Amber: Building Small Molecules and Minimizing their Energy Lab 3

Table of Contents

Note: Portions of this lab were taken from the Amber website. Most notably is the section describing Antechamber. My thanks to David Case.

Objectives

In this lab you will build several small molecules and run them through **Sander**, **Amber's** energy minimizer (both molecular mechanics and molecular dynamics). You will also solvate a molecule and prepare the solvated molecule for dynamics.

Outline

Setup: Login
Setup up required directories
Copy required files
Prepare control files for minimization (scripts)

Leap: Build methane
Solvate methane
Write molecular mechanics and molecular dynamics files for methane
Build butane in the fully eclipsed (*syn*) , *anti*-, and *gauche* forms
Write butane files for molecular mechanics
Build the diequatorial and diaxial forms of dimethylcyclohexane
Write dimethylcyclohexane files for molecular mechanics

Antechamber Preparation of prepin files
Preparation of frcmod files
Use of prepin and frcmod files in leap

Sander:Run **Sander** on butanes
Run **Sander** on dimethylcyclohexanes
Compare energies of the butanes and dimethylcyclohexanes
View minimized structures

Setup

Login and Setup

If you are on a Unix machine, log into your account. If you are on taz or kali, you will need to connect to sonny or tambor using the command:

```
ssh -X username@sonny.hsc.wvu.edu
```

```
or

ssh -X username@tambor.hsc.wvu.edu
```

The 'X' in the command is uppercase and username is your username on the corresponding machine (sonny or tambor). The problem is that xleap is not yet functional on the Linux machines. This lab gives instructions for **Amber** version 8 or 9 (**Amber8 or 9**) , which are installed on the Unix machines. After you have logged into your account on (directly or remotely), create a new subdirectory in your home directory called `hydrocarbons` and change to this directory.

```
mkdir ./hydrocarbons
cd hydrocarbons
```

Then, copy the files located in ../amberlab/amberlab3 to your home directory using the command:

```
cp ../amberlab/amberlab3/* .
```

And then start **xleap**.

```
xleap
```

Building small molecules in Leap

Methane

Methane is a simple molecule and will serve as an example. The other molecules you will build will be done is a similar fashion.

Begin by typing at the > prompt:

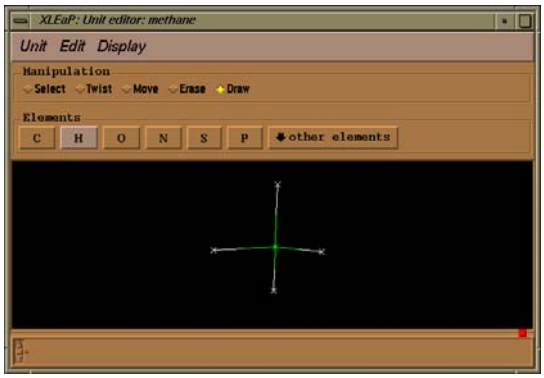
```
edit mth
```

This will start up the Unit Editor.

Once the Unit Editor pops up, make sure 'C' is selected under element, click on the diamond before 'Draw' and move the cursor to somewhere in the black window. Click with the left mouse button once to draw a carbon. Click on the diamond before 'Select', select 'H' under element, and then select 'Draw.' Draw the four hydrogens out from the carbon by first clicking (left button) on the carbon atom, holding the button down, and moving away from the carbon. Then release the left mouse button. It doesn't matter how long you draw each bond. They can be long, short, or

all different lengths. Use 'Erase' mode if you make a mistake. Don't worry about the geometry - that will be cleaned up later.

Methane in the Unit Editor



Now pull down 'Unit / Build' and the geometry is complete. Even simpler would have been to just draw the carbon and choose 'Unit / Add H & Build', but it's fun to draw the hydrogens, at least at first.

Hold down the center and right buttons and push forward/back to zoom in/out. The middle button alone rotates, the right button translates, and the spacebar recenters the molecule.

To complete the model, we must add atom types and charges. While we are at it, we'll add the perturbed parameters for 'disappearing' the molecule as we will when we do molecular dynamics. Choose 'Select' mode and snap a box around the molecule. This is done by placing the cursor at what will be a box that the molecule will fit inside of, press and hold the left mouse button, move the mouse diagonally to create the box, and then release the mouse button. You will see the molecule change to a magenta. Then pull down 'Edit / Edit selected atoms' and fill in the table as shown below. Clicking on a cell 'activates' it, then you can type in it; you can also copy from one cell (press the left mouse button and highlight the text to be copied) and paste into a cell (using the middle mouse button) without activating it. When done, pull down 'Operations / Check Table'. The message area will report if you have made any mistakes. The message window can be enlarged, at the expense of the table, by a left click and hold with the cursor on the red square at the lower right hand corner of the box.

XLEaP: Edit selected atoms: methane

Table Operations

Table Editor: Check table...
Table Editor: There are no mistakes in the table.

NAME	TYPE	CHARGE	ELEMENT	PERTURB	PERT.name	PERT.type	PERT.charge
C1	CT	-0.46410	C	true	DC1	DC	0.0000
H2	HC	0.116025	H	true	DH2	DH	0.0000
H3	HC	0.116025	H	true	DH3	DH	0.0000
H4	HC	0.116025	H	true	DH4	DH	0.0000
H5	HC	0.116025	H	true	DH5	DH	0.0000

(Don’t worry too much about the entries in the table right now. Briefly, the entries for type refer to their definition in the **Amber** force field, the CHARGE column gives the atom charges and were calculated using the **Amber** module **resp**, The 'PERT.name' is not important; the PERT.type will be used for selecting the end-state VDW, bond, and angle parameters.) Now pull down 'Table / Save and quit', and deselect the atoms by holding down the Shift key while clicking the left mouse button anywhere in the blackness, away from the atoms (this is more for cosmetic purposes). Now pull down 'Unit / Close' and save this model for future reference from the main window by entering:

```
saveoff mth methane.lib
```

In a later session, this molecule can be loaded by

```
loadoff methane.lib
```

(Note: When you ask for a list of possible commands by typing ‘help’ you will see some commands with both upper and lower case letters. It is not necessary to use uppercase. The use of uppercase is only to help you remember the commands.)

You can solvate (‘dissolve in water’) the molecule by typing the following commands.

```
x = copy mth
solvatebox x TIP3PBOX 10
edit x
```

After you solvate the molecule you’ll notice what appear to be a bunch of triangles surrounding your methane molecule. These are the waters - called TIP3 waters. To reduce computational time, the hydrogens of the water molecules are bonded and makes them appear as triangles.

Now, save 'parm' files for dynamics, perturbation with no bond shrinkage, and perturbation with bond shrinkage. When we get to using **Amber** for molecular dynamics, we will use these files:

```
saveamberparm x mebox.top mebox.crd

fmod1 = loadamberparams me_1.frcmod
saveamberparmpert x me_p1.top xxx

fmod2 = loadamberparams me_2.frcmod
saveamberparmpert x me_p2.top xxx
```

(The coordinate file, normally given the extension `.crd`, has the throwaway name `xxx` because the one resulting from the dynamics equilibration will be used for each perturbation.)

Butane - Eclipsed, gauche and anti

You should still be in **xleap**. To start drawing the butane structures enter the commands shown:

```
edit butane
```

This will put you in the unit editor. Select the draw mode and draw three line segments as follows: Left click and hold, draw a line at about a 45^{o} angle (downward) and release. Left click where you left off and hold, draw a horizontal line and release. Left click and hold, draw a line at about a 45^{o} angle (downward) and release. You know have the skeleton of butane. Click on 'Unit/Add H and build' and release. Bond lengths and angles will be cleaned up and hydrogens added. The molecule will also appear to shrink but it didn't, just zoom to resize. Exit the Unit Editor and enter:

```
savepdb butane tbutane.pdb
```

and finally, quit out of xLeap.

Antechamber: In the previous section parameters were provided for methane. As molecules become larger and the range of atoms increases, this process can become quite tedious. In addition, there is the matter of developing the parameters required and these parameters are required in order to write topology and coordinate files for simulations of organic molecules.

In Amber 7 the program Antechamber was added and is designed to be used with the "general AMBER force field (GAFF)[1]". This force field has been specifically designed to cover most pharmaceutical molecules and is compatible with the traditional AMBER force fields in such a way that the two can be mixed during a simulation. Like the traditional AMBER force fields, GAFF uses a simple harmonic function form for bonds and angles but unlike the traditional protein and DNA orientated AMBER force fields the atom types used in GAFF are much more general such that they cover most of the organic chemical space. The current implementation of the GAFF force field consists of 33 basic atom types and 22 special atom types. The charge methods used can be HF/6-31G* RESP or AM1-BCC[2].

By design, GAFF, is a complete force field (so that missing parameters rarely occur), it covers almost all the organic chemical space that is made up of C, N, O, S, P, H, F, Cl, Br and I. Moreover, while GAFF is totally compatible with the AMBER macromolecular force fields it should prove to be an useful molecular mechanical tool for rational drug design. Especially in binding free energy calculations and molecular docking studies.

The Antechamber tool set is designed to allow the rapid generation of topology files for use with the AMBER simulation programs. Unlike the previous tutorial where we manually assigned our atom types and parameters here we will allow antechamber to do this automatically for us using GAFF. With Antechamber, one may solve the following problems:

1) Automatically identify bond and atom types
2) Judge atomic equivalence
3) Generate residue topology files
4) Find missing force field parameters and supply reasonable suggestions

You should note, however, that Antechamber is not a replacement for due dilligence. You should always closely examine the atom types that Antechamber assigns and verify to yourself that the choices are reasonable. You should never use scientific software in a "Black Box" approach! Antechamber is a big help for creating new molecules that can be read by xLeap but it is far from perfect.

We shall use Antechamber to assign atom types to this molecule and also calculate a set of point charges for us. Antechamber is the most important program within the set of Antechamber tools. It can perform many file conversions and can also assign atomic charges and atom types. As required by the input antechamber executes the following programs (all provide with AMBER 8): *divcon*, *atomtype*, *am1bcc*, *bondtype*, *espgen*, *respgen* and *prepgen*. It will also generate a series of intermediate files (all in capital letters).

Creating topology and coordinate files for Butane

We shall use the Antechamber tools with xLeap to create topology and coordinate files for butane beginning with the *trans* butane you just build.

We shall use Antechamber to assign atom types to this molecule and also calculate a set of point charges for us. Antechamber is the most important program within the set of Antechamber tools. It can perform many file conversions and can also assign atomic charges and atom types. As required by the input antechamber executes the following programs (all provide with AMBER 8): *divcon*, *atomtype*, *am1bcc*, *bondtype*, *espgen*, *respgen* and *prepgen*. It will also generate a series of intermediate files (all in capital letters).

Let's try using antechamber on our `tbutane.pdb` file. To create the "prepin" file we require to define a new unit in leap we simply run the following command:

```
antechamber -i tbutane.pdb -fi pdb -o tbutane.prepi -fo prepi -c bcc -s 2 -at amber -rn ANE
```

Here the `-i tbutane.pdb` specifies the name of the 3D structure file and the `-fi pdb` tells antechamber that this is a pdb format file (we could easily have used any number of other supported formats including Gaussian Z-Matrix [gzmat], Gaussian Output [gout], MDL [mdl], amber Restart [rst], Sybyl Mol2 [mol2]). The `-o tbutane.prepin` specifies the name of our output file and the `-fo prepi` states that we want the output file to be of amber PREP format (this is an internal format supported by Leap). The `-c bcc` option tells antechamber to use the BCC charge model in order to calculate the atomic point charges while the `-s 2` option defines the verbosity of the status information provided by antechamber. In this case we have selected verbose output (2). Finally, `-at amber` sets the atom type names to amber format and the `-rn`

option gives the molecule the name ANE, best if it is three letters long and in caps as is typical of a .pdb file.

So, go ahead and run the above command. The screen output should be as follows:

ANTECHAMBER_BOND_TYPE.AC -f ac -j full

Running: /disk02/usr/local/amber8/exe/atomtype -i ANTECHAMBER_AC.AC0 -o ANTECHAMBER_AC.AC -p gaff

Total number of electrons: 34; net charge: 0

Running: /disk02/usr/local/amber8/exe/divcon

Running: /disk02/usr/local/amber8/exe/am1bcc -i ANTECHAMBER_AM1BCC_PRE.AC -o ANTECHAMBER_AM1BCC.AC -f ac -p /disk02/usr/local/amber8/dat/antechamber/BCCPARM.DAT -s 2 -j 1

Running: /disk02/usr/local/amber8/exe/atomtype -f ac -p bcc -o ANTECHAMBER_AM1BCC.AC -i ANTECHAMBER_AM1BCC_PRE.AC

Running: /disk02/usr/local/amber8/exe/atomtype -i ANTECHAMBER_PREP.AC0 -o ANTECHAMBER_PREP.AC -p gaff

Running: /disk02/usr/local/amber8/exe/prepgen -i ANTECHAMBER_PREP.AC -f int -o tbutane.prepi -rn "ANE " -rf molecule.res

You should also get a whole series of files written to your directory.

ANTECHAMBER_AC.AC	ANTECHAMBER_PREP.AC	divcon.out
ANTECHAMBER_AC.AC0	ANTECHAMBER_PREP.AC0	tbutane.pdb
ANTECHAMBER_AM1BCC.AC	ATOMTYPE.INF	tbutane.prepi
ANTECHAMBER_AM1BCC_PRE.AC	NEWPDB.PDB	
ANTECHAMBER_BOND_TYPE.AC	PREP.INF	
ANTECHAMBER_BOND_TYPE.AC0	divcon.in	

The files in CAPITALS are all intermediate files used by antechamber and are not required here. You can safely delete them. These files are not deleted by default since they may be of interest if things didn't work correctly. The divcon.xxx files are input and output from the divcon quantum mechanics code used by Antechamber to calculate the atomic point charges. We are not interested in the data here except to check that the divcon calculation completed successfully:

```
divcon.out
```

```
*****************************************************************************
```

```
 *                                                                            *
 *                          DIVCON 99a                                        *
 *                                                                            *
 *                                          $Revision: 1.1 $ (*)              *
 *                                                                            *
 ******************************************************************************

                                       (*) COMPILED WITH DIRECTIVES:
                                           MPI_IS_ON            NOT DEFINED
                                           MPI_NOMC             NOT DEFINED
                                           CUTREPUL_IS_ON       NOT DEFINED
                                           LARGE_MEMORY_MC      NOT DEFINED
                                           MEMORY_OVERLAP           DEFINED
                                           STEWART_INTEGRALS        DEFINED
                                           SCRF_IS_ON           NOT DEFINED
                                           PARAM_IS_ON          NOT DEFINED


CARTESIAN AM1 STANDARD DIRECT OPT=BFGS XTEST=0.0001 CHARGE=0

created by wmopcrt()

 CARTESIAN      - CARTESIAN COORDINATE FORMAT SELECTED
 AM1            - AM1 HAMILTONIAN TO BE USED
 STANDARD       - STANDARD CLOSED-SHELL CALCULATION (NO D&C)
 DIRECT         - DO NOT STORE 2-ELECTRON INTEGRALS IN SCF CALCULATIONS
 OPT=BFGS       - CARRY OUT A BFGS GEOMETRY OPTIMIZATION
 XTEST=0.0001   - GEOMETRY OPTIMIZATION COORDINATE CHANGE CRITERION = 0.00010
 CHARGE=0       - A NET CHARGE OF     0.00 TO BE PLACED ON SYSTEM

 NUMBER OF ATOMS                =    14
 TOTAL NUMBER OF BASIS FUNCTIONS =    26
 TOTAL NUMBER OF ELECTRONS       =    26

 DEFAULT SETTINGS FOR CALCULATION
 --------------------------------

 - D&C TEMPERATURE OF 1000.00 K WILL BE USED
 - NO TIME LIMIT ON CALCULATION

 ETEST  =  0.002000
 XTEST  =  0.000100
 GNTEST =  0.810093
 GTEST  =  0.500000

 NUMBER OF VARIABLES=    42

 UPDATING PAIRLIST
 NPAIRS =       91

  CYCLE =     0         TIME =      0.060  ENERGY =      38.618553
  GNORM =    437.349  GRDMAX =  119.3808   GRDAVR =   56.4854
 DELTAE =    0.000000 DELTAX = 0.000000

.
.
.

  CYCLE =    45         TIME =      0.077  ENERGY =     -31.090009
  GNORM =      0.226  GRDMAX =    0.0986   GRDAVR =    0.0274
 DELTAE =   -0.000023 DELTAX = 0.000065
 ENERGY TEST PASSED
 COORDINATE TEST PASSED
 GRADIENT NORM TEST PASSED
 GRADIENT COMPONENT TEST PASSED

 -- GEOMETRY OPTIMIZED --


 FINAL QUANTITIES:
 -----------------

 NUMBER OF FULLY DIAGONALIZED SUBSYSTEMS:      1
```

```
NUMBER OF FROZEN SUBSYSTEMS:                 0
NUMBER OF SCF CALCULATIONS =     51

ELECTRONIC ENERGY     =            -2454.89885480 EV
CORE-CORE REPULSIONS  =             1804.08725391 EV
TOTAL ENERGY          =             -650.81160089 EV
HEAT OF FORMATION     =              -31.09000923 KCAL/MOL
FERMI ENERGY          =                0.00000000 EV

TIMINGS:
-------
* NUMBER OF DIAGONALIZATIONS:     2
* SCF CYCLE: TOTAL     0.0076, AV.     0.0038 SEC.
   OF WHICH: FOCK BUILD  51.50%. (TOTAL     0.0039, AV.     0.0013 SEC.)

    ATOM     ELEMENTAL            CARTESIAN COORDINATES
   NUMBER     SYMBOL            X           Y           Z

      1         C            -0.01735    -0.02861     0.02953
      2         H            -0.43454     1.00741     0.03920
      3         H            -0.38082    -0.56325     0.93999
      4         H            -0.41371    -0.55645    -0.87146
      5         C             1.48883     0.00488     0.00689
      6         H             1.89022    -1.04241     0.04099
      7         H             1.86797     0.53274     0.92165
      8         C             2.01230     0.70341    -1.23057
      9         H             1.61073     1.75062    -1.26472
     10         H             1.63314     0.17542    -2.14527
     11         C             3.51852     0.73702    -1.25324
     12         H             3.91483     1.26551    -0.35260
     13         H             3.88172     1.27114    -2.16411
     14         H             3.93589    -0.29894    -1.26230

  ATOMIC CHARGES:

  ATOM   ELEMENTAL  PARTIAL    PARTIAL    PARTIAL
   NO.    SYMBOL    MULLIKEN   CM1        CM2
                    CHARGE     CHARGE     CHARGE

    1       C     -0.21050    -0.21050    -0.15334
    2       H      0.07146     0.07146     0.05240
    3       H      0.07145     0.07145     0.05239
    4       H      0.07157     0.07157     0.05252
    5       C     -0.15902    -0.15902    -0.12165
    6       H      0.07749     0.07749     0.05881
    7       H      0.07755     0.07755     0.05886
    8       C     -0.15902    -0.15902    -0.12165
    9       H      0.07750     0.07750     0.05881
   10       H      0.07754     0.07754     0.05886
   11       C     -0.21049    -0.21049    -0.15333
   12       H      0.07158     0.07158     0.05253
   13       H      0.07144     0.07144     0.05238
   14       H      0.07147     0.07147     0.05241

TOTAL MULLIKEN CHARGE =   0.0000
TOTAL CM1 CHARGE      =   0.0000
TOTAL CM2 CHARGE      =   0.0000

-------------------------------------------------------------
TOTAL TIME:            1.742459 SECONDS
        0 DAYS  0 HOURS  0 MIN  1.74 SEC

EE=           -2454.89885480 EC=            1804.08725391
ET=            -650.81160089 HF=             -31.09000923
```

The file that we are really interested in, and the reason we ran Antechamber in the first place, is the `tbutane.prepi` file. This contains the definition of our tbutane residue including all of the charges and atom types that we will load into xLeap to when creating our topology and coordinate. Let's take a quick look at the file:

```
      tbutane.prepi
    0    0    2

This is a remark line
molecule.res
ANE   INT  0
CORRECT     OMIT DU   BEG
  0.0000
   1  DUMM  DU    M    0  -1  -2     0.000      .0        .0      .00000
   2  DUMM  DU    M    1   0  -1     1.449      .0        .0      .00000
   3  DUMM  DU    M    2   1   0     1.522    111.1       .0      .00000
   4  C1    CT    M    3   2   1     1.540    111.208   180.000  -0.09260
   5  H5    HC    E    4   3   2     1.000     70.496  -180.000   0.03216
   6  H6    HC    E    4   3   2     1.000     70.490   -59.954   0.03215
   7  H7    HC    E    4   3   2     1.000     70.490    59.954   0.03227
   8  C2    CT    M    4   3   2     1.500    180.000    90.000  -0.08042
   9  H8    HC    E    8   4   3     1.000    109.504    90.000   0.03819
  10  H9    HC    E    8   4   3     1.000    109.519   -29.994   0.03825
  11  C3    CT    M    8   4   3     1.500    109.510  -149.954  -0.08042
  12  H10   HC    E   11   8   4     1.000    109.451    60.015   0.03820
  13  H11   HC    E   11   8   4     0.999    109.532   -59.944   0.03824
  14  C4    CT    M   11   8   4     1.500    109.510   180.000  -0.09259
  15  H12   HC    E   14  11   8     1.000    109.503   -60.010   0.03228
  16  H13   HC    E   14  11   8     1.000    109.517  -179.917   0.03214
  17  H14   HC    E   14  11   8     0.999    109.523    60.014   0.03217


LOOP

IMPROPER

DONE
STOP
```

This file contains, in internal coordinates, the 3 dimensional structure of our tbutane molecule as well as the charge on each atom, final column, the atom number (column 1), its name (column 2) and it's atom type (column 3). It also specifies loops and improper torsions. This file does not, however, contain any parameters. The GAFF parameters are all defined in $AMBERHOME/dat/leap/parm/gaff.dat. The other thing you should notice here is that all of the GAFF atom types are in lower case. This is the mechanism by which the GAFF force field is kept independent of the macromolecular AMBER force fields. All of the traditional AMBER force fields use uppercase atom types. In this way the GAFF and traditional force fields can be mixed in the same calculation.

While the most likely combinations of bond, angle and dihedral parameters are defined in this file it is possible that our molecule might contain combinations of atom types for bonds, angles or dihedrals that have not been parameterised. If this is the case then we will have to specify any missing parameters before we can create our prmtop and inpcrd files in xLeap.

We can use the utility `parmchk` to test if all the parameters we require are available.

```
parmchk -i tbutane.prepi -f prepi -o tbutane.frcmod
```

Run this command now and it will produce a file called tbutane.frcmod. This is a parameter file that can be loaded into xLeap in order to add missing parameters. Here it will contain all of the missing parameters. If it can, antechamber will fill in these missing parameters by analogy to a similar parameter. You should check these parameters carefully before running a simulation. If antechamber can't empirically calculate a value or has no analogy it will either add a default value that it thinks is reasonable or alternatively insert a place holder (with zeros everywhere) and the comment "ATTN: needs revision". In this case you will have to manually parameterise this yourself. It is hope that as GAFF is developed so the number of missing parameters will decrease. Let's look at our frcmod file:

```
     tbutane.frcmod
remark goes here
MASS

BOND

ANGLE

DIHE

IMPROPER

NONBON
```

We can see that there are no missing parameters. For the purposes of this lab we shall assume that the parameters Antechamber has suggested for us are acceptable. Ideally you should really test these parameters (by comparing to *ab initio* calculations for example) to ensure they are reasonable.

We now have everything we need to load tbutane as a unit in xLeap. We just need to start xLeap and ensure the GAFF force field is available. Since we can mix the traditional AMBER force fields with GAFF we could at this point load a fragment of, say, aliphatic hydroxylase and treat this using the FF99 force field while treating the tbutane molecule using the GAFF force field. For this lab, however, we will simply load our GAFF tbuane and embed it in a water box of TIP3P water. The TIP3P water parameters are loaded as part of the FF99 Leap script so we will have xleap load this on startup:

```
$AMBERHOME/exe/xleap -s -f $AMBERHOME/dat/leap/cmd/leaprc.ff99
```

Once xLeap is up and running we also need to ensure that it knows about the GAFF force field. There is a script in $AMBERHOME/dat/leap/cmd/ that will do this for us. We can load it into xLeap with:

```
>source leaprc.gaff
```

Your xleap window should now look like this:

Now we can load our `tbutane` unit:

```
>loadamberprep tbutane.prepi
```

If you now type list in xleap you should see a new unit called `ANE`. This is the same as the unit name on line 5 of our prepin file.

At this point we haven't loaded the frcmod file that parmchk gave us. While for the sake of generality, we will do this. However, since there were no missing parameters, it is really not necessary.

```
>loadamberparams tbutane.frcmod
```

Now, load the tbutane pdb file, and create topology, coordinate and pdb files. In addition, to make it simplier, should you need to load the tbutane structure, with force field parameters, we will save a lib file. The complete set of commands are:

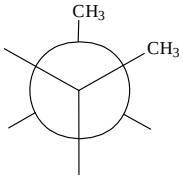
```
>ANE = loadpdb tbutane.pdb
>tbutane = copy ANE
>saveoff tbutane tbutane.lib
>savepdb tbutane tbutane.pdb
>saveamberparm tbutane tbutane.top tbutane.crd
```

If we had needed/wanted to solvate the system with water we would first have to load the file that will define our solvent (water, called TIP3 waters) and solvate the system prior to saving the topology, coordinate and pdb files with the following two commands (but don't enter these command):

```
>loadoff /disk02/usr/local/amber8/dat/leap/lib/tip3pbox.off
>solvatebox ANE TIP3PBOX 10
>saveamberparm tbutane tbutane_wat.top tbutane_wat.crd
>savepdb ANE tbutane_wat.pdb
```

The `tbutane.pdb` is a `.pdb` file that can be reloaded into **xleap** or viewed in **rasmol**. For use by most of the programs in **Amber**, files that have topology information (connectivity, etc) and files that give coordinates (or velocities) are required. These are created by the saveamberparm command.

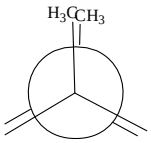
Return to the Unit editor (enter edit `ANE`). Choose 'Select' and click on the two methylene carbons and the four protons attached to these carbons. Now choose 'Twist', move the cursor to the black area, left click and hold and move the cursor. You will see the molecule rotate about the CH_2-CH_2 bond. Twist until you get the gauche conformation as shown to the right in the Neuman projection.



Then, close the Unit editor and write the following files and re-enter the unit editor by entering:

```
gbutane = copy ANE
saveoff gbutane gbutane.lib
savepdb gbutane gbutane.pdb
saveamberparm gbutane gbutane.top gbutane.crd
edit ANE
```

Choose 'Twist', move the cursor to the black area, left click and hold and move the cursor. (Note, the butane structure should still be in the gauche conformation and the central methylenes still highlighted. If they are not, follow the instructions for converting the *anti* form to the *gauche* form). Twist about the methylene bond until the two methyls are eclipsed as shown at the right.



Exit the Unit editor and write the files you will need by entering the following commands:

```
ebutane = copy ANE
saveoff ebutane ebutane.lib
savepdb ebutane ebutane.pdb
saveamberparm ebutane ebutane.top ebutane.crd
```

Diaxial and Diequatorial Dimethylcyclohexane

You should still be in **xleap**. To start drawing the cyclohexane structures enter the commands shown:

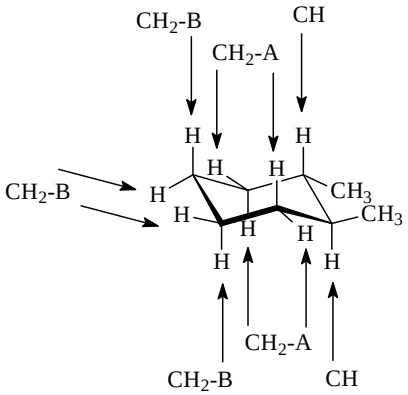
```
edit ch
```

This will put you in the unit editor. Select the draw mode and draw six line segments in a hexagon as follows: Left click and hold, draw a line at about a 60° angle (upward) and release. Left click where you left off and hold, draw a horizontal line and release. Left click and hold, draw a line at about a 60° angle (downward) and release. Continue this process until you have drawn a hexagon. Now left click and hold on any of the ring carbons and move away from the ring and release. Then, left click and hold on a carbon adjacent to this carbon, move the mouse away from the ring and release. You now have the skeleton of dimethylcyclohexane. Click on 'Unit/Add H and build' and release. Bond lengths and angles will be cleaned up and hydrogens added. The molecule will also appear to shrink but it didn't, just zoom to resize.

After building/adding hydrogens, the molecule may be in the diequatorial or diaxial conformation - check to determine which conformer has been built. The conformer shown to the right is the diequatorial. If you have built the diequatorial conformer, follow the instructions as written. If you have built the diaxial, follow the directions below up to where you execute the copy command (`chee = copy ch`), `saveoff`, `savepdb` and `saveamberparm` commands and use the four commands for the diaxial conformer (below) beginning with `chaa = copy ch`.

```
chee = copy ch
savepdb chee dimechee.pdb
```

Return to the Unit editor (enter `edit ch`). Choose 'erase' and click on one of the two methyl groups to delete it. Delete the hydrogen from the carbon the bore the methyl group you just deleted. Select 'Draw' and draw a line where the hydrogen you just deleted was. Select 'H' and draw a line to where the methyl group was that you deleted. Repeat this process for the other methyl group. Now, select 'Unit/Add H and build'. If you started with the diequatorial conformer, you should now have the diaxial conformer of dimethylcyclohexane. If you started with the diaxial conformer, you should now have the diequatorial conformer. Use the zoom/rotation/translation tools to confirm this. Then, select the molecule and then 'Edit' and 'Selected atoms'. Complete the table by adding the missing information for the new methyl groups. Use the charge information for this methyl as was used for the diequatorial cyclohexane above. Check the table, then close with 'Save and Close Table'. Finally, if you were editing the diaxial conformer at this point then close the Unit editor and write the following files and re-enter the unit editor by entering the commands below. If you were editing the diequatorial conformer, use the four commands given above for the diequatorial isomer beginning with `chee = copy ch`.



```
chaa = copy ch
savepdb chaa dimechaa.pdb
```

Now, you have a pdb file for both the aa and ee forms of dimethylcyclohexane. Exit xleap and prepare a prepi and frcmod file that will enable you to write topology and coordinate files. The process you will follow is the same as with the trans, gauche and eclipsed butanes. Be sure that, at the appropriate time in xleap you execute the following commands for the ee form:

```
saveoff chee dimechee.lib
saveamberparm chee dimechee.top dimechee.crd
```

and the following commands for the aa form so that the topology and coordinate files are written for each. You will need these to run the molecular mechanics simulations. However, you will use the same '.in' file as you used for the three butanes.

```
saveoff chaa dimechaa.lib
saveamberparm chaa dimechaa.top dimechaa.crd
```

Its time for running **sander**. You can quit **xleap** if you want or just open another unix shell in which you can run **sander**. The latter is preferred as you will want to view the output from **sander** and it is most convenient to do this in **xleap**.

Molecular Mechanics Energy Minimization - Sander

Methane

Control file

Write the input file. This can be done use jot. Type

```
jot methane.in
```

Then enter the following lines exactly as shown:

```
min (no shake, periodic) - just enough to relax for md
 &cntrl
  imin = 1, maxcyc = 500, ncyc =500,
  ntb = 1, cut = 10.0,
 /
```

Then, click on file/save and then exit **jot**.

(Note: Be sure to write this file exactly as shown. The parameters have the following meaning:

imin	controls whether the run is molecular mechanics (=1) or dynamics (=0)
maxcyc	maximum number of minimization cycles
ncyc	switching point from steepest decent to conjugate gradient
cut	distance after which atom-atom interactions will no longer be computed
ntb	whether there is a periodic boundary (=0, no), run is under constant volume (=1) or constant pressure (=2)

Energy minimization

This is done to clean up any steric problems at the box boundary. These can occur because the initial, replicated periodic box (`TIP3PBOX`) is trimmed at the cutoff we specified (10A) without considering the waters on the opposite sides of the box. To run **sander** type the command shown below (This is typed as one line. The \ at the end of the first line says that the command is continued on the next line):

```
sander -O -i methane.in -p mebox.top -c mebox.crd
-o mebox_min.out -r mebox_min.rst &
```

Your can also write this command line into a file and then when you want to run it, just source the file. The files used by **sander** are (1) `methane.in`, the control file, (2) `mebox.top`, the topology file, (3) `mebox.crd`, the input coordinate file, (4)mebox_ `min.out`, the file to which energies of each interaction will be printed along with other useful information, and (5)mebox_`min.rst`, the output coordinate file. It is called the restart file since it is used to continue an energy minimization if maxcyc was too small. The -O means to over-write output files that may exist (mebox_`min.out` or mebox_ `min.rst`). This is usually O.K. as the only time you will re-run a minimization is if something failed but you'll have to make sure it is O.K. Finally, the '&' at the end of the line tells unix to run the program in the background so that you can do other stuff. If you leave it off you will have to wait until the program finishes or open another unix shell in order to continue working. The order of the files does not matter, though the `-O` option, if used, must go immediately following the program name (**sander**) and the indication of the file type (`-i, -o, -p,` etc., must immediately preceed the filename.

Viewing the energy minimized structure

The combination of mebox_`min.rst` and `mebox.top` defines the minimized structure. These two files can be used to generate a `.pdb` file. This is done using a program call **ambpdb**. Enter the command

```
ambpdb -p mebox.top < mebox_min.rst > mebox_min.pdb
```

:

The structure of the command line is `-p mebox.top` is the topology file you want to use, the coordinate file will be input from `min.rst`, and the output will be directed into the file called `mebox_min.pdb`.

Once you have generated the `.pdb` file, run **xleap** and once it starts enter the following commands:

```
loadoff methane.lib
m = loadpdb mebox_min.pdb
edit m
```

This set of commands will load parameters needed for **xleap** to 'understand' you methane structure (`loadoff methane.lib`), then load the structure into `mth` (`mth = loadpdb mebox_min.pdb`) and put you in the unit editor to view it (`edit mth`). An alternative is to use **rasmol** instead. If you do, after you generate the `.pdb` file, run **rasmol** and load the file once **rasmol** starts. Note that if you are running on either Hal or Taz, you will have to ftp the file and run **rasmol** locally. At this time, ruby does not have a wroking version of **rasmol** so, if running on ruby you will have to use **xleap**.

The methane/water energy minimized structure is not very interesting as only the waters will have moved around. However, if you are successful up to this point, you are ready to do something a bit more interesting.

Eclipsed, gauche and anti butane

Energy minimization

Running the butane conformers is pretty simple. You will use a modified control file of the one you used to run methane. First copy `methane.in` to `butane.in`. Then, modify `butane.in` as follows. Type

```
jot butane.in
```

This will open the `butane.in` file (which is just a copy of `methane.in` at this point, shown below).

```
min (no shake, periodic) - just enough to relax for md
 &cntrl
  imin = 1, maxcyc = 500, ncyc =250,
  ntb = 1, cut = 10.0,
 /
```

Modify the `butane.in` file so that it reads as shown below:

```
min (no shake, periodic) - just enough to relax for md
 &cntrl
  imin = 1, maxcyc = 500, ncyc = 250,
```

```
 ntb = 0, igb = 1, cut=16.0,
/
```

The change is that ‘`ntb = 0`'. Once you have modified `butane.in` click on File, Save As and save the file as butane.in and then exit **jot**.

The other files necessary, the topology and coordinate files were previously created in **xleap**. Your biggest problem will be keeping track of all the files you are about to create.

You will run **sander** three times, once for each of the three butane conformers you created. This can be done by entering the following three commands:

```
sander -O -i butane.in -p tbutane.top -c tbutane.crd -o \
tbutane.out -r tbutane.rst &

sander -O -i butane.in -p gbutane.top -c gbutane.crd -o \
gbutane.out -r gbutane.rst &

sander -O -i butane.in -p ebutane.top -c ebutane.crd -o \
ebutane.out -r ebutane.rst &
```

Viewing the minimized structures

Once these have run, you’ll have to generate .pdb files and then view them either in **xleap** or **rasmol**. The `.pdb` files will be generated using **ambpdb** and should have the names shown below. You will need to determine what the command line is for their generation by substituting the appropriate `topology`, `coordinate` and `.pdb` filenames for the ones used for methane.

Butane conformer	.pdb file name
anti	tbutane_min.pdb
gauche	gbutane_min.pdb
eclipsed	ebutane_min.pdb

Use **rasmol** to view the results. It can be done in **xleap** but is a bit more complicated. Be sure to make a drawing of what each butane looks like using the space provided below. What changes occured? You can load the original `pdb` file into **rasmol** to compare.

tbutane	gbutane	ebutane

Energy of the minimized structure

Before you move on the dimethylcyclohexane molecule, take a look at the files called `tbutane.out`, `gbutane.out`, and `ebutane.out`. What you are most interested in is at the end of the file. No matter, take a look at the whole `tbutane.out` file first with the command

```
more tbutane.out
```

First you will see information regarding the input into **sander** including file names and program control parameters. Then, you will get a listing of the energy every 5 iterations and then a listing of the final results which looks like:

```
                    FINAL RESULTS

   NSTEP       ENERGY          RMS            GMAX         NAME    NUMBER
     267   1.7596E+00     8.3704E-05     2.3877E-04     H5           2

 BOND    =        0.0359  ANGLE   =        0.1011  DIHED      =        0.4071
 VDWAALS =       -0.1507  EEL     =        0.8728  EGB        =       -0.1097
 1-4 VDW =        0.7268  1-4 EEL =       -0.1236  RESTRAINT  =        0.0000
```

In this example, the final energy is 1.76 kcal/mole (under ENERGY), though your values may be different - this is just an example. This value is split out into various energy types. You can see that the main contributors to the energy are 1-4 VDW , EEL (electrostatic energy) and 1-4 EEL. Strain due to bond stretch, bond angles and torsions (DIHED) are rather small. You can also see the run did not take very long. Your run may take longer as the data above was obtained when amber (the computer) could pay full attention to **sander**.

For the `gbutane.out` and `ebutane.out` files, take a look at the last part of the file with the tail command:

```
tail -40 gbutane.out
```

or

```
tail -40 ebutane.out
```

and make note of the final energies for each.

Energy			
	`tbutane`	`gbutane`	`ebutane`

You might think that regardless of where you started (`tbutane, gbutane` or `ebutane`) you would end up at the same place. You don't. Why? (see Exercises, below)

Diequatorial and diaxial dimethylcyclohexane

Energy minimization

Running the dimethylcyclohexane conformers is also pretty simple. You will follow the same procedure as for the butanes.

You will run **sander** twice, once for each of the two dimethylcyclohexane conformers you created. This can be done by entering the following two commands (note, if you named the files differently, be sure to use your file names):

```
sander -O -i butane.in -p dimechee.top -c dimechee.crd -o \
dimechee.out -r dimechee.rst &

sander -O -i butane.in -p dimechaa.top -c dimechaa.crd -o \
dimechaa.out -r dimechaa.rst &
```

Viewing the minimized structures

Once these have run, make the `.pdb` files using **ambpdb** and should have the names shown below. Once again, you will need to determine what the command line is for their generation by substituting the appropriate `topology`, `coordinate` and `.pdb` filenames for the ones used for methane.

Dimethylcyclohexane conformer	.pdb file name
diequatorial	dimechee_min.pdb
diaxial	dimechaa_min.pdb

and view them using **rasmol**. Make a drawing of what each looks like and note whether there has been any significant change.

diequatorial	diaxial

Energy of the minimized structure

Examine the output files as for the butanes and record the final energies.

```
tail -40 dimechee.out
```

or

```
tail -40 dimechaa.out
```

and make note of the final energies for each

Energy	`dimechee`	`dimechaa`

Again, you might think that regardless of where you started (dimechee or dimechaa) you would end up at the same place. You don't. Why? The answer is the same as for the butanes.

Exercises

To be turned in upon completion of the lab:

Butane

1) What energies did you obtain for the final structure of *trans-*, *gauche-*, and *eclipsed* butane. Which had the smallest energy?

2) What did the final structures look like. You can draw them but it would be good practice to produce pictures from either **rasmol** or **moilview**. Note, if you produce printed pictures, they are to have a white background, not the default black background.

3) Why didn't the minimization converge to the same structure? Consider which initial structures did converge, which didn't, and the energies of the final structures.

4) Which structure should all three of the butanes converge to?

Cyclohexanes

1) What were your final energies for diaxial- and diequatorial- dimethylcyclohexane?

2) Provide pictures of the final geometries for each. Note, if you produce printed pictures, they are to have a white background, not the default black background.

3) Why don't the two starting points converge to the same structure?

4) Propose a solution to this problem. How could you get both starting conformers to minimize to one structure (and they really should since there should only be one 'most stable' conformation' since they are just two different conformers of the same compound).

References

References

[1]Wang, J., Wolf, R.M., Caldwell, J.W., Kollman, P.A., Case, D.A. "Development and Testing of a General Amber Force Field", J. Comp. Chem., 2004, **25**, 1157 - 1173.

[2]Jakalian, A., Bush, B.L., Jack, B.D., Bayly, C.I., "Fast, Efficient Generation of High-Quality Atomic Charges. AM1-BCC Model: I. Method.", J. Comp. Chem., 2000, **21**, 132-146.