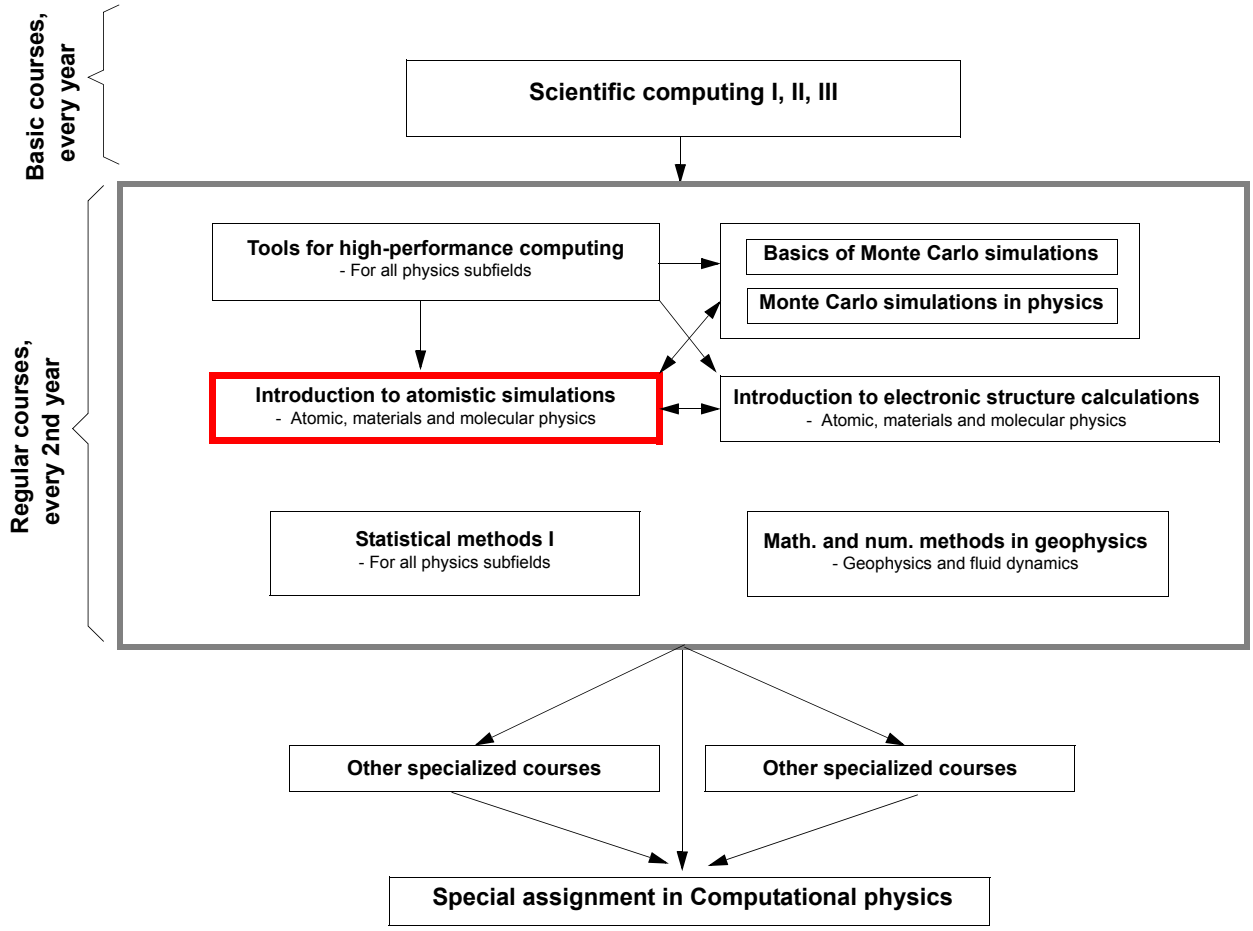


53363 Introduction to atomistic simulations 2008

- Lecturer: university lecturer Antti Kuronen (antti.kuronen@helsinki.fi, tel. 19150083)
- Exercise assistant: the same.

- Lectures: Wed 12-14 room D116
- Exercises: Fri 12-14 room D116

- The idea of the course is to teach the students the basics of atom-level computer simulations, which are widely used in materials and atomic physics, chemistry and biology.
  - The course deals with 2 basic simulation types: molecular dynamics (MD) and structural optimization (by using conjugate gradients (CG) and genetic algorithm (GA) methods).
  - During the first part of the course a basic MD code is written under guidance in several steps during the exercises.
  - After this the MD code is used in applications.
- Course home page: http://www.physics.helsinki.fi/courses/s/atomistiset/

Course evaluation:

- Exercises 50 % of the points
- Exam 50 %

Exercises

- Returned by email to the lecturer no later than the Wednesday before the exercise
- Many of the exercises involve writing subroutines or full computer programs.
  - These will be compiled, and **if they do not compile, the exercise return automatically gives 0 points.**
- The programs are to be returned either in Fortran90 or C.
  - Unix is the preferable environment.
- Return also sample input and output (text or figures).
- Explanations on what you have done preferably in PDF.

Computer environment

- For exercises you need an computer environment with C or Fortran compiler.
- Good non-commercial (i.e. free) alternatives are the GNU compilers.
  - They can be easily installed in any Linux distribution.
- Cluster computer mill.physics.helsinki.fi can also be used.
  - If you want to have a user account on mill contact the lecturer.
  - mill is a 64-processor (32-node) Linux cluster. (A bit old now.)
    - GNU compilers for C (`gcc`) and Fortran (`gfortran`) are installed there.
    - You can run your exercises on the frontend node.
    - Of course, you may use the batch job system to use the computing nodes.
    - For instructions, see http://beam.acclab.helsinki.fi/mill/ . (Note, however, that some of the information on the page is outdated; e.g. there is no Intel Fortran compiler installed.)

Contents (might change a bit)

1. Introduction: Atomistic visulization
2. Basics of MD: Initialization
3. Neighbour lists: **`mdmorse`** code
4. Algorithms to solve equations of motion
5. Force calculation: Basics of potentials
6. Theory: P and T control
7. Quantum mechanical methods (very briefly)
8. Metal interaction models
9. Semiconductor interaction models
10. Molecular interaction models

11 a. Ionic interactions

11 b. How to choose and test a potential ?

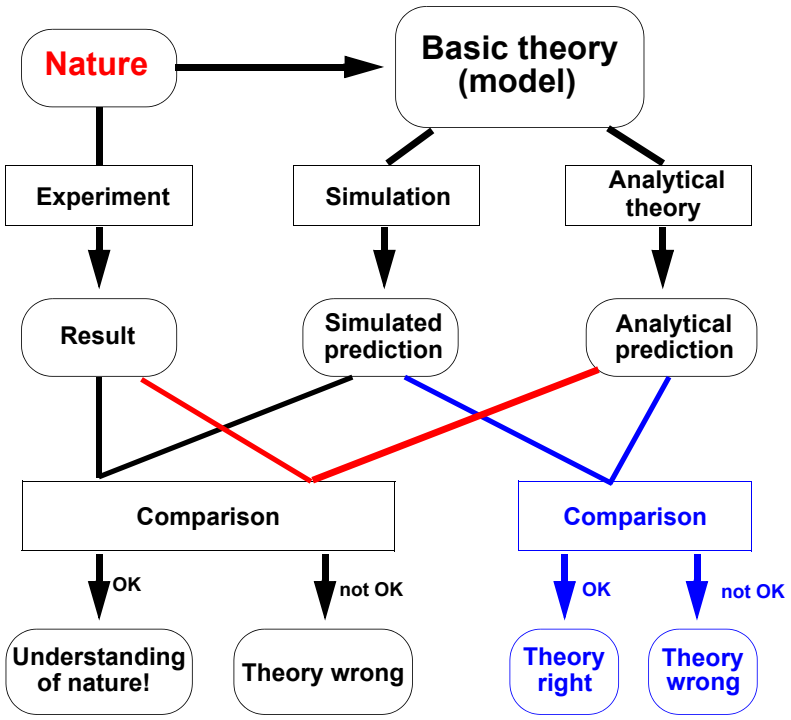
12. Conjugate gradients, genetic algorithms, minimum energy paths
13. Comparison to experiments
14. Summary and end

Literature

- Lecture notes
  - Will appear on the web a bit before the lecture (http://www.physics.helsinki.fi/courses/s/atomistiset/lecturenotes/).
  - The web page also has links to similar courses elsewhere in the world.

- As background information you can use e.g.:
  - M. P. Allen, D. Tildesley: *Computer simulation of Liquids* (Oxford University Press, Oxford,1989)
    - The classical simulation textbook everybody refers to.
    - Statistical mechanics approach.
  - J. von Boehm: *Molekyylidynamiikkamenetelmä* (in Finnish)(Otatieto, Espoo, 2000)
    - Some material on interatomic potentials.
    - Overlaps the Allen-Tildesley book.
  - D. Frenkel, B. Smit: *Understanding Molecular Simulation: From Algorithms to Applications*, 2nd edition (Academic Press, 2001)
    - Statistical mechanics approach.
    - Note that the 1st edition has quite a few printing errors.
    - Book home page (http://molsim.chem.uva.nl/frenkel_smit/) has exercises and case studies.
  - R. Phillips: *Crystals, defects and microstructure : modeling across scales* (Cambridge University Press, 2001)
    - A nice textbook on computational methods in materials research in general; from atomistics to elastic continuum.
    - Includes a chapter on interaction models.
  - A. R. Leach: *Molecular modelling: Principles and applications*, 2nd edition (Prentice Hall, 2001)
    - In addition to simulation methods includes also nice chapters on interaction models (classical and quantum mechanical).
    - Molecular mechanics and force fields.

Computer simulations in physics



- Simulation can bridge the gap between theory and experiment.
- Sometimes only choice (theory too complicated).
- Sometimes simulation impossible: not enough computer capacity.
- Also comparison between analytical theory and simulation: if both are based on the same basic theory (e.g. Newtons laws), but analytical theory uses approximations, simulation can be a perfect way to test the approximation.

Atomistic simulation: What is it?

- Model where the basic object is (roughly) a spherical object.
  - This object can be an
    - atom
    - molecule
    - nanocluster
    - a particle in a fluid
    - a planet or a part of a galaxy
  - On this course, we almost always talk about “atoms”, but in many cases the algorithms are such that the “atom” could be almost any of the above.
- Application areas:
  - atom movement in equilibrium: thermodynamics
  - nonequilibrium phenomena: irradiation, material heat or pressure processing, phase transitions, nucleation, surface growth (thin film deposition)
  - properties of lattice defects
  - nanostructures: $N_{atoms} \sim 10^4 - 10^9$: can be simulated!
  - interactions inside a molecule: vibration, rotation, protein folding
  - intermolecular interactions
  - chemical reactions

And what is it not?

- Continuum modelling (e.g. Finite Element Modelling, FEM)
- Fixed lattice or grid model
  - Although here the limit is sometimes hard to draw.
  - Modeling of amorphous materials using continuous random networks: bond-switch simulations.
- Particle physics
- Electronic structure calculation (for fixed positions of nuclei)
  - But these are often used as basis for atomistic simulation: *ab initio* MD.
- Since the basic object is an atom, and a computer memory is limited, atomistic simulations are always somehow size limited.
  - Hence usually simulating macroscopic (mm size and up) objects is usually out of the question.
  - E.g. silicon: volume/atom $v = \frac{(5.43\text{Å})^3}{8\text{ at.}} = 20.0\frac{\text{Å}^3}{\text{at.}}$

$$V(10^8\text{atoms}) = 2.0\times10^9\text{Å}^3$$

$$\text{cube edge} = \sqrt[3]{2.0\times10^9}\text{Å} = 1260\text{Å} = 0.126\mu\text{m}$$

- Time scale of normal MD limited to tens of nanoseconds (but more about that later).

Important types of atomistic simulations

- Molecular dynamics (**MD**)
  - Simulate the dynamic atom motion based on some interaction model.
- Monte Carlo (**MC**)
  - MC is in the broadest sense any simulation which uses random numbers.
  - Even most MD simulations do use random numbers, but they are still conventionally not considered true MC simulations.
  - There are a few varieties of MC which are often used for atomistic simulations. The most important are maybe:
    - Metropolis MC (**MMC**)
      - Simulate a thermodynamic ensemble, energy minimization by simulated annealing.
    - Kinetic MC (**KMC**)
      - Simulation of activated processes (e.g. diffusion)
  - The MC courses deal with all this. (http://beam.acclab.helsinki.fi/~eholmstr/mc/)
- Structural optimization
  - Find the equilibrium state of of an atomistic system based on some interaction model: energy minimization.
  - Global vs. local minimum: simulated annealing.
  - Conjugate Gradient (**CG**) method
    - An efficient way to find a local minimum.
    - Can also be used for atoms.
  - Genetic algorithms (**GA**)
    - Sometimes an efficient way to find a global minimum.
    - Can also be used for atoms.
  - Minimum energy path determination

- Binary collision approximation (BCA)
  - In nuclear and ion beam physics (and almost nowhere else)
  - Event-driven simulations in general (e.g. interaction of electron and photon radiation with matter)

How to present atomistic data

- There exist about a zillion different file formats for presenting atom positions.
  - An example: how should we present the coordinates?
    - For 8 Cu atoms in the corners of the unit cube
    - Trivial format 1 “x y z”:

```
0.0 0.0 0.0
1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0
1.0 1.0 0.0
1.0 0.0 1.0
0.0 1.0 1.0
1.0 1.0 1.0
```

    - No information about time (for a dynamic system)
    - Trivial format 2: “x y z t”

```
0.0 0.0 0.0 3.0
1.0 0.0 0.0 3.0
0.0 1.0 0.0 3.0
0.0 0.0 1.0 3.0
1.0 1.0 0.0 3.0
1.0 0.0 1.0 3.0
0.0 1.0 1.0 3.0
1.0 1.0 1.0 3.0
```

    - Downside of both formats: All file has to be read in before we know how many atoms there are.

From program `babel`:

```
alc       -- Alchemy file
bgf       -- BGF file
bmin      -- Batchmin Command file
box       -- DOCK 3.5 box file
bs        -- Ball and Stick file
c3d1      -- Chem3D Cartesian 1 file
c3d2      -- Chem3D Cartesian 2 file
caccrt    -- Cacao Cartesian file
cache     -- CAChe MolStruct file
cacint    -- Cacao Internal file
cdct      -- ChemDraw Conn. Table file
contmp    -- Conjure Template file
csr       -- MSI Quanta CSR file
cssr      -- CSD CSSR file
diag      -- DIAGNOTICS file
dock      -- Dock Database file
dpdb      -- Dock PDB file
feat      -- Feature file
fhz       -- Fenske-Hall ZMatrix file
gamin     -- Gamess Input file
gcart     -- Gaussian Cartesian file
gotmp     -- Gaussian Z-matrix tmplt
gr96A     -- GROMOS96 (A) file
gr96N     -- GROMOS96 (nm) file
gzmat     -- Gaussian Z-matrix file
hin       -- Hyperchem HIN file
icon      -- Icon 8 file
idatm     -- IDATM file
m3d       -- M3D file
maccs     -- MDL Maccs file
macmod    -- Macromodel file
macmol    -- Mac Molecule file
mdl       -- MDL Molfile file
micro     -- Micro World file
miv       -- MolInventor file
mm2in     -- MM2 Input file
mm2out    -- MM2 Ouput file
mm3       -- MM3 file
mmads     -- MMADS file
mol       -- Sybyl Mol file
mol2      -- Sybyl Mol2 file
mopcrt    -- Mopac Cartesian file
mopint    -- Mopac Internal file
pcmod     -- PC Model file
pdb       -- PDB file
psc       -- PS-GVB Cartesian file
psz       -- PS-GVB Z-Matrix file
report    -- Report file
sdf       -- MDL Isis SDF file
smiles    -- SMILES file
spar      -- Spartan file
tinker    -- Tinker XYZ file
torlist   -- Torsion List file
unixyz    -- UniChem XYZ file
wiz       -- Wizard file
xed       -- XED file
xyz       -- XYZ file
```


How to present atomistic data

- In this course we use the **XYZ standard format**.
  - First line has number of atoms at this time
  - Second line is comment
  - Then come the coordinates of the atoms with the element symbol as the 1st column.

```
 8
Molecule name or comment or whatever (Might, however, be used by some applications.)
Cu 0.0 0.0 0.0 320.0
Cu 1.0 0.0 0.0 310.0
Cu 0.0 1.0 0.0 305.0
Cu 0.0 0.0 1.0 280.0
Cu 1.0 1.0 0.0 290.0
Cu 1.0 0.0 1.0 320.0
Cu 0.0 1.0 1.0 310.0
Cu 1.0 1.0 1.0 320.0
```

  - The fifth column can also hold other information, or be empty.
  - It is a very good idea to include useful information on the second line (a non-standard feature), e.g.

```
 8
Frame number 1 3.0 fs boxsize 3.0 3.0 3.0
Cu 0.0 0.0 0.0 320.0
Cu 1.0 0.0 0.0 310.0
...
```

How to present atomistic data

- For dynamic information, the info for each time can simply be put after each other in the same file:

```
 2
Frame number 1 0.0 fs boxsize 3.0 3.0 3.0
Cu 0.0 0.0 0.0 320.0
Cu 1.0 0.0 0.0 310.0
 2
Frame number 1 2.0 fs boxsize 3.0 3.0 3.0
Cu 0.1 0.0 0.0 330.0
Cu 1.1 0.1 0.0 300.0
 2
Frame number 1 4.0 fs boxsize 3.0 3.0 3.0
Cu 0.2 0.1 0.0 340.0
Cu 1.2 0.1 0.0 290.0
...
```

- For very large simulation systems this text format may become too inefficient (both from the point of view of space and time).
  - Binary format: standards?

Visualization of atomic data

- Visualization is fun but also useful.
- Plot each atom as a sphere, either statically or dynamically.
- Plot bonds between atoms: ball-and-stick model.
- As with file formats, there are about a zillion programs which can do that.
- One much used visualization program is **RasMol**. It is

  \+ free and open source (easy to modify for your needs)
  \+ works at least in Linux, Unix, Windows, and Mac
  \+ fast
  \+ supports many of the most common chemistry formats, including XYZ
  \+ can produce publication-quality output
  \- poor at adding text to the graphics
  \- can not draw much else than atoms, bonds and protein backbones
  \- no perspective transformation

  - Home page: http://rasmol.org/

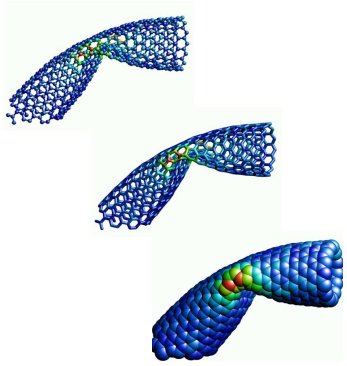
- `rasmol` is installed on the computing cluster mill.physics.helsinki.fi.
  - Run it by command: `rasmol`
  - Program is located at `/usr/local/bin` and this should be on your `$PATH`.

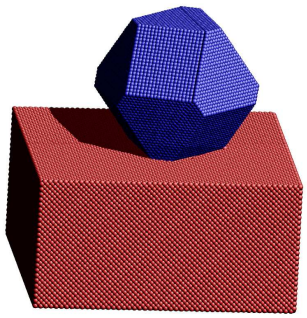
Visualization of atomic data

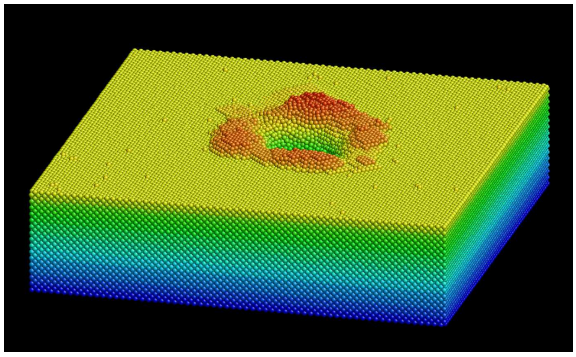
- Useful Rasmol commands (see also http://www.physics.helsinki.fi/courses/s/atomistiset/refcardUS.pdf):

`load xyz file`	Read in a file
`write gif image.gif`	Store an image in the gif format
`write ppm image.ppm`	Store an image in the ppm format
`write ps image.ps`	Store an image in the ps format
`zap`	Remove all data, needed before new load command
`quit`	
`wireframe <on/off/value>`	Adjust bond width
`spacefill <on/off/value>`	Adjust atom size
`spacefill temperature`	Get atom size from column 5 in XYZ file
`zoom 150`	Zoom display, default=100
`set ambient <value>`	Ambient light strength
`set specular on`	Use a nice 3D shade on atoms
`set specpower <value>`	Remove the 3D shade
`set shadows on/off`	Use/don't use atom shadowing (slow)

When started, rasmol reads the file `~/.rasmolrc` for initial settings.







"Quick and dirty"-animations

- A simple way to make animations is to use program `dpc`[1].
- It reads XYZ files that have many frames concatenated.
- `dpc` is also installed on the computing cluster mill.physics.helsinki.fi.
  - Run it by command: `dpc`
  - It is also located at `/usr/local/bin`
  - Basic usage for XYZ files:

```
dpc xyz erase sort 2 3 4 5 moviefile.xyz
```

    - "`2 3 4`" tells that the x, y- and z- info is taken from columns 2, 3 and 4.
    - "`5`" does not mean anything for XYZ, but must still be there.
    - All options are between "`dpc`" and "`2 3 4 5`"

- The program is not as flexible as `rasmol`, but it is reasonably fast (it is written under plain `Xlib`)

- Help is given by command `dpc` with no options.

- Most important options

`s 600 800`	Window size
`sd 600 600`	Draw area size
`x 0 83`	x limits
`y 0 65`	y limits
`z 37.8 43.6`	Color (z) limits
`m 1`	Form of atoms: 0 rect, 1-4 circle
`d 4`	Dot size
`gifdump`	Make gif file dump of each window frame

1. By K. Nordlund

Making presentation animations

- To make animations for the web or a presentation:
  - Use `rasmol` or `dpc` to make a separate a bitmap image of every single time step.
    - Each image should be stored with a name having the frame number encoded; e.g. `frame001.png`, `frame002.png`, ...
  - Make an animation from these separate frames by using any animation program (many of them available freely; e.g. `ffmpeg`).

"Quick and dirty" data-analysis: awk

- Swiss army knife of Unix: `awk` (or `gawk`)
  - A lot can be done using simple 'one-liners'.
  - Example: we have a XYZ file:

```
2632
Time (fs)       74500
C       3.14286         5.13682         9.99465         -7.30347
C       3.54844         3.00536         11.1538         -4.55679
C       4.20179         5.13682         12.1936         -7.30347
C       5.07013         3.00536         13.0619         -4.55679
C       6.10993         5.13682         13.7152         -7.30347
C       7.26903         3.00536         14.1208         -4.55679
C       8.48933         5.13682         14.2583         -7.30347
C       9.70963         3.00536         14.1208         -4.55679
```

  - And want to check the potential energy (5th column or so-called temperature column) distribution:

```
cat file.xyz |
gawk '$1=="C" {i=int(10*$5+0.5); e[i]++} END {for (i in e) print i/10,e[i]}'|
sort -n | xgraph
```

- Quick and dirty plotting: `xgraph`
  - This is also installed on mill at `/usr/local/bin`.
- These tools reduce the need to build C or F90 programs or to launch Matlab for every small task.

Other visualization programs

- **OpenDX**
  - A commercial IBM program package which was later made open source
  - Philosophy
    - build a network of modules through which data flows
    - data analysis and visualization in the same program package
  - Home page: http://www.opendx.org/
- **VMD**
  - More features than in RasMol.
  - Slows down for large systems.
  - Home page: http://www.ks.uiuc.edu/Research/vmd/
- And many many more...

On-line visualization

- Visualize the simulation run in realtime
  - It possibly change the simulation settings during the run.
  - A simple 2D Lennard-Jones system: J. Merimaa, L.F. Perondi, K. Kaski, An interactive simulation program for visualizing complex phenomena in solids, *Computer Physics Communications* **124** (2000) 60-75.

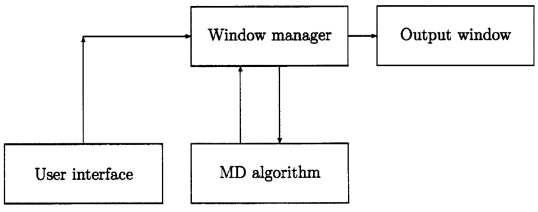


Fig. 1. Schematic structure of an interactive simulation program. The window manager handles the exchange of data among the different modules.

- For example see: http://www.acclab.helsinki.fi/~aakurone/boundary2d/