
Using **carnal** to generate RMSd plots of deviation from the initial structure and to create average structures.
Using **mdanal** to look at atomic positional fluctuations over a dynamics run.
rdparm is a program to analyze AMBER parameter / topology files.
- analyzing `top/crd` files
- calculating mean square displacements (or diffusion constants)
- calculating radial distribution functions

2D RMS plots
Using **curves** and/or **dials_and_windows** to generate helicoidal parameters.

As the outline suggests, minimization is a stepwise process. On large, solvated molecules, MM and MD usually fail if you attempt to minimize without gradually removing constraints. This is mainly because the methods used to build large molecules with solvation result in structures that are far from the ideal. While one might attempt to improve the methods used to build starting structures - and this is an area of research - it is currently easier to use the approach described here.

Setup

If you are working on Taz or Hal then log on to Amber, Frodo, Iris, or Opal. Once you've logged in, create a new subdirectory called `polyatA`. Change to this directory and copy all of the files that are in the directory called `/disk02/usr/people/amberlab/amberlab9` (or . `/usr/people/amberlab/amberlab9` on Opal).

```
telnet frodo (or amber, iris, opal)
mkdir polyatA
cd polyatA
cp /disk02/usr/people/amberlab/amberlab9/* .
```

The following files should be copied into your polyat directory:

```
fixit.in
ata_fsi_min1.in
ata_fsi_min2.in
ata_fsi_md1.in
ata_fsi_md2.in
nucgen.dat
ata_fsi_calc_backbone_rms.in
process_mdout.perl
process_carnal.perl
carnal_rms_start.in
ata_fsi_reimage.ptraj
```

These files are the sander control files you will need (`.in` files) and a few miscellaneous files needed for building the initial structure and analyzing the results. I have set them up for you to the extent this is possible. However, as you will see, they will still require some minor modifications. I recommend that you examine these files and understand what they are doing.

Notes

1) The biggest problem you are likely to have with this laboratory is keeping track of files.

The files listed above are only the input files for **sander**. Each minimization (MM or MD) has 5-6 files associated with it, though some files will be used repititively. No matter, when you are done, you will have generated more than 60 files. It is recommended that you use the file names indicated. It will help should you run into problems.

2) You will not be able to complete this lab in six hours. Some of the minimizations will require over seven hours to complete assuming that you have the computer to yourself. During the normal lab time you will be able to get most of the set up done and some of the initial minimizations. To complete the lab will require that you come to the lab several times and start the next step - which will only require a few minutes. To help you with this, I have given times required for each step. However, the times given are for runs where the computer devoted 98%+ resources to the run. If there is more than one job running, your job will take longer. To ease this process, check which Octane (Amber, Iris, Frodo) is running the fewest jobs and make your runs on that machine. Also, Iris and Frodo are dual processor machines so two jobs running on them is the equivalent to one job on Frodo/Opal). To determine what is currently running on a Amber, Frodo, Iris or Opal run the `top` command:

```
top
```

You'll see several jobs running. Look for **sander** in the last column. The machine with the fewest copies of **sander** running is the machine to run your job on. To exit the `top` command type q or CTRL-C.

Building the solvated $d(A)_{10}{\cdot}d(T)_{10}$ A DNA

Introduction

The following routine should be pretty familiar to you by now so the procedure will be only outlined. If you have difficulty, please refer to previous labs where structures were being built in **Amber**.

Build and Solvate the $d(A)_{10}{\cdot}d(T)_{10}$ A DNA

1) Run **nukit**. Make $d(A)_{10}{\cdot}d(T)_{10}$ in the A DNA conformation (ie, the DNA type is ADNA).

2) Run **nucgen**. Be sure to 'fix' the ata.pdb generated by inserting a TER card between residues 10 and 11, and to be safe, at the end of the file.

```
nucgen -O -i nuc.in -o ata.out -d nucgen.dat -p ata.pdb
```

3) Run **xleap**, load ata.pdb, add hydrogens, write a new `pdb` file and `crd` and `top` files for **sander**. Do this (once in **xleap**) by loading `ata.pdb` (here, the molecule will be called `ata` and loading will add missing hydrogens), then save as both a `pdb` file (`savepdb` command) and as `crd` and `top` files (`saveamberparm`).

4) Outside of **xleap** (either quit or open a new shell) run **sander** to do a fixup on the added hydrogens. The command line is:

```
sander -O -i fixit.in -o fixit.out -p ata.top -c ata.crd \
       -r ata.rst -ref ata.crd &
```

When **sander** finishes, check `fixit.out` to make sure it ran without any errors. Run **ambpdb** using `ata.top` and `ata.rst` as your topology and coordinate files, respectively. Call the output file `ata_fix.pdb`.

5) Next, it's time to add some water and some ions. Run **xleap**. Load `ata_fix.pdb`. (here it is referred to as `ata`). Solvate (`solvatebox ata TIP3PBOX 10.0`) and then add some sodium ions (`addions ata Na+ 0`). It would be a good idea to make sure you have a block of water with some DNA and sodium ions in it so check it out in the unit editor). Save the resulting structure (`pdb`, `top`, and `crd` files with `savepdb` and `saveamberparm`. Recommended filenames: `ata_fsi.pdb`, `ata_fsi.top`, `ata_fsi.crd`)

Equilibrating and running the poly(A)-poly(T) structure

Introduction

After you've built up your structure, you are ready to begin minimization and molecular dynamics simulations. In general, before production dynamics should be run, potentially bad van der Waal (nonbonded) and electrostatic interactions should be minimized. If you used **xleap** to solvate the structure, water was placed from a pre-equilibrated water box around the solute. This water hasn't felt the influence of the solute and moreover there may be gaps between the solvent and solute and solvent and box edges. Therefore, before one runs production dynamics on the system, it is wise to let the water relax (or equilibrate) around the solute to come to an equilibrium density, etc., and unless the water density is perfectly matched to the box size and all "holes" around the solute filled, constant pressure dynamics should be used to allow the box size to change. Otherwise, if constant volume simulations are run, "vacuum" bubbles may appear in the box.

In this section is presented one path towards equilibrating a solvated DNA structure for use in

simulations with the particle mesh Ewald (PME) method. After equilibration is demonstrated, we set up a production "PME" run. There are many differing opinions on how one should equilibrate a simulation. Herein is presented only one.

One big issue with the constant pressure simulations is that in order to match a target pressure, the box size is changed by scaling of the atom or molecule positions to match a target pressure [See Berendsen, H.J.C., Postma, J.P.M., van Gunsteren, W.F., DiNola, A., & Haak, J.R. *J. Comp. Phys.* **81** 3684-3690 (1984)]. This means that one has to be very careful using belly simulations with constant pressure. A belly simulation (IBELLY = 1) implies that you have chosen a subset of the system to be held fixed. This is performed by zeroing the forces on each of the fixed atoms at each step.

However, atom positions may change as a result of pressure scaling. If atom based scaling is used (NPSCAL = 0), then if all the solute atoms are held fixed, the positions of the individual atoms will be shifted as the box size changes. This means that you will move away from your initial structure, even with a belly on all the solute atoms. On the other hand, if you use molecule based pressure scaling (NPSCAL = 1), then only the center of mass of the molecules will be shifted. This is fine if your solute is only a single molecule. However, the two strands of the DNA are each in a separate molecule. Therefore, without some trickery, you would be advised to run a (slightly slower) simulation with position restraints (NTR = 1) rather than using a belly. This is the approach used here.

At the end of this lab will be discussed the trickery (or modifications to the prmtop file) required to get belly to work correctly and will speed equilibrations (since it is slightly faster).

NOTE: In this lab you may not be able to run all of the MM and MD calculations unless you are willing to start them at the end of the day, over the weekend, etc. The results for an entire equilibration cycle on the input files describe below takes ~1 day (21.5 hours with 98% of CPU resources) to run on **Amber**.

Periodic Boundaries

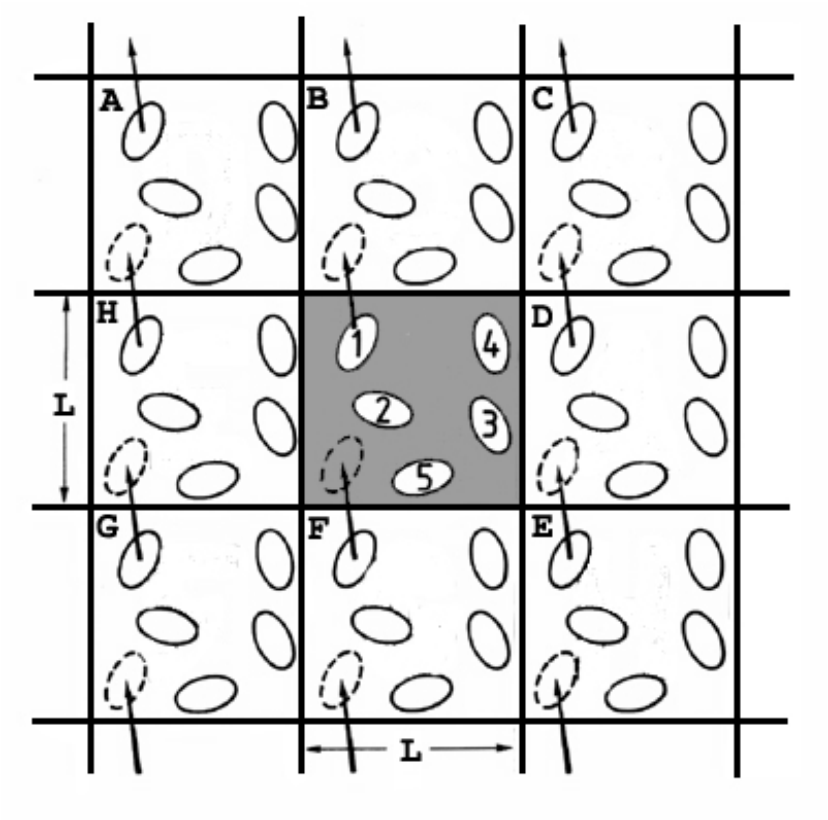
A realistic model of a solution requires a very large number of solvent molecules to be included along with the solute. Simply placing the solute in a box of solvent is not sufficient, however, since some solvent molecules will be at the boundary between solute and solvent and others will be within the bulk of the solvent a large number will be at the edge of the solvent and the surrounding vacuum. This is obviously not a realistic picture of a bulk fluid. In order to prevent the outer solvent molecules from ‘‘boiling’’ off into space, and to allow a relatively small number of solvent molecules to reproduce the properties of the bulk, periodic boundary conditions are employed. In this method the particles being simulated are enclosed in a box which is then replicated in all three dimensions to give a periodic array, a two-dimensional representation of which is shown below. In this periodic array a particle at position *r* represents an infinite set of particles at:

$$\mathbf{r} + i\mathbf{x} + j\mathbf{y} + k\mathbf{z} \quad (i,j,k \rightarrow -\infty, +\infty)$$

where **x, y** and **z** are the vectors corresponding to the box edges. During the simulation only one

of the particles is represented, but the effects are reproduced over all the image particles with each particle not only interacting with the other particles but also with their images in neighbouring boxes. Particles that leave one side of the box re-enter from the opposite side as their image. In this way the total number of particles in the central box remains constant.

Upon initial inspection such a method would appear to be very computationally intensive requiring the evaluation of an infinite number of interacting pairs. However, by choosing the non-bonded cutoff distance such that each atom ‘‘sees’’ only one image of all of the other atoms the complexity is greatly reduced. Hence as long as the box size is more than twice the cut-off distance it is impossible for a particle to interact with any two images of the same particle simultaneously. This cutoff requirement is important for us to remember when we choose what cutoff to use later on. It must not be larger than half the smallest box dimension.

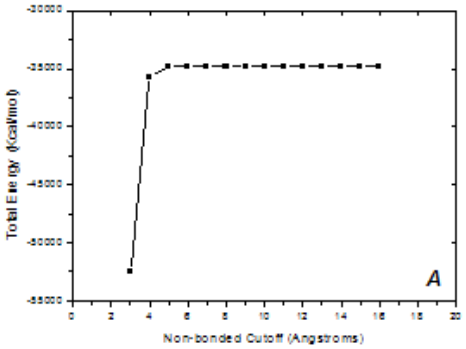


A two-dimensional array of boxes. As molecule 1 moves from the central box into box B it is replaced by it's image which moves from box F into the central box. This movement is replicated across all the boxes.

Choosing a suitable cut off

The choice of cutoff can have a dramatic influence on the results obtained from the MD simulation. The use of the particle mesh Ewald method for treating the long range electrostatics also reduces the effect of the cutoff. However, there is still a trade off between cutoff size and simulation time.

The choice of cutoff is therefore an important factor in carrying out a MD simulation. The figure below shows the variation in total system energy after 100 steps of steepest descent minimisation of our solvated DNA 10-mer as a function of cutoff distance. It can be seen that the total energy converges quickly and that by 10 angstroms the gain in going to large cutoffs is small and no longer justifies the extra computational cost. Hence for the solvated simulations that will be carried out in the remainder of this tutorial a cutoff of 10 angstroms will be utilized.



Energy of 10-mer DNA after 100 steps of steepest descent minimisation (**A**), and processor time (1 x 3 GHz P4) in seconds (**B**), as a function of cutoff distance.

Minimization holding the solute fixed

The first step is a short minimization run to remove any potentially bad contacts. Here you will be using position retraints which are more general and advised (unless you resort to trickery and modify your prmtop file to support belly equilibration). In this step, we are placing positional restraints of 500.0 kcal on each of the solute atoms (the DNA). Thing's to notice in the input file:

- IMIN = 1: Minimization is turned on.
- NTR = 1: means we have turned on position restraints and therefore have to specify via GROUP input (see the appendix of the AMBER 7.0 manual) the atoms which are restrained as well as the force constant. In this example, we use a force constant of 500 kcal/mol-angstrom and restrain residues 1 through 20. This means that the water and counterions are free to move. This GROUP input looks like the following:

```
Hold the DNA fixed
500.0
RES 1 20
END
```

Note that whenever you run using the GROUP option in the input file, carefully check the top of the output file to make sure you've selected as many atoms as you thought you did.

```
GROUP   1 HAS HARMONIC CONSTRAINTS   500.0000
  GRP   1 RES    1 TO   20
       Number of atoms in this group =   638
```

- MAXCYC = 1000, NCYC >= MAXCYC: This means that we will be running 1000 steps of steepest descent minimization. Conjugate gradient minimization typically fails with rigid water (TIP3P) so by keeping NCYC > MAXCYC we never turn it on. This avoids the common "LINMIN failure".
- CUT = 10.0, NSNB = 10: We are running with a 10 angstrom cutoff and updating the pairlist every 10 steps.

When running NTR = 1 simulations, the reference coordinate file for the restraints must also be listed. It is the same format as the standard AMBER coordinates and must match the `prmtop`. To run enter:

```
sander -O -i ata_fsi_min1.in -o ata_fsi.out -c ata_fsi.crd \
       -p ata_fsi.top -r ata_fsi.rst -ref ata_fsi.crd &
```

```
Run time: 24 min
```

```
Note: When the run is done, generate a pdb file.
      Filename ata_fsi_min1.pdb
```

Take a look at the output file referenced above. You will notice that there is initially a rather high electrostatic energy (EEL) drops rather rapidly. This represents the optimization of the favorable water-water and water-DNA electrostatic interactions. The CONSTRAINT energy represents the effect of the harmonic restraints and it rises rapidly.

Minimization Stage 2 - Minimizing the entire system

Now we have minimised the water and ions the next stage of our minimisation is to minimise the entire system. In this case we will run 2,500 steps of minimisation without the restraints this time. Here is the input file:

```
ata_fsi_min2.in

poly AT 10-mer, initial min, whole system
 &cntrl
  imin =1,
  maxcyc = 2500,
  ncyc = 1000,
  ntb = 1,
  ntr = 0,
  cut = 10,
 /
```

Note, choosing the number of minimisation steps to run is a bit of a black art. Running too few can lead to instabilities when you start running MD. Running too many will not do any real harm since we will just get ever closer to the nearest local minima. It will, however, waste cpu time. Since minimisation is generally very quick with respect to the molecular dynamics it is often a

good idea to run more minimisation steps than really necessary. Here, 2,500 should be more than enough.

We run this using the following command. Note, we use the `rst file from the previous run, which contains the last structure from the first stage of minimisation, as the input structure for stage 2 of our minimisation. Note, we also no longer need the -ref switch:`

```
sander -O -i ata_fsi_min2.in -o ata_fsi_min2.out -p ata_fsi.top \
      -c ata_fsi.rst -r ata_fsi_min2.rst &
```

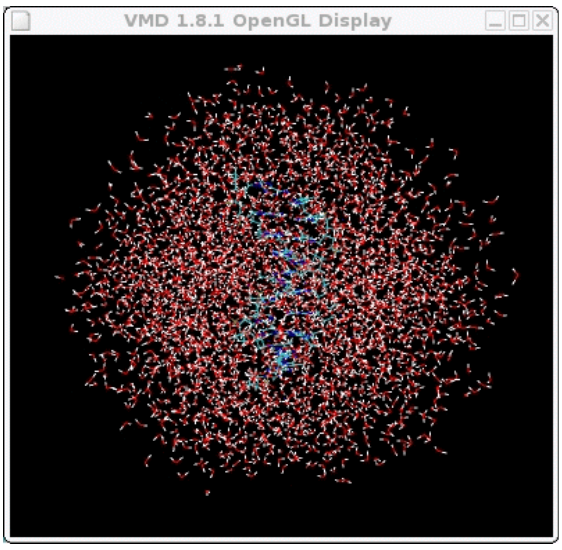
Run time ~ 1 hour

Lets take a quick look at the structure, in order to ensure it still sensible, before proceeding:

```
ambpdb -p ata_fsi.prmtop < ata_fsi.crd > ata_fsi.pdb

ambpdb -p ata_fsi.top < ata_fsi_min2.rst > ata_fsi_min2.pdb
```

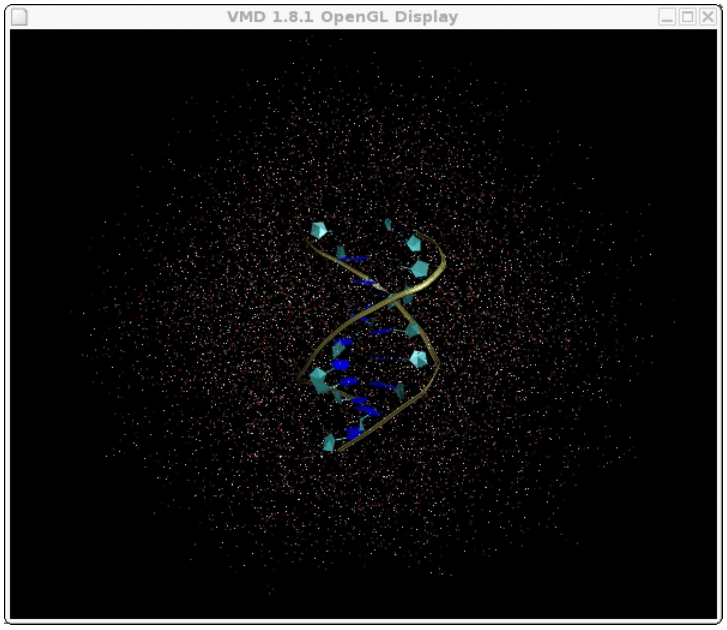
We shall use VMD to visualise the 2 structures (`ata_fsi.pdb, ata_fsi_min2.pdb`) and compare the structure of the DNA at the beginning and after our minimisation:

```
vmd
File -> New Molecule (load the first pdb ata_fsi.pdb)
```



Lets make the DNA stand out from the water better:

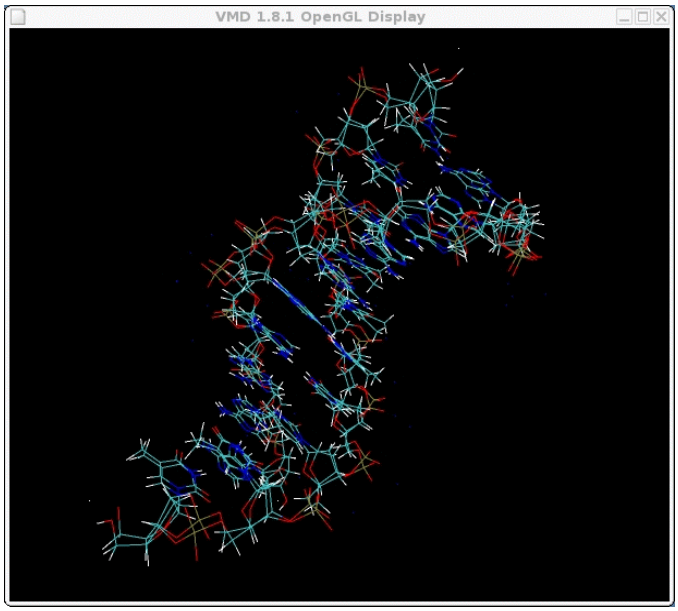
`Graphics -> Representations`
Click `"Create Rep"` This will create a second representation of our molecule. In the `"Selected Atoms"` box type: `all not water` (be sure to press the enter key) and then under `"Drawing Method"` select `"Ribbons"`. Next select the representation that still has the style `"Lines"` and change the drawing method for this to `"Points"`



Now, lets compare the starting structure and the minimised structure. This time we will instruct **VMD** not to draw the solvent. Quickest way to do this is to close **VMD** (`File->Quit`), and then re-load it. We will specify the first pdb file on the command line so we only need to load the second one manually:

`vmd ata_fsi.pdb`
`File -> New Molecule` Load the `ata_fsi_min2.pdb file.` Turn off the water to make it easier to compare the DNA's: Select `Graphics->Representations.` In the `Selected Atoms` box enter: `all not water` Click `"Apply"` and now under the `Selected Molecule` drop down box at the top of the `Graphical Representations` window select `molecule 0: ata_fsi.pdb` and repeat the process above.

You can see that the structure has changed slightly but there are no major problems so it would appear that the minimisation was successful. Note: The non-bonded blue dots are the sodium ions.



Molecular Dynamics (heating) with restraints on the solute

Now that we have successfully minimised our system the next stage in our equilibration protocol is allow our system to heat up from 0 K to 300 K. In order to ensure this happens without any wild fluctuations in our solute we will use a weak restraint, as we did in stage 1 of the minimisation, on the solute (DNA) atoms. Prior to AMBER 8 the recommended method for maintaining temperature was to use the Berendsen thermostat (NTT=1). This method is not very efficient at ensuring the temperature remains even across the system and so one would typically have to use NMR restraints in order to ensure that the heating occurred very gradually over a timescale of about 20 ps. This was essential in order to avoid problems with hot solvent cold solute etc. AMBER 8 now supports the new Langevin temperature equilibration scheme (NTT=3) which is significantly better at maintaining and equalising the system temperature. As a result the use of NMR restraints is no longer necessary. In fact it may even be possible to simply start our system at 300 K but we shall remain cautious and start from 0 K.

Note: you should use NTT=3 with care since while the Langevin temperature regulation scheme equilibrates the temperature better, as a method of maintaining a constant temperature during a production run it is probably not so good. Experience with using Langevin dynamics for temperature regulation has shown that the best approach in terms of obtaining accurate dynamics, would be to equilibrate the system initially with ntt=3, since it equilibrates temperature very well, and then switch to ntt=1 for the production phase. Alternatively, assuming the system is well equilibrated, you should be able to switch off the thermostat all together (ntt=3) and just run the explicit solvent calculation with no thermostat. This is true since the force field is conservative, hence you should be able to maintain constant energy for a very long time. However, errors in the integrator (the fact that you make a timestep approximation) and errors involved in using a cutoff will eventually lead to you bleeding energy over time. Hence a weak thermostat is often a good idea as it corrects for these errors.

The problem with using ntt=3 in a production calculation, is that it alters the dynamics of the system. Essentially the short term dynamics (fast dynamics) are radically altered and so if one is interested in obtaining information about the fast dynamics of a system during the production phase then ntt=3 cannot be used since it corrupts this information. Berendsen (ntt=1) does not do this.

Since these simulations are significantly more computationally expensive than the *in vacuo* and implicit solvent simulations it is essential that we try to reduce the computational complexity as much as possible. One way to do this is to use triangulated water, that is water in which the angle between the hydrogens is kept fixed. One such model is the TIP3P water model with which we solvated our system. When using such a water model it is essential that the hydrogen atom motion of the water also be fixed since failure to do this can lead to very large inaccuracies in the calculation of the densities etc. Since the hydrogen atom motion in our DNA is unlikely to effect its large scale dynamics we can fix these hydrogens as well. One method of doing this is to use the SHAKE algorithm in which all bonds involving hydrogen are constrained (`NTC=2`). This method of removing hydrogen motion has the fortunate effect of removing the highest frequency oscillation in the system, that of the hydrogen vibrations. Since it is the highest frequency

oscillation that determines the maximum size of our time step (1 fs for the *in vacuo* runs) by removing the hydrogen motion we can safely increase our time step to 2 fs without introducing any instability into our MD simulation. This has the effect of allowing us to cover a given amount of phase space in half as much time since we need only 50,000 steps to cover 100 ps in time as opposed to the 100,000 we required with a 1 fs time step.

Note, if your system is very inhomogeneous you may need to run short MD runs with smaller time steps, say 0.5 fs, before conducting long MD runs with a 2 fs time step. This approach allows the system to rapidly move out of an unfavourable geometry without wild oscillations in the energy.

So, stage 1, 20 ps of MD with weak positional restraints on the DNA. Here is the input file:

```
ata_fsi_md1.in

20 ps MD with res on DNA
 &cntrl
  imin = 0,
  irest = 0,
  ntx = 1,
  ntb = 1,
  cut = 10,
  ntr = 1,
  ntc = 2,
  ntf = 2,
  tempi = 0.0,
  temp0 = 300.0,
  ntt = 3,
  gamma_ln = 1.0,
  nstlim = 10000, dt = 0.002,
  ntpr = 100, ntwx = 100, ntwr = 1000
 /
Keep DNA fixed with weak restraints
10.0
RES 1 20
END
END
```

The meaning of each of the terms are as follows:

- IMIN = 0: Minimisation is turned off (run molecular dynamics)
- IREST = 0, NTX = 1: We are generating random initial velocities from a Boltzman distribution and only read in the coordinates from the inpcrd. In other words this is the first stage of our molecular dynamics. Later we will change these values to indicate that we want to restart a molecular dynamics run from where we left off.
- NTB = 1: Use constant volume periodic boundaries (PME is always "on" with AMBER 8 when NTB>0).
- CUT = 10: Use a cutoff of 10 angstroms.

- NTR = 1: Use position restraints based on the GROUP input given in the input file. In this case we will restrain the DNA with a force constant of 10 angstroms.
- NTC = 2, NTF = 2: SHAKE should be turned on and used to constrain bonds involving hydrogen.
- EMPI = 0.0, TEMP0 = 300.0: We will start our simulation with a temperature, derived from the kinetic energy, of 0 K and we will allow it to heat up to 300 K. The system should be maintained, by adjusting the kinetic energy, as 300 K.
- NTT = 3, GAMMA_LN = 1.0: The langevin dynamics should be used to control the temperature using a collision frequency of 1.0 ps-1.
- NSTLIM = 10000, DT = 0.002: We are going to run a total of 10,000 molecular dynamics steps with a time step of 2 fs per step, possible since we are now using SHAKE, to give a total simulation time of 20 ps.
- NTPR = 100, NTWX = 100, NTWR = 1000: Write to the output file (NTPR) every 100 steps (200 fs), to the trajectory file (NTWX) every 100 steps and write a restart file (NTWR), in case our job crashes and we want to restart it, every 1,000 steps.

We run this using the following command. Note, we use the `rst` file from the second stage of our minimisation since this contains the final minimised structure. We also use this as the reference structure with which to restrain the DNA to:

```
sander -O -i ata_fsi_md1.in -o ata_fsi_md1.out -p ata_fsi.top \
     -c ata_fsi_min2.rst -r ata_fsi_md1.rst -x ata_fsi_md1.traj \
     -ref ata_fsi_min2.rst &
```

Run time ~ 5.5 hours

Running MD Equilibration on the whole system

Now that we have successfully heated our system at constant volume with weak restraints on the DNA the next stage is to switch to using constant pressure, so that the density of our water can relax. At the same time, now we are 300 K we can safely remove the restraints on our DNA.

We will run this equilibration for 100 ps to give our system plenty of time to relax.

ata_fsi_md2.in

```
poly AT 10-mer: 100 ps MD
 &cntrl
  imin = 0, irest = 1, ntx =7,
  ntb = 2, pres0 = 1.0, ntp = 2,
  taup = 2.0,
  cut = 10, ntr = 0,
  ntc = 2, ntf = 2,
  tempi = 300.0, temp0 = 300.0,
  ntt = 3, gamma_ln = 1.0,
```

```
 nstlim = 50000, dt = 0.002,
 ntpr = 100, ntwx = 100, ntwr = 1000
/
```

The meaning of each of the terms are as follows:

- IMIN = 0: Minimisation is turned off (run molecular dynamics)
- IREST = 1, NTX = 7: We want to restart our MD simulation where we left off after the 20 ps of constant volume simulation. IREST tells sander that we want to restart a simulation, so the time is not reset to zero but will start at 20 ps. Previously we have had NTX set at the default of 1 which meant only the coordinates were read from the `rst` file. This time, however, we want to continue from where we finished so we set NTX = 7 which means the coordinates, velocities and box information will be read from a formatted (ASCII) restrt file.
- NTB = 2, PRES0 = 1.0, NTP = 1, TAUP = 2.0: Use constant pressure periodic boundary with an average pressure of 1 atm (PRES0). Isotropic position scaling should be used to maintain the pressure (NTP=1) and a relaxation time of 2ps should be used (TAUP=2.0).
- CUT = 10: Use a cut off of 10 angstroms.
- NTR = 0: We are no longer using positional restraints.
- NTC = 2, NTF = 2: SHAKE should be turned on and used to constrain bonds involving hydrogen.
- TEMPI = 300.0, TEMP0 = 300.0: Our system was already heated to 300 K during the first stage of MD so here it will start at 300 K and should be held at 300 K.
- NTT = 3, GAMMA_IN = 1.0: The Langevin dynamics should be used to control the temp using a collision frequency of 1.0 ps^{-1}.
- NSTLIM = 500000, DT = 0.002: Run a total of 50,000 step with a time step of 2 fs per step. This is possible since SHAKE is on. Total simulation time will be 100 ps.
- NTPR = 100, NTWX = 100, NTWR = 1000: Write NTPR every 100 steps (200 fs), the trajectory file (NTWX) every 100 steps, and the rst file (NTWR) every 1000 steps.

We run this using the following command. Note, we use the `rst` file from the first stage of our MD since this contains the final MD frame, velocities and box information.

```
sander -O -i ata_fsi_md2.in -o ata_fsi_md2.out -p ata_fsi.top \
     -c ata_fsi_md1.rst -r ata_fsi_md2.rst -x ata_fsi_md2.traj &
```

Note: This simulation may take considerable amount of time to run (about 5 times more than the 20 ps run above).

Run time ~ 30 hours

Analysing the results to test the equilibration

Now that we have equilibrated our system it is essential that we check that equilibrium has successfully been achieved before moving on to running any production MD simulations in which we would hope to learn something new about our molecule.

There are a number of system properties that we should monitor to check the quality of our equilibrium. These include a) Potential, Kinetic, and Total Energy, b) Temperature, c) Pressure, d) Volume, e) Density, and f) RMSd.

The following steps will illustrate how to extract this data and plot it and also discuss what we expect to see for a successful equilibration.

Analysing the output files

In this section we will look at the system properties that can be extracted from the data written to the mdout files during our 120ps of MD simulation (`ata_fsi_md1.out`, `ata_fsi_md2.out`). This includes the potential, kinetic and total energies, the temperature, pressure, volume and density.

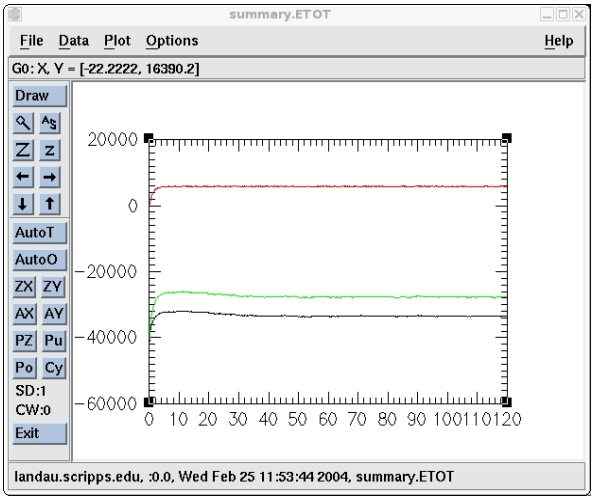
Lets start by extracting the various properties from our two output files. For this we will use the `process_mdout perl` script that was introduced earlier. This script can be given any number of files to process and it will append the results to single summary files (this assumes that the location of `process_mdout.perl` is in your path, it also assumes that you are currently in polyatA):

```
mkdir analysis
mv process_mdout.perl analysis
cd analysis
process_mdout.perl ../ata_fsi_md1.out ../ata_fsi_md2.out
```

This should give us a series of summary files in our analysis directory. Lets plot some of these to see if we have successfully reached an equilibrium. First of all the energies, potential, kinetic and total (= potential + kinetic):

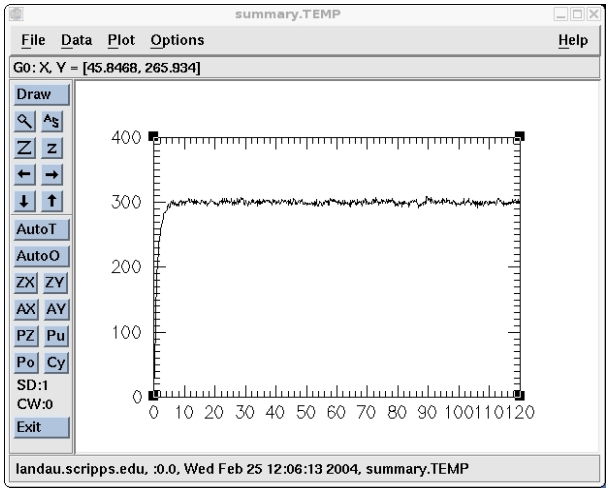
```
xmgr summary.EPTOT summary.EKTOT summary.ETOT
```

Here the red line, which is positive, is our kinetic energy, the black line is the potential energy (negative) and the green line is our total energy. The important points to note with this plot is that all of the energies increased during the first few ps, corresponding to our heating from 0 K to 300 K. The kinetic energy then remained constant for the remainder of the simulation implying that our temperature thermostat, which acts on the kinetic energy, was working correctly. The potential energy, and consequently the total energy since total energy = potential energy + kinetic energy, initially increased and then plateaued during the constant volume stage (0 to 20 ps) before decreasing as our system relaxed when we switched off the DNA restraints and moved to constant pressure (20 to 40 ps). The potential energy then levelled off and remained constant for the remainder of our simulation (40 to 120 ps) indicating that the relaxation was completed and that we reached an equilibrium.



Next we plot the temperature:

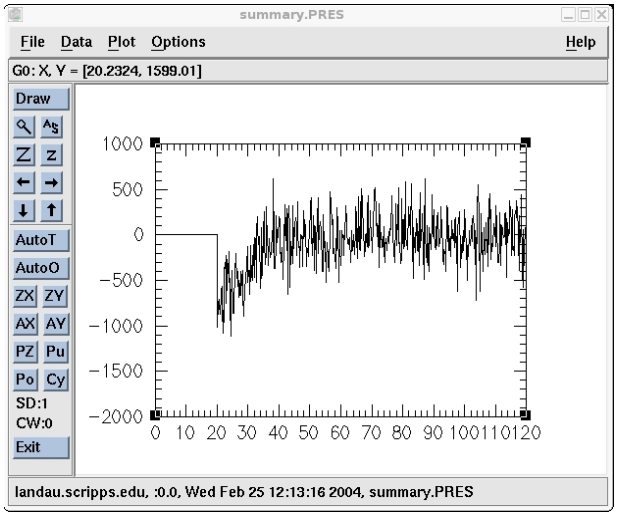
```
xmgr summary.TEMP
```



Here we see the expected behaviour, our system temperature started at 0 K and then increased to 300 K over a period of about 5 ps. The temperature then remained more or less constant for the remainder of the simulation indicating the use of Langevin dynamics for temperature regulation was successful.

Next the pressure:

```
xmgr summary.PRES
```



The pressure plot is slightly different than the previous plots. For the first 20 ps the pressure is zero. This is to be expected since we were running a constant volume simulation in which the pressure wasn't evaluated. At 20 ps we switched to constant pressure, allowing the volume of the box to change, at which point the pressure drops sharply becoming negative. The negative pressures correspond to a "force" acting to decrease the box size and the positive pressures to a "force" acting to increase the box size. The important point here is that while the pressure graph seems to show that the pressure fluctuated wildly during the simulation the mean pressure stabilised around 1 atm after about 50 ps of simulation. This is sufficient to indicate successful equilibration.

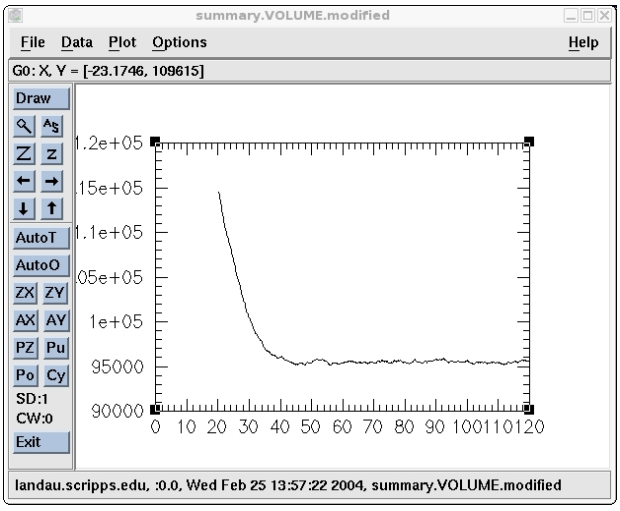
Next we will consider the volume. Unfortunately we can simply plot the summary.VOLUME file since the first 20 ps do not contain any volume information since during constant volume simulations the volume of the box is not written to the output file. So, we need to remove the first 101 lines of the file which you can use your favorite text editor. While most of you will use nedit or jot, this is a case where vi is faster. Try this (first copy summary.VOLUME to summary.VOLUME.mod:

```
vi summary.VOLUME.mod
d100
:wq!
```

FYI, the command `vi` starts the vi editor, `d100` deletes the current + the next 100 lines, `:wq!` tells `vi` a) your done editing (‘`:`’), b) to write the file (‘`w`’), c) to quit `vi` (‘`q`’) and d) do it now (‘`!`’).

And then plot it

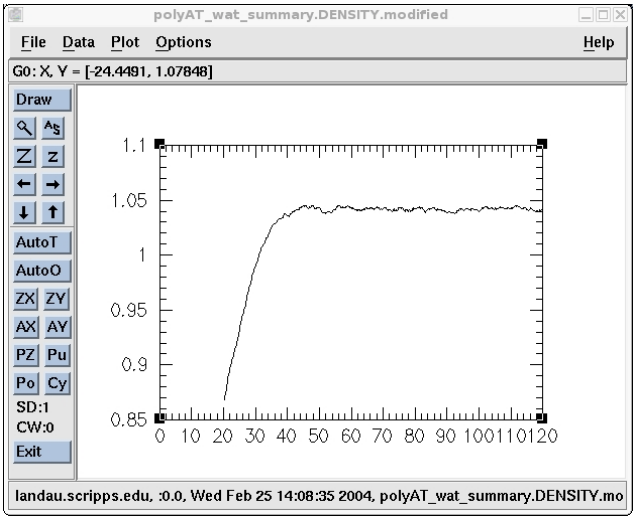
```
xmgr summary.VOLUME.mod
```



Notice how the volume of our system (in angstroms^3) initially decreases as our water box relaxes and reaches and equilibrated density, and thus volume. The smooth transitions in this plot followed by the oscillations about a mean value suggest our equilibration has been successful. Finally, lets take a look at the density, we expect this to mirror the volume. We have the same problem with this summary file as we did with the volume since the density is not written to the

output during constant volume simulations.

Here is the modified file with the top 101 lines removed (summary.DENSITY.modified).



Analysing the trajectory

Now we have analysed our output files and seen that the main system properties suggest that our equilibration has been successful the next step is to consider the structures. Have they remained reasonable? One useful measure to consider is the root mean square deviation (RMSd) from the starting structure. We can use `ptraj` to calculate this for us as a function of time. In this situation we are interested in the backbone of our DNA so we shall consider just the main backbone atoms, P, O3*, O5* C3*, C4* and C5* (P, O3', O5'. C3', C4' and C5' in our `top`). We will conduct a standard (not mass weighted) RMS fit to the first structure of our MD (the final structure from the minimisation). Note, the time is set to 0.2 this time around since although we wrote to the trajectory file every 100 steps the time step was 2 fs - hence 100 steps = 0.2 ps. Here is the input file for `ptraj`:

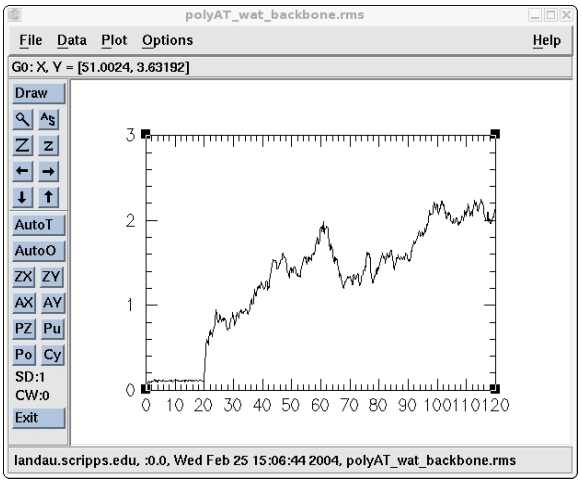
```
ata_fsi_calc_backbone_rms.in

trajin ata_fsi_md1.traj
trajin ata_fsi_md2.traj
rms first out ata_fsi_backbone.rms @P,O3,O5',C3',C4',C5' time 0.2
```

Lets run `ptraj` to calculate the RMSd of the DNA backbone atoms (P, O3*, O5* C3*, C4* and C5*):

```
ptraj ata_fsi.top < ata_fsi_calc_backbone_rms.in
```

Here is the output file (polyAT_wat_backbone.rms).

```
xmgr ata_fsi_backbone.rms
```



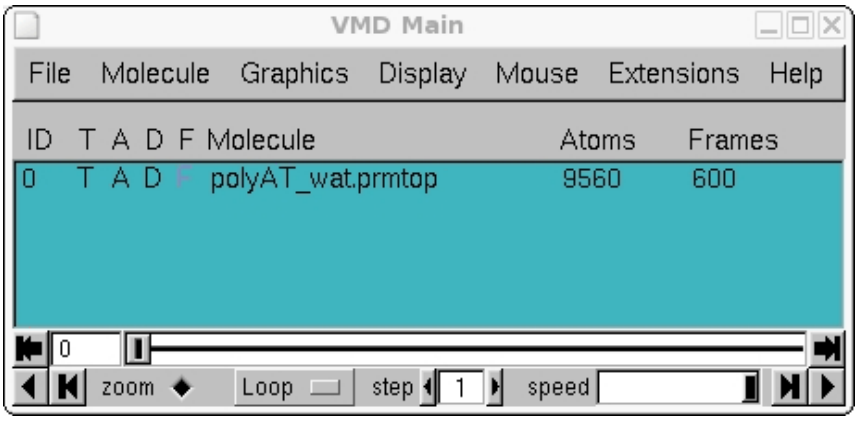
Here we see that the RMSd of the DNA backbone atoms remained low for the first 20 ps, this was due to the restraints. Upon removing the restraints the RMSd shot up as the DNA relaxed within the solvent. After that our RMSd is fairly stable with no wild oscillations.

Viewing the trajectory

Lets take a look at the trajectory using **vmd**. First of all we will do this without re-imaging back to the primary box. We will then use ptraj to re-image our trajectory and then look at it again. So, first of all lets load both trajectory files into **vmd**:

```
vmd
```

Step 1 - Load the topology file (ata_fsi.top) remembering to select “parm7"
Step 2 - Load the two trajectory files (ata_fsi_md1.traj, ata_fis_md2.traj). Note this time you need to select CRDBOX as the file type since the simulations are now periodic boundry runs with box information stored in the trajectory file. Finally, rewind to the first frame and press play. It appears like the water box is breaking down and the waters are boiling off into space. This would indeed be true if we were *not* using periodic boundaries. However, as mentioned above the atom coordinates written to the trajectory files are those of the original atom, even if it migrates outside of the original box and is replaced by one of its images. We can "fix" this by re-imaging our trajectory as we will now do.



Re-imaging the trajectory back into the primary box

Lets use **ptraj** to re-image our trajectory files so that only the atoms in the primary box are shown. Here is the **ptraj** input file:

```
ata_fsi_reimage.ptraj

trajin ata_fsi_md1.traj
trajin ata_fsi_md2.traj
trajout ata_wat_md_imaged.traj
center :1-20
image familiar
go

ptraj ata_fsi.top < ata_fsi_reimage.ptraj
```

This will append the md2 trajectory file to the md1 file, change the trajectory such that the center of geometry of residues 1 to 20 is placed at the origin and then image all of the coordinates outside of the primary box back to their images inside the box. The familiar keyword makes sure **ptraj** writes out the box as a familiar truncated octahedron as opposed to the default, triclinic imaging.

You can now open this trajectory file in VMD where you will find our flying water droplet is much more intact. Have a play with the representations in vmd, remove the water and look at just the DNA etc. Observe how the dynamics of the DNA molecule change after the first 20ps when the restraints are switched off. Also, observe how much more stable our DNA chain is now than it was in the *in vacuo* simulations.

Using **moilview** to look at a movie

This is an alternative to VMD. This section is optional. Knowing how to do this can be helpful, but what you've learned with VMD should suffice for now.

Next, it's time to take a look at a movie of the trajectory. This was done in the previous lab but just in case you've forgotten:

Moilview is a program for looking at Moil, AMBER and pdb files. In addition to displaying structures, one can also look at movies of a trajectory and perform some rudimentary trajectory analysis.

To view a movie of this trajectory, start up moil-view

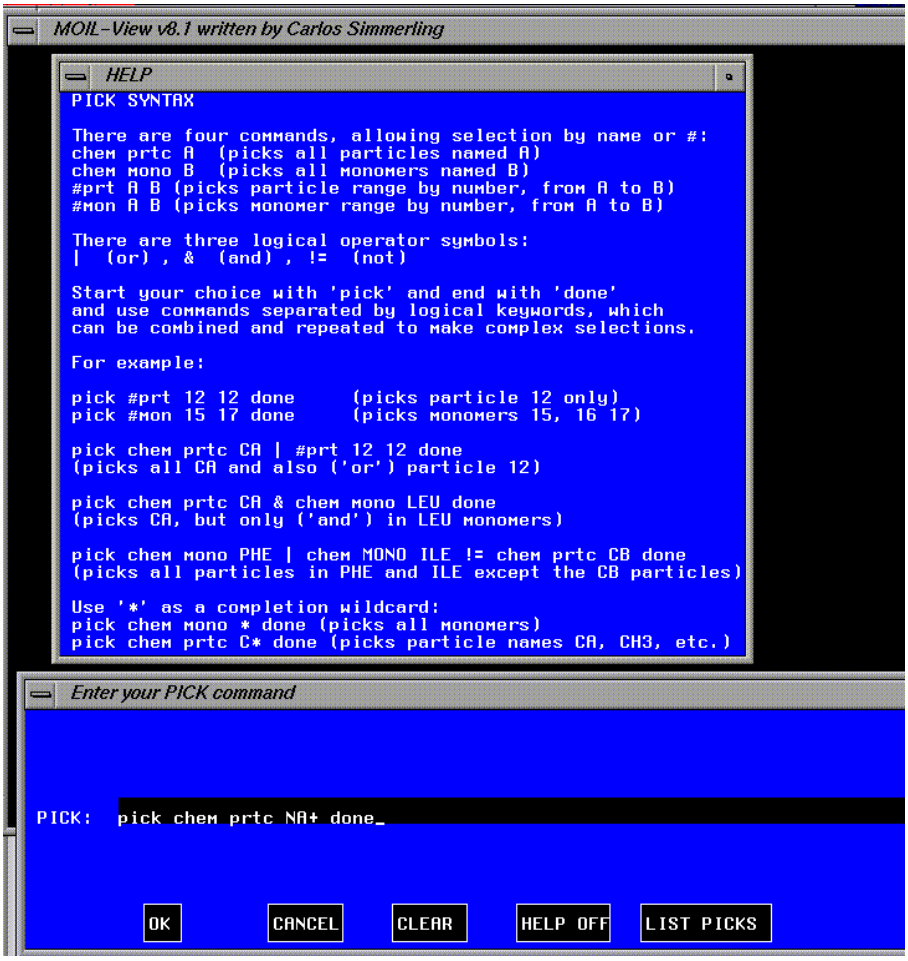
```
moilview
```

With the right button, chose the "file" menu and "Read 1st coordinates". Select "AMBER" format and then specify the file `ata_fsi.crd` (the amber coordinate file) at the dialog. When asked, say that "no", you do not want to use bonds as connectivity, but "yes" you would like to

load a connectivity file. Again specify the "AMBER" format and load up `ata_fsi.top` (the amber topology file). The structure should appear.

Next, since the water molecules make it difficult to see what's going on, right click and pick "Color Options" and then pick "no waters".

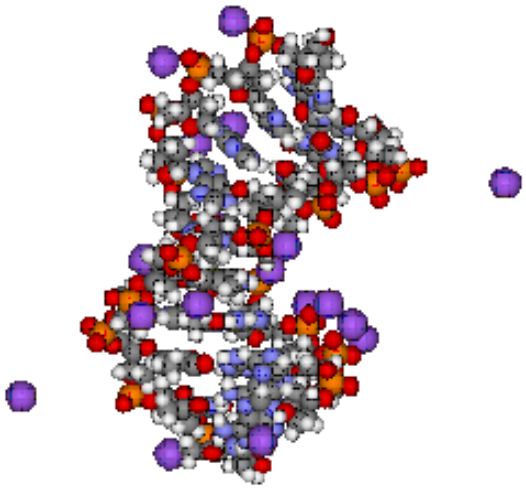
To look at a movie of the trajectory, first shrink the structure so that about 10% of the full screen area is used and move it to the center of the screen. Then, use the right mouse button and chose the 'Dynamics' menu and 'View trajectory' submenu. The trajectory we are looking at was run with water, so chose 'AMBER PBC' (PBC stands for partial boundary constrant) as the type and load up `ata_mdp.traj`. Input values, for the next dialog that pops up, a delay of 5 (5 works well to begin with) start (1) and end frame (100) (end frame information is given in the output from **carnal**) and for continuous answer yes.



To stop the dynamics, hit the right button.

If you neglected to remove the waters, it's not to late, just hide the water by going into the main menu (right button), drag 'Color options' and choose 'No waters'. Additionally, those ions are kind of hard to visualize, so lets make them solid spheres. Chose main menu 'Objects are' and submenu 'Mix lines/spheres' which will pop up a dialog box which will prompt you for a

selection of atoms which you want as spheres. We will choose all the atoms named Na+. The selection format is kind of tricky; turn help on for more information. Here is what it will look like and what the pick string looks like (The text in the information box should read 'pick chem prtc Na+ done' and don't forget the done. Note that the Na+ in the screen shot is wrong. It should be Na+ and not NA+):



RMSd and Trajectory Analysis

There are several programs available to perform analysis of an energy minimized structure or a trajectory analysis. These include **moilview**, **vmd**, and **dials_and_windows (aka dandw, MDTools)**. You should analyze the trajectory with **dials_and_windows** though you will probably need to look back at the previous lab.

Your are free to explore using **moilview**, now. Documentation for **moilview** is in the molecular modeling lab and I (PMG) have had some success using the tools for trajectory analysis available within **moilview**.

Summary

You have now covered the techniques necessary to create an initial input structure for a small DNA 10-mer, modify the pdb file to make it compatible with **LEaP** and then load this into **xleap**. We have covered how to charge neutralise this structure, solvate it in water and then create the necessary input files for running MD using **sander**. What to do when you have residues that are not in the default **AMBER** templates will be covered in later tutorials.

You should now know how to use these input files to run simulations *in vacuo*, in implict solvent and in explicit solvent. The concept of minimisation and equilibration has been covered and finally you now know how to perform simple analysis of the results and view trajectory files.

The final stage of this tutorial is to use these newly learned techniques in a practical example. One might ask the question: What happens if I start the simulations using the structure for A-DNA? In the last stage of this tutorial we shall try just that.

Questions

1) In this lab you have built and solvated a ds DNA, $d(A)_{10}{\cdot}d(T)_{10}$ in the A DNA conformation and then taken it through a series of minimizations and dynamics. Go through the out files and find the values of the total energy of the systems at the following steps:

   a) After the fixit minimization.
   b) After the minimization that relaxed the waters and ions.
   c) After the minimization that relaxed the entire system.
   d) After the 20 ps equilibration.
   e) After the 100 ps MD run.

Consider these numbers. What trends do you see, if any? Are these energy numbers found for (a)-(e) directly comparable? Why or why not?

2) Prepare and turn in plots analogous to those shown on page 17-20.

3) Prepare and turn Dials_and_Windows plots of your final 100 ps trajectory.

4) Suppose you need to generate a topology, coordinate, and pdb file that has been stripped of the waters but still containing the sodium ions that corresponded to the rst file generated at the end of the 100 ps MD run. Describe how you would do this. Use the following names for your new topology file `ata_md_prod_nowat.top`, for your new coordinate file call it `ata_md_prod_nowat.crd` and the `pdb` file should be called `ata_md_prod_nowat.pdb`. Of course, you should check and make sure your proposed method works and turn in a picture of your DNA, stripped of waters and showing the sodium ions. Please, on a white background.

Appendix

Extra-Credit

The DNA you ran dynamics on was in the A DNA conformation. In water, this DNA will switch to the B DNA conformation if the dynamics is run for a sufficiently long period of time. The trajectory file, a rst file, and a topology file are located in

/disk02/usr/people/amber_ans/amberlab9_ec

on Amber, Frodo and Iris and in:

/usr/people/amber_ans/amberlab9_ec

on Opal. Also in this directory is a file that when plotted with xmgr displays the RMSd to the starting structure over the course of the trajectory. Copy the first three of these files to your own directory (perhaps somewhere under the polyatA directory).

```
ata_1820.top
ata_1820.rst
ata_1820.rmsd
ata_1820.traj
```

The trajectory file (`ata_1820.traj`) is quite large and rather than consume a lot of disk space by making multiple copies, create a link to the trajectory file with the following command:

on amber, frodo or iris use

```
ln -s /disk02/usr/people/amberlab/amberlab9/amberlab9_ec/ata_1820.traj ata_md.traj
```

and on opal use:

```
ln -s /usr/people/amberlab/amberlab9/amberlab9_ec/ata_1820.traj ata_md.traj
```

This will create a soft link to the trajectory. When you reference ata_md.traj, it will follow the link you have created to the trajectory file. Note that the trajectory file is stripped of waters as is the top and rst files. However, there is box information and it is a periodic system. Analyze the trajectory using dials_and_windows. When does the structure convert from A to B DNA? The RMSd plot suggests when this might happen. Do the dials_and_windows plots confirm this?

What helicoidal parameters support the switch from A to B? Select a structure from the trajectory file that is clear A DNA and one that is clearly B DNA and provide plots of these structures.