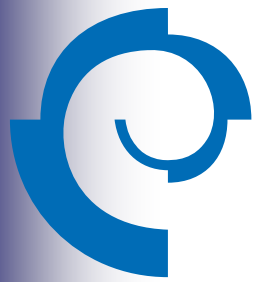
From nano to macro: Introduction to atomistic modeling techniques

IAP 2006

Introduction to the problem set

Lecture 5



Department of
Civil & Environmental Engineering
Massachusetts Institute of Technology

Markus J. Buehler

Room 1-272

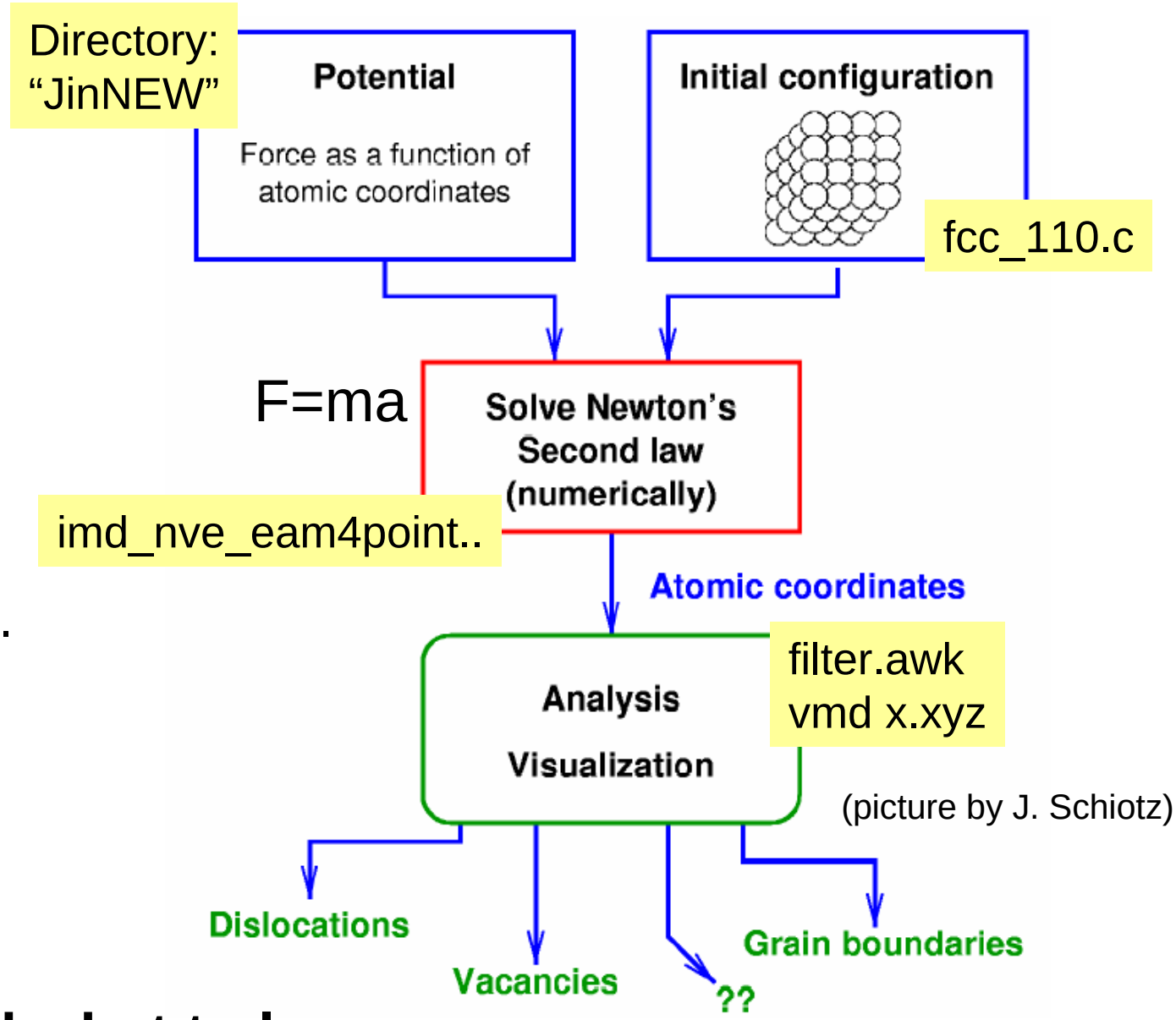
Email: mbuehler@MIT.EDU

Problem set

- Problem set will be sent by email
- Contains:
  - C code to generate initial crystal structure
    fcc_110.c, suitable for perfect crystal and cracked system
  - File “inp_perfect_crystal” with example parameters for fcc_110.c
  - IMD code (MD code), compiled for different potentials and different integration schemes
  - Example input file (Tr_ELASTIC) prepared for EAM potential and for LJ potential
  - Chkpt-to-xyz conversion code “chkpt2xyz.py”
  - no_velocities.awk (script to remove velocities)
  - filter.awk (filter high energy atoms)
  - “src” subdirectory contains the IMD source code
- References for programming:
  - http://www.physics.drexel.edu/courses/Comp_Phys/General/C_basics/c_tutorial.html (online C tutorial)
  - http://www.wag.caltech.edu/home/rpm/python_course/ (Python short course)
  - And many others

Typical simulation procedure

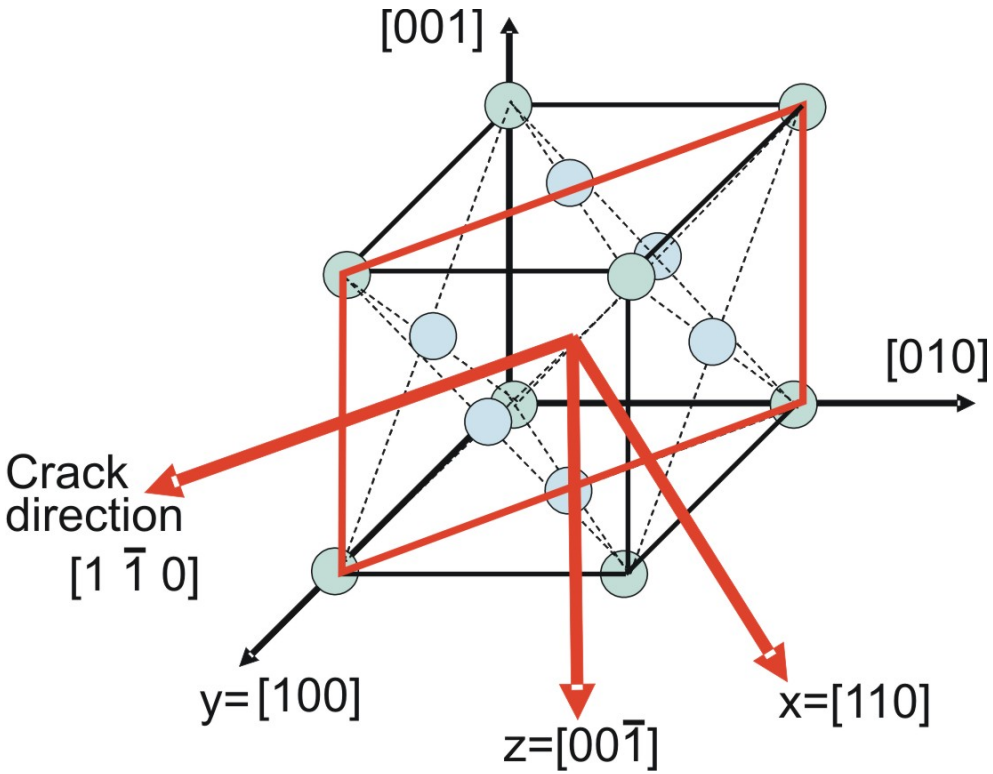
1. Pre-processing (define geometry, build crystal etc.)
2. Energy relaxation (minimization)
3. Annealing (equilibration at specific temperature)
4. “Actual” calculation; e.g. apply loading to crack
5. Analysis



Real challenge:
Questions to ask and what to learn

Pre-processing

- The code fcc_110.c creates a copper single crystal with the following crystal orientations:



Pre-processing

- fcc_110.c is a C code that can be compiled as follows:

```
gcc -O2 -o el110 fcc_110.c –lm
```

GNU compiler

- To run the code, type

```
./el110
./el110 < input
```

Pre-processing

- The program fcc_110.c may be used to create a small perfect crystal by not creating a crack
- You may also modify the program, if desired, for example to generate crystals with different orientations or other crystal structures
- The code can also be used to create nanostructures such as nanowires

GNUPLOT

- GNUPLOT is an easy-to-use tool to generate plots, or e.g. snapshots of a simulated system

- Type “gnuplot” and then

```
plot [x0..x1][y0..y1] ‘out.chkpt’ u 4:5 w d
```

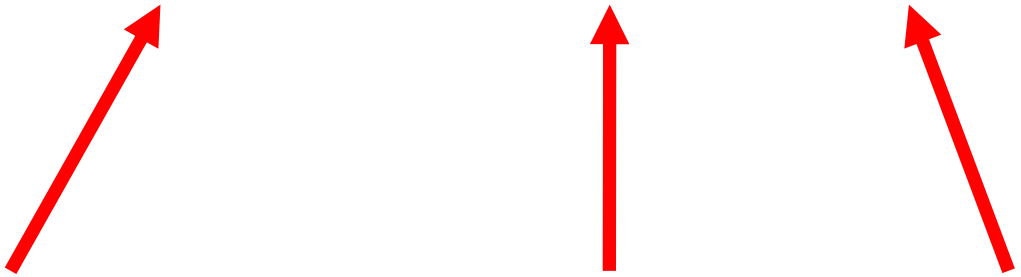
The ITAP MD code

http://www.itap.physik.uni-stuttgart.de/~imd/

- The directory “src” contains the IMD source code

- To compile, type

```
gmake clean
gmake imd_nve_eam4point_deform_stress
```



Program name indicates features activated
(conditional compilation)

Parameter file - EAM

```
simulation      1

coordname       OUT
outfiles        OutFile1000  #

ensemble        nve
startstep       0
maxsteps        200000

timestep        0.25

total_types     3
ntypes          1

core_potential_file     ./JinNEW/cu_pair.dat
embedding_energy_file   ./JinNEW/cu_emb.dat
atomic_e-density_file   ./JinNEW/cu_den.dat

box_x   315.619101 0.00 0.00
box_y   0.00 127.809551 0.00
box_z   0.00 0.00 7.230000

restrictionvector 1 0 0 0  # substrate fixed
restrictionvector 2 0 0 0  # substrate fixed

pbc_dirs        0 0 1  # PBCs in z

max_deform_int  20
deform_shift    1  -0.05 0.  0.
deform_shift    2   0.05 0.  0.

eng_int         5
checkpt_int     100

starttemp       0.0000000000001

#cpu_dim         4 1 8

seed            308989
```

Input file coordinates

Time step

Potential (EAM)

Box size

Constraints (fixed atoms)

Displacement BCs

Start temperature

Parameter file - LJ

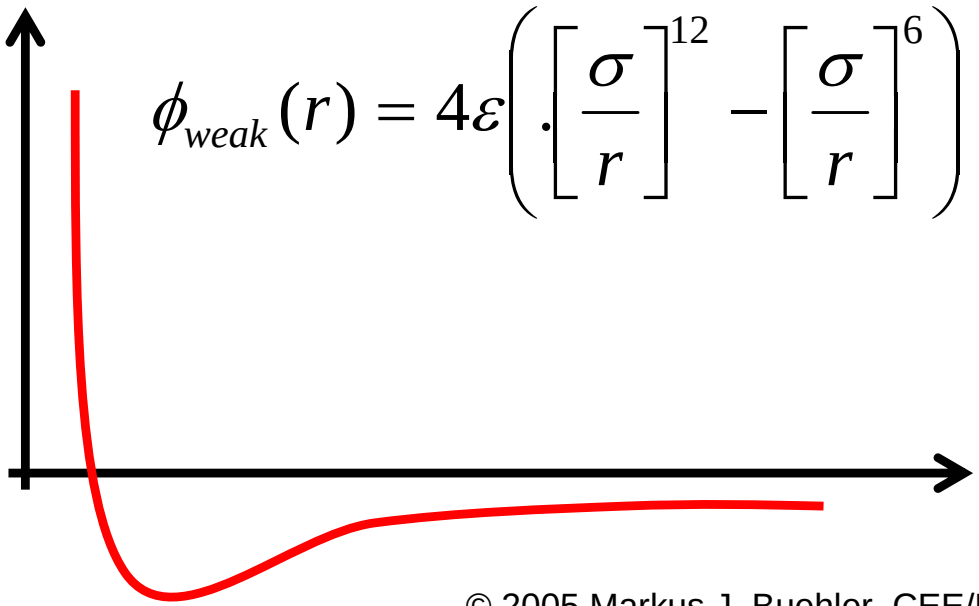
```
#core_potential_file     ../JinNEW/cu_pair.dat
#embedding_energy_file   ../JinNEW/cu_emb.dat
#atomic_e-density_file   ../JinNEW/cu_den.dat

r_cut            5.38 # between 3rd and 4th nearest neighbor
#r_cut           3.4
r_begin          0.4
pot_res          1000

lj_epsilon       0.167
lj_sigma         2.314   # CLERI

#lj_epsilon      0.161
#lj_sigma        2.277
##lj_epsilon     0.35
```

Information re. potential table

Potential parameters

Lennard-Jones parameters

$$\phi_{weak}(r) = 4\varepsilon\left(\left[\frac{\sigma}{r}\right]^{12} - \left[\frac{\sigma}{r}\right]^{6}\right)$$



Remarks

- List of all parameters of IMD

  http://www.itap.physik.uni-stuttgart.de/~imd/userguide/params.html

Output files

- OutFileXXXX.eng: Energy information, average stress in system etc.

```
# time Epot temperature pressure volume Press_xx Press_yy Press_zz
Press_yz Press_xz Press_xy
```

- OutFileXXXX.00YYY.chkpt: Checkpoint files

```
#  number type mass x y z vx vy vz Epot
```

- Include conversion tools, such as

```
python chkpt2xyz.py out.chkpt x.xyz rescale_factor
```

```
vmd –xzy x.xyz
```

filter.awk

```
#! /bin/awk -f
{}{
OFMT = "%.16f";
CONVFMT ="%.16f";
if($1 !~ /#/)
if($10 > -3.45)
printf("%d %d %.16f %.16f %.16f %.16f\n",$1,$2,$3,$4-
  0,$5-0,$6);
}
```

more OutFile1000.00001.chkpt | ./filter.awk > OUT

Make sure “filter.awk” is executable (chmod +x filter.awk)

“no_velocities.awk” is used in the same way

Interatomic potentials

- The potential used:

- LJ 12:6 potential

$$\phi(r) = 4\varepsilon\left(\left[\frac{\sigma}{r}\right]^{12} - \left[\frac{\sigma}{r}\right]^{6}\right) \qquad r_0 = \sqrt[6]{2} \approx 1.12246$$

$$\sigma = a / \sqrt{2} / \sqrt[6]{2}$$

  See also Cleri *et al.*

- EAM potential for copper
  - Mishin *et al*., 2001, Phys. Rev. B http://prola.aps.org/pdf/PRB/v63/i22/e224106
  - Potential EAM 1 (new potential described in that paper)