Exercise 1 Setting up your environment

We will use MATLAB and LAMMPS. The latter is a Molecular Dynamics simulation engine and it is written in C++. Like many scientific computer programs, it is easiest to run in a linux environment. Since you all have different devices with different software, we have decided to run LAMMPS through Jupyter Notebook, which can be accessed in a browser.

A table with some useful linux commands can be found at the end of this document.

a) Go to `https://jupyterhub.science.ru.nl` and login with your science account. If it does not work, let me know so we can arrange a login for you. You will be needing it next week.

b) Press "Spawn". This will send a request to the compute server to open a job for you and it may take a while if it compute server is busy. Don't press a second time.

c) You will now see a directory list of your home directory. We will refer to this as the "Dir Tab". Open a terminal by clicking "New" $\longrightarrow$ "terminal". This will open a new tab on your browser with a command line on the clusternode where you are running your Jupyter session. This new tab will be refered to as "Term Tab".

d) Create a new directory for this course by typing `mkdir MolecularModeling` in the "Term Tab" and check in the "Dir tab" if you can find it there. Click on it. It should be empty. In the "Term Tab" you can also go to this directory by `cd MolecularModeling` .

e) Download all the files by typing `wget http://www.theochem.ru.nl/~hcuppen/MM.tgz`

f) Extract the files by `tar -gxvf MM.tgz`

g) In the "Dir Tab" you should now be able to find the downloaded files. All files that you need together with MATLAB are in separate directory. You can download them to your own computer and use them there. There is directories with LAMMPS input files and one called `papers`.

h) You are now all set to start and you will not need jupyterhub today. To exit the Jupyter Notebook, you need to

   (a) stop running the notebook and let it save your work. Click "Running" in "Dir Tab" and stop what ever process is running.

   (b) let the compute server know that you are done and that someone else can take up your space. Click on "Control Panel" in the top-right corner of one of the tabs. Press "Stop Server".

i) Finally, you will need VMD during this course which you can download from `https://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=VMD`. The structures that you will download in the next exercise you can also view with VMD.

Exercise 2 Obtaining structures from databases

Every modeling effort starts with obtaining a 3D model of the system of interest. This can be done in three ways:

a) by constructing a Z-matrix

b) by drawing it by hand in a viewer

c) by taking a previously measured or calculated geometry from for instance a database

A Z-matrix uses the internal coordinates of a molecule whereas most viewer represent the atoms in their Cartesian coordinates. Method 1 is therefore good for relatively simple molecules with symmetry, where symmetry must be retained during geometry optimization and the second method is better for asymmetric organic molecules. In this exercise we will cover the latter of the three possibilities.

Structures of molecules can be determined experimentally or predicted computationally. For small volatile molecules, such as carbon monoxide, and water, analysis of rotational lines in the microwave or infrared spectra of gaseous samples provides very accurate geometries. For larger volatile molecules, gas electron diffraction data are commonly used in structure determination, typically in conjunction with analysis of rotational spectrum. As a result of such efforts, bond lengths of nearly every stable diatomic and triatomic molecule are known with accuracy better than 0.005 angstroms. Experimental structures of many tetra-atomic and larger molecules are also known with similarly good accuracy. Structure determination of larger molecules in the gas phase becomes

challenging as the spectra become more crowded and weaker due to low vapor pressure of larger molecules. The analysis is further complicated by the presence of multiple conformations in flexible molecules. However, structures of most organic molecules and of many biological macromolecules can be determined accurately in the solid state using X-ray or neutron diffraction.

In the solid state, the field of structure determination is dominated by X-ray and neutron diffraction and very many crystal structures are known. Nuclear magnetic resonance (NMR) also has a role to play, especially for proteins. All of these topics are well discussed in every university-level general chemistry text. Over the years, a vast number of molecular structures have been determined and there are several well-known structural databases. One is the Cambridge Structural Database (CSD) (`http://ccdc.cam.ac.uk/`), which is supported by the Cambridge Crystallographic Data Centre (CCDC). The CCDC was established in 1965 to undertake the compilation of a computerized database containing comprehensive data for organic and metal-organic compounds studied by X-ray and neutron diffraction. It was originally funded as part of the UK contribution to international data compilation. According to its mission statement, the CCDC serves the scientific community through the acquisition, evaluation, dissemination and use of the worlds output of small molecule crystal structures.ps For each entry in the CSD, three types of information are stored. First, the bibliographic information: who reported the crystal structure, where they reported it and so on. Next comes the connectivity data; this is a list showing which atom is bonded to which in the molecule. Finally, the molecular geometry and the crystal structure. The molecular geometry consists of Cartesian coordinates. The database can be easily reached through the Internet, but individual records can only be accessed on a feepaying basis. We will work with data from the CSD at a later stage.

The Brookhaven Protein Data Bank (PDB) is the single worldwide repository for the processing and distribution of three-dimensional biological macromolecular structural data. It is operated by the Research Collaboratory for Structural Bioinformatics. At the time of writing, there were 19 749 structures in the databank, relating to proteins, nucleic acids, proteinnucleic acid complexes and viruses. The databank is available free of charge to anyone who can navigate to their site `http://www.rcsb.org/`. Information can be retrieved from the main website. A four-character alphanumeric identifier, such as 1PCN, represents each structure. The PDB database can be searched using a number of techniques, all of which are described in detail at the homepage.

a) Go to the website of the PDB and search for a structure of human insulin. Click on "Display Files" to view the content of the PDB-file. This file has the following layout: A typical .pdb file starts with bibliographic data and details on the experimental conditions under which the structure has been obtained, it then moves on to the cartesian coordinates (expressed in ångstroms and relative to an arbitrary reference frame) and connectivity data. Try to find all components.

b) Search for the structure with label 1YEB. View again the PDB file. It contains quite a number of water molecules (HOH). With "3D View" you can see where these are situated. This is because these were resolved during the X-ray structure determination and are essential for the stability of the protein structure. Display the unit cell. This is a rather empty cell, but this because other molecules are present at symmetry related positions. The $P4_32_12$ space group has $Z = 8$ which means that there will be eight of these clusters in the unit cell. If we want to use this structure simulate the dynamics in the crystal, we would first need to add these seven to the cell.

c) The structure consists of different components. If you hover your mouse above these components, you can see them light up. The "Polymer", which is the protein sequence is represented in "Cartoon" style here. Protein sequence can be represented in different ways. Here we can already view three of them and in simulations a fourth is important as well.

Cartoon representation in terms of helices or rods. In simulations of polymer dynamics, the systems is is often represented in terms of connected rods, since the flexibility/stiffness of the polymers is determining and atomistic detail is not required. This model is hence very "coarse".

Protein sequence like the letter on top of the view screen. For simulation of large protein structures a little less coarse model is used, where the system is simulated at the level of individual amino acids.

Unit atom where several atoms are joined. This is typically done for H atoms. You can view this by changing the representation of the "Polymer" to "Ball & Stick". You will notice that you can now see the individual atoms, although the H atoms are missing. This is because they cannot be resolved by X-ray diffraction.

All atom including the H atoms. For smaller systems (or large systems of small molecules) one would like to consider all atoms. Starting from an X-ray structure one would need to add the H atoms before the simulation.

Exercise 3 Examine a 2D PES

In this exercise we will examine a potential energy surface using MATLAB. For a good introduction into MATLAB I refer to `http://www.theochem.ru.nl/matlab`; for an overview of useful commands I refer to `http://web.mit.edu/18.06/www/Spring09/matlab-cheatsheet.pdf`.

You will need three data files called `PES.dat`, `x.dat`, `y.dat`. We will use these files to examine a possible potential energy surface.

a) Open MATLAB and load the three files.

b) View the PES using the commands `mesh` or `contour`.

c) What are the stationary points (guess by eye)?

d) Write a short routine that outputs two matrices: one with the numerical first derivative in the x (`Dx`) and one in the y direction (`Dy`).

e) You can now find the grid points closely located near the stationary points of the PES using

```
[iy,ix] = find(abs(Dx) < threshold & abs(Dy) < threshold)
```

with `threshold` some value. The chance that we have the exact stationary points on our grid is rather small, so we will not find them using

```
[iy,ix] = find(Dx == 0 & Dy == 0)
```

Play with the threshold value and determine the stationary points. Plot these on the PES. With the command `hold` you keep the current graph and plot things on top. Are the stationary points maxima, minima or saddle points?

f) You can use your script to obtain the second derivative in both directions. This will result in four matrices, from which you can construct a Hessian matrix in each stationary point. Use the function `eig` to determine the eigenvalues of this Hessian and classify each stationary point. Does this correspond with your visual inspection?

Unix commands

This table summarises some useful Unix terms, commands and programs. The square brackets indicate command options or arguments which are optional.

Command	Description	Syntax
`/`	Root directory	
`./`	Current directory	
`../`	Parent directory of current directory	
`/`	Default home directory	
`man`	Unix manual	`man commandname`
`whatis`	Short description of command	`whatis commandname`
`apropos`	Search manual using keyword	textttapropos keyword
`pwd`	Print working directory	`pwd`
`ls [-al]`	List contents of directory	`ls -altr`
`cat`	Print contents of file to terminal	`cat filename`
`more`	Print scrollable contents of file to terminal	`more filename`
`cd`	Change directory	`cd directory`
`mkdir`	Make new directory	`mkdir directory`
`mv [-i]`	Move, or change name of, files and directories	`mv oldname newname` `mv filelist directory`
`rm [-i]`	Delete files	`rm -i filename`
`rmdir`	Delete directories	`rmdir directory`
`cp [-i]`	Copy files and directories	`cp -i sourcefile newfile`
`grep`	Print lines matching a pattern	`grep pattern filelist`
`ps`	List all current processes by PID	`ps` `ps | grep programname`
`kill`	Kill a process running in the background	`kill pid`
`nano`	Opens Nano (in terminal) text editor	`nano file`