Berkeley Computational Nanoscience

NSE C242 & Phys C203

Spring, 2008

Lecture 4:

Energy Minimization, Geometry Optimization, and Seeing What You're Doing

January 31, 2008

Jeffrey C. Grossman

Elif Ertekin

Energy Minimization

In nearly all cases, the potential energy surface is complicated and highly multidimensional

It is impossible to visualize the entire energy surface

Minimum energy atomic arrangement corresponds to stable states of the system.

There could be a number of “local minima” although there is only one “global energy minimum.”

Finding this point is called energy minimization or geometry optimization.

Energy Minimization

Given a function *f* which depends on one or more independent variables, x_1, x_2, …, x_i, at the minimum point the first derivative of the function with respect to each variable is zero and the second derivatives are all positive:

$$\frac{\partial f}{\partial x_i} = 0 \qquad \frac{\partial^2 f}{\partial x^2{}_i} > 0$$

Typically *f* is the total energy and the x_i are the 3N Cartesian coordinates of the system (or the 3N-6 *internal* coordinates).

There are essentially two classes of minimization algorithms:

Those which use derivatives of the energy

Those which do not use derivatives of the energy

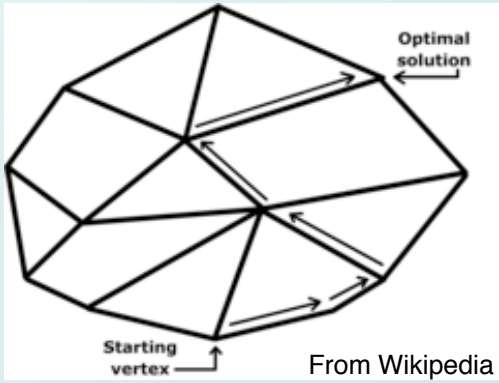
Non-Derivative Methods

The simplex method or the sequential univariate method are two examples of non-derivative minimization approaches.

A geometrical figure is generated with M+1 vertices, where M is the dimensionality of the energy function.

One of the vertices corresponds to the initial coordinates, and all the others are generated by moving each coordinate in turn.

The energy is evaluate at each vertex, so for a N-particle system we need 3N+1 energy calculations.



From Wikipedia

Various “moves” in phase space are then attempted from a geometric standpoint: reflection, expansion, and contraction, for example, about a given face of the simplex structure.

This is a painfully slow method, but sometimes can have uses as a way to “prep” a starting configuration that is at a very high (unphysical) energy.

Derivative Methods

Derivatives may be obtained either analytically or numerically.

Analytic derivatives are preferred if available.

Energy functions in simulations are rarely quadratic.

An incomplete list of derivative methods for minimization:

First Derivative methods

- Steepest Decents
- Line Search
- Conjugate Gradients

Second Derivative Methods

- Newton-Raphson
- Quasi-Newton

Example Derivative Approaches

Conjugate Gradient

This approach* has the property that for a quadratic function of *M* variables, the minimum will be reached in *M* steps.

A move is taken in a direction **v** at a step *k* by: $$\mathbf{v}_k = -\mathbf{g}_k + \frac{\mathbf{g}_k \cdot \mathbf{g}_k}{\mathbf{g}_{k-1} \cdot \mathbf{g}_{k-1}} \mathbf{v}_{k-1}$$

where **g** is the gradient of the energy with respect to position

Newton-Raphson

If the potential energy is V(**x**) where **x** has 3N components, then V'(**x**) is a vector (the derivative) and V''(**x**) is a matrix (force constant matrix or Hessian).

The first derivative can be expanded: $V'(\mathbf{x})=V'(\mathbf{x}_k)+(\mathbf{x}-\mathbf{x}_k)V''(\mathbf{x}_k)$

At the minimum coordinate $\mathbf{x}_{min}$: $V'(\mathbf{x}_{min})=0$, so $\mathbf{x}_{min}= \mathbf{x}_k - V'(\mathbf{x}_k)\, V''(\mathbf{x}_k)^{-1}$

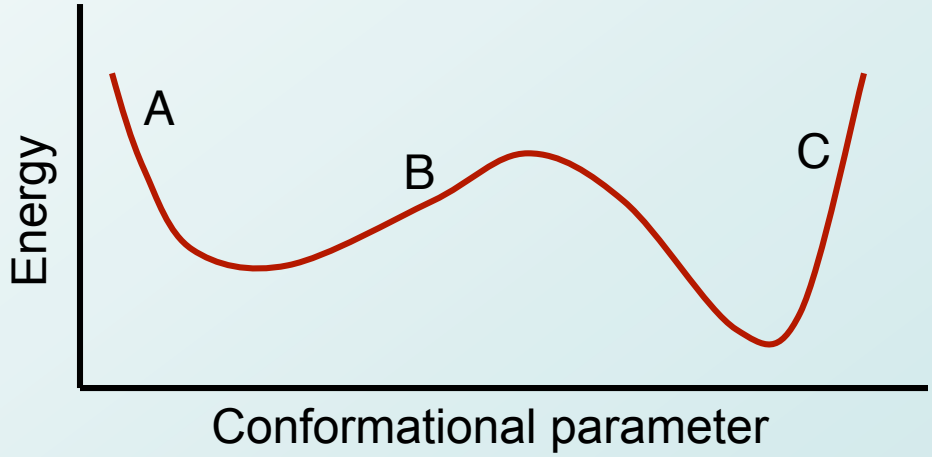
* Magnus R. Hestenes and Eduard Stiefel (1952), Methods of conjugate gradients for solving linear systems, *J. Research Nat. Bur. Standards* 49, 409–436.

Energy Minimization

What would make a minimization algorithm a good one?

- it's fast
- uses little memory
- it's fast
- it's accurate

Most minimization algorithms mostly can only go downhill. Than means that the result could depend on where you start.



Initial Conditions for Minimization

So, where do we start?

In an “atomistic” calculation, we need to come up with an (x,y,z) position for every atom in the system.

For small molecules, we can just guess and then minimize starting from the guess. The change in energy from where we started will tell us how good of a guess we made.

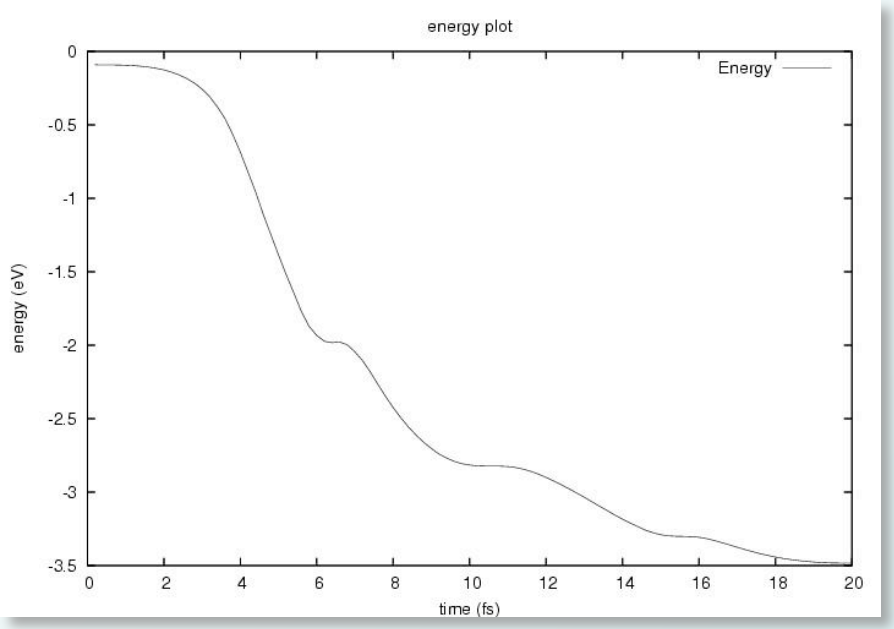
Let’s take an example: CH_4 (methane).

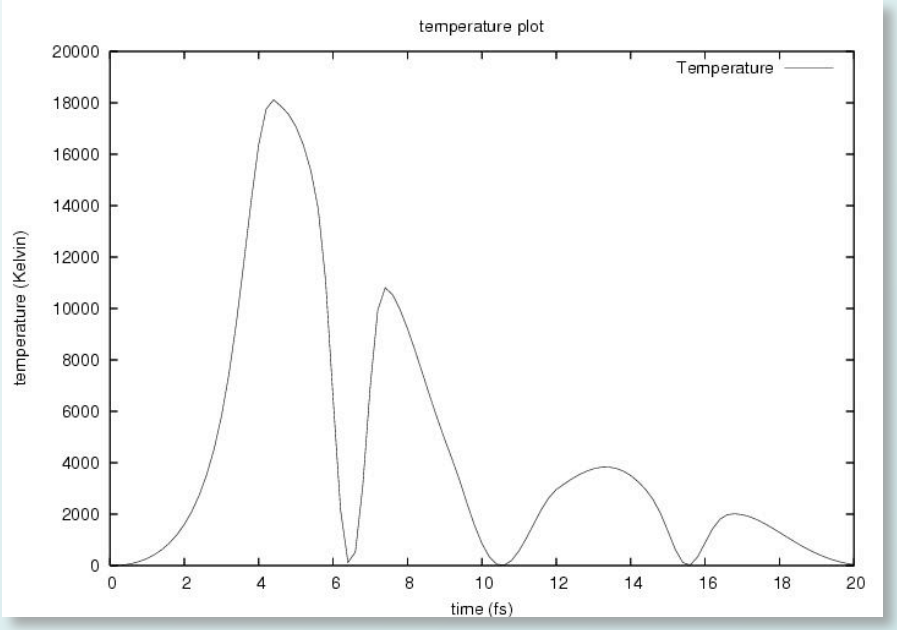
Here’s a guess for the initial structure:

atom	**X**	**Y**	**Z**
C	0.0	0.0	0.0
H	1.0	1.0	1.0
H	-1.0	-1.0	1.0
H	-1.0	1.0	-1.0
H	1.0	-1.0	-1.0

Initial Conditions

Here's what we get for the energy and temperature:



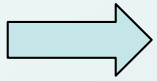


Did this work well? Was it a good optimization?

Minimization of Methane

Initial Coordinates

atom	X	Y	Z
C	0.0	0.0	0.0
H	1.0	1.0	1.0
H	-1.0	-1.0	1.0
H	-1.0	1.0	-1.0
H	1.0	-1.0	-1.0



Final Coordinates

atom	X	Y	Z
C	0.0	0.0	0.0
H	0.7	0.7	0.7
H	-0.7	-0.7	0.7
H	-0.7	0.7	-0.7
H	0.7	-0.7	-0.7

First of all, was it accurate? How can we tell?

One possibility is to check the C-H bond length: we know the C-H bond length from experiment to be around 1.1 Angstrom.

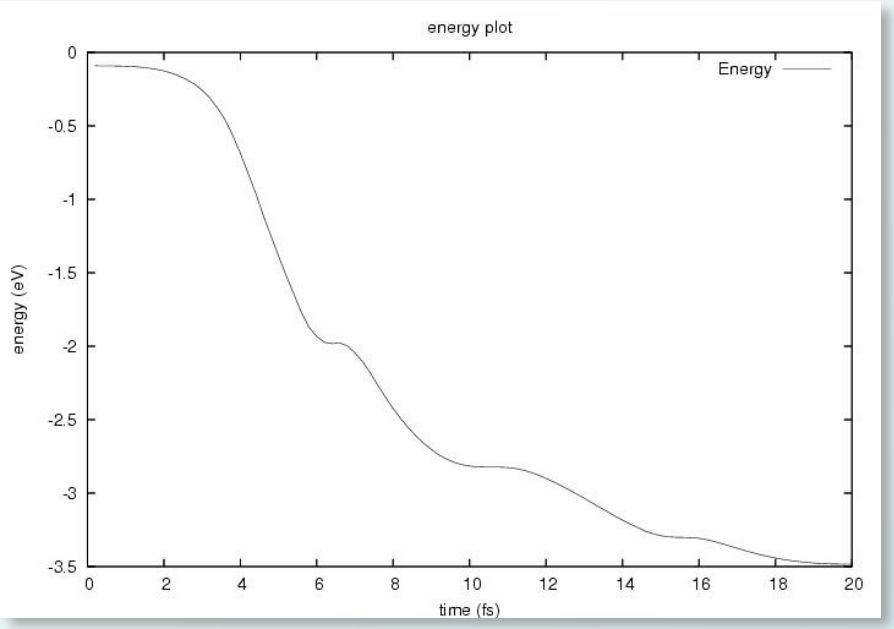
Our initial guess was 1.7 Angstrom, not so good. After optimization the distance came down to 1.2 Angstrom - much better.

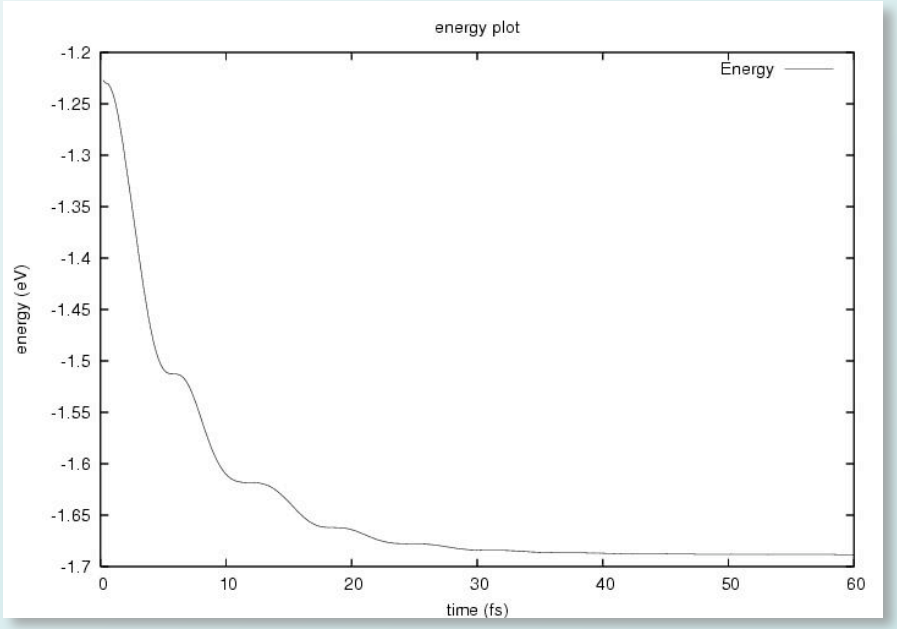
Why didn't the distance match exactly with experiment?

Starting Configuration

In general, the “guessing” approach is ok but can be risky or at least challenging, even for very small molecules.

Consider starting from two different guesses:





Starting Configuration

A better way to start would be to get coordinates that are as close to the real system as possible.

This can mean finding a structure from experiments such as X-ray or NMR.

It can also mean finding a structure from previous theoretical calculations.

It can also mean being a bit creative.

Starting Configuration

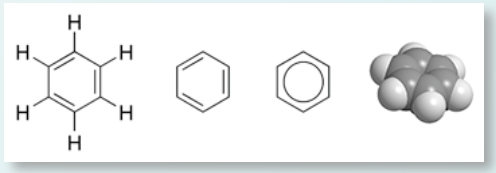
Where else can one get coordinates?

The web!! Some resources:

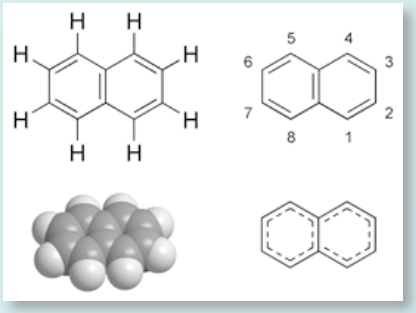
- Google
- Wikipedia (and the “computational chemistry wiki”)
- Random people’s web sites
- Archives

And if we get a structure from the web, say from an experimental measurement, do we even need to minimize it?

You can also combine different approaches. For example, you may find a good starting structure for benzene but not for napthalene.



Remove 2 H atoms and duplicate a C_4H_4 group



Reaching the Minimum

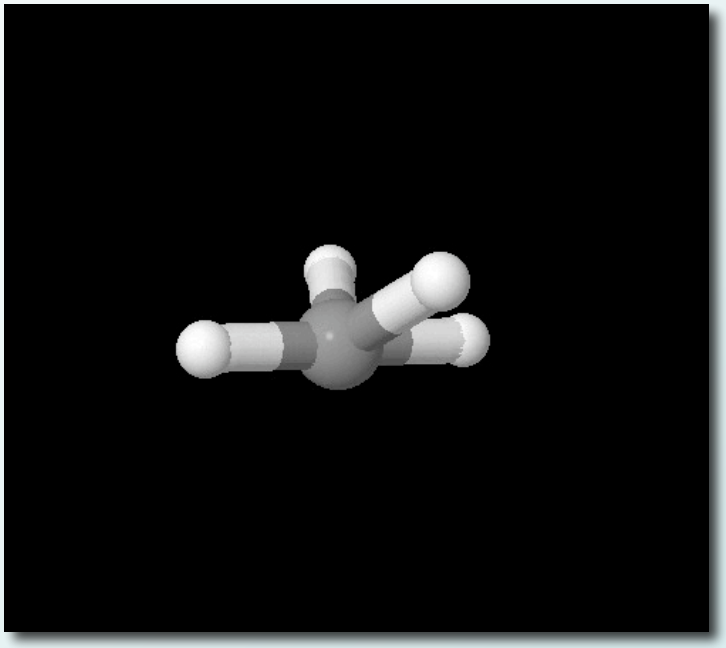
How do we know when we've obtained a minimum?

Typically, we monitor more than one thing to see if the simulation has reached a suitable minimum. For example, at each minimization step we could monitor:

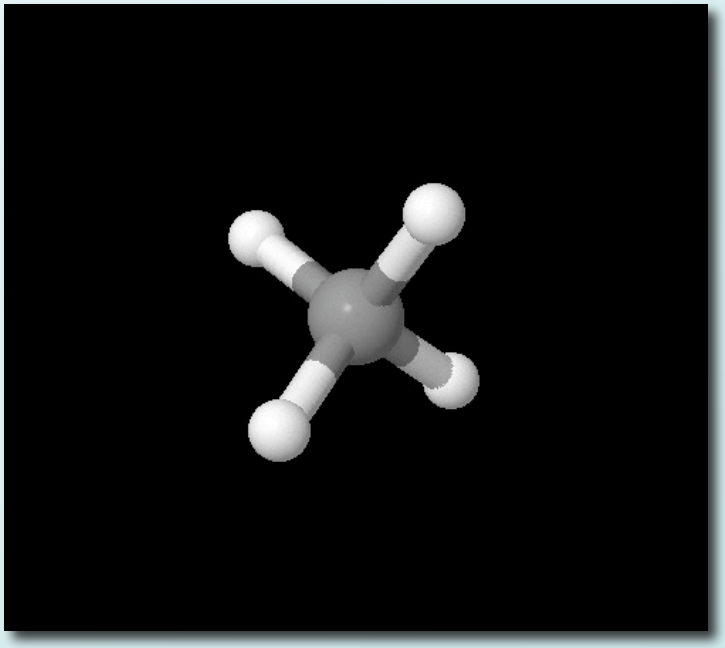
- Energy
- Change in coordinates
- Root-mean-square gradient
- Maximum gradient

The Need to See What We're Doing

atom	X	Y	Z
C	0.0	0.0	0.0
H	0.7	0.7	0.7
H	-0.7	-0.7	-0.7
H	-0.7	-0.7	0.7
H	0.7	-0.7	-0.7



atom	X	Y	Z
C	0.0	0.0	0.0
H	0.7	0.7	0.7
H	-0.7	-0.7	0.7
H	-0.7	0.7	-0.7
H	0.7	-0.7	-0.7



http://jmol.sourceforge.net/

Starting Configuration Formats

Ok, so I download some coordinates, and they're in PDB format…what is that?

```
HEADER    Nanotube
TITLE     (4,0) Nanotube
AUTHOR    TubeGen 3.3, J T Frey, University of Delaware
EXPDTA    THEORETICAL MODEL
ATOM      1  C   UNK     1       1.597   0.000  -0.724  1.00  0.00
ATOM      2  C   UNK     1       1.597   0.000  -2.145  1.00  0.00
ATOM      3  C   UNK     1       1.129   1.129   1.421  1.00  0.00
ATOM      4  C   UNK     1       1.129   1.129   0.000  1.00  0.00
ATOM      5  C   UNK     1       0.000   1.597  -0.724  1.00  0.00
ATOM      6  C   UNK     1       0.000   1.597  -2.145  1.00  0.00
ATOM      7  C   UNK     1      -1.129   1.129   1.421  1.00  0.00
ATOM      8  C   UNK     1      -1.129   1.129  -0.000  1.00  0.00
ATOM      9  C   UNK     1      -1.597   0.000  -0.724  1.00  0.00
ATOM     10  C   UNK     1      -1.597   0.000  -2.145  1.00  0.00
ATOM     11  C   UNK     1      -1.129  -1.129   1.421  1.00  0.00
ATOM     12  C   UNK     1      -1.129  -1.129  -0.000  1.00  0.00
ATOM     13  C   UNK     1      -0.000  -1.597  -0.724  1.00  0.00
ATOM     14  C   UNK     1      -0.000  -1.597  -2.145  1.00  0.00
ATOM     15  C   UNK     1       1.129  -1.129   1.421  1.00  0.00
ATOM     16  C   UNK     1       1.129  -1.129   0.000  1.00  0.00
CONECT    1    2    4   16
CONECT    2    1
CONECT    3    4
CONECT    4    1    3    5
CONECT    5    4    6    8
CONECT    6    5
CONECT    7    8
CONECT    8    5    7    9
CONECT    9    8   10   12
CONECT   10    9
CONECT   11   12
CONECT   12    9   11   13
CONECT   13   12   14   16
CONECT   14   13
CONECT   15   16
CONECT   16    1   13   15
MASTER        0    0    0    0    0    0    0    0   16    0   16    0
END
```

This is one of many, many types of coordinate file formats.

Different simulation tools use different formats.

Some tools can read multiple formats.

In other cases, one needs to convert from one format to another.

http://openbabel.sourceforge.net

Starting Configuration Formats

The simple “XYZ” coordinate files are quite nice:

```
16
(4,0) Nanotube -- TubeGen 3.3, J T Frey, University of Delaware
  C         1.597332     0.000000    -0.724310
  C         1.597332     0.000000    -2.145310
  C         1.129484     1.129484     1.421000
  C         1.129484     1.129484     0.000000
  C         0.000000     1.597332    -0.724310
  C         0.000000     1.597332    -2.145310
  C        -1.129484     1.129484     1.421000
  C        -1.129484     1.129484    -0.000000
  C        -1.597332     0.000000    -0.724310
  C        -1.597332     0.000000    -2.145310
  C        -1.129484    -1.129484     1.421000
  C        -1.129484    -1.129484    -0.000000
  C        -0.000000    -1.597332    -0.724310
  C        -0.000000    -1.597332    -2.145310
  C         1.129484    -1.129484     1.421000
  C         1.129484    -1.129484     0.000000
```

Line #1 is the total number of atoms.

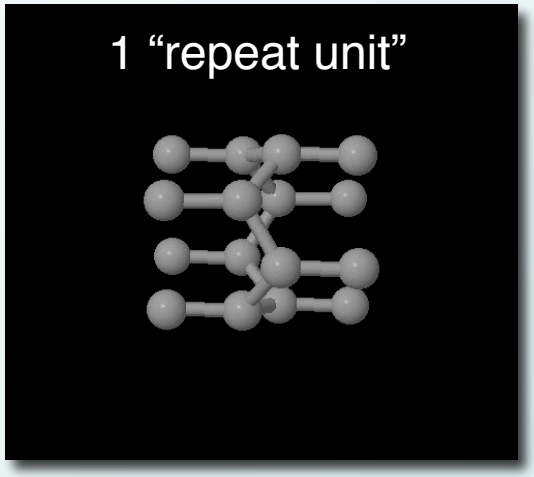
Line #2 is a comment line.

Rest is a list of atoms, where each line contains the atom species followed by its Cartesian coordinates.

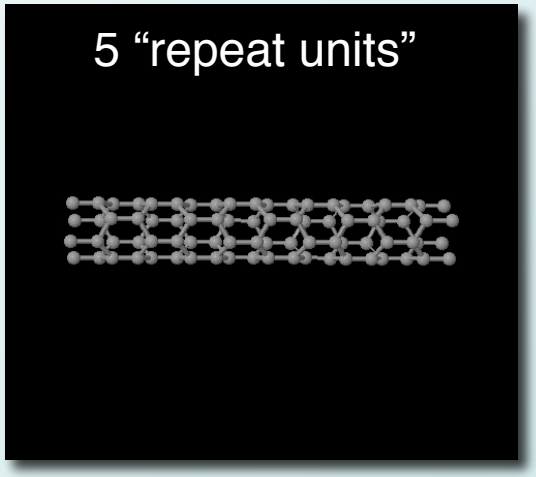
Starting Configuration Formats

For this carbon nanotube example, the simulation cell is periodically repeating in 1 of the 3 dimensions. Given that, how do we determine the cell size?

Both of these coordinate files are for a (4,0) nanotube.



z-dimension of cell = 4.263

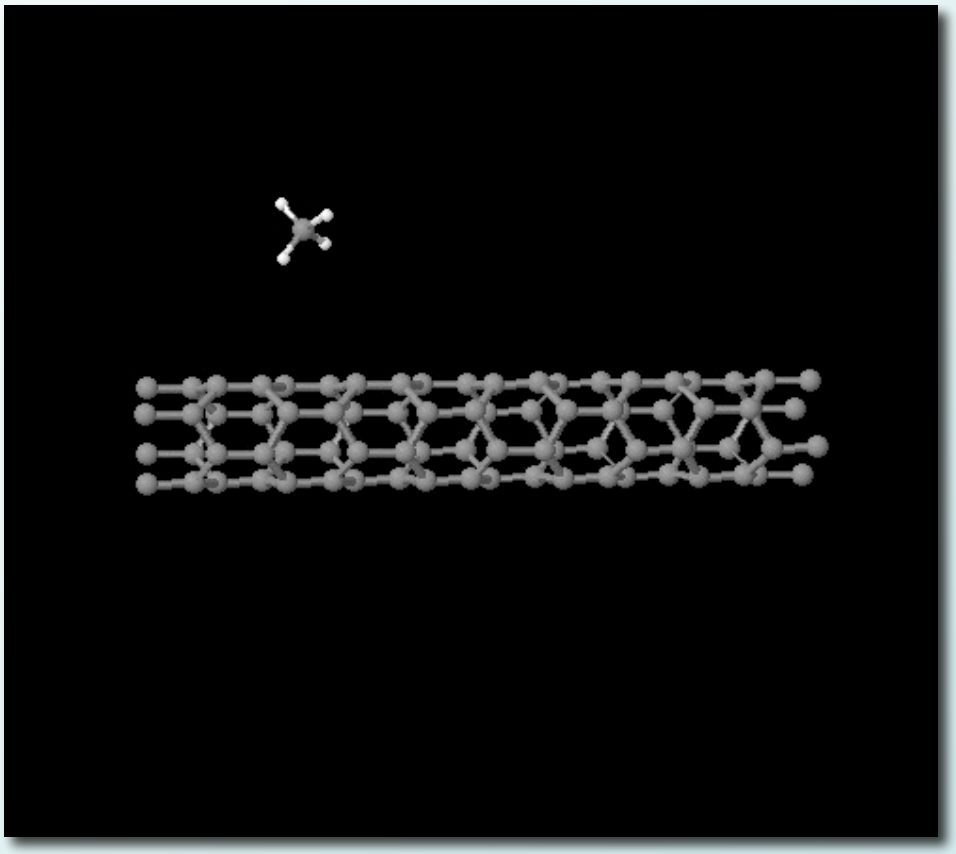


z-dimension of cell = 21.315

Which one should be used for a simulation?

Starting Configuration Formats

As a warm-up for next week, let's make this system together:



In-Class Simulation

Please Bring Your Laptop (wireless capable) on Tuesday!