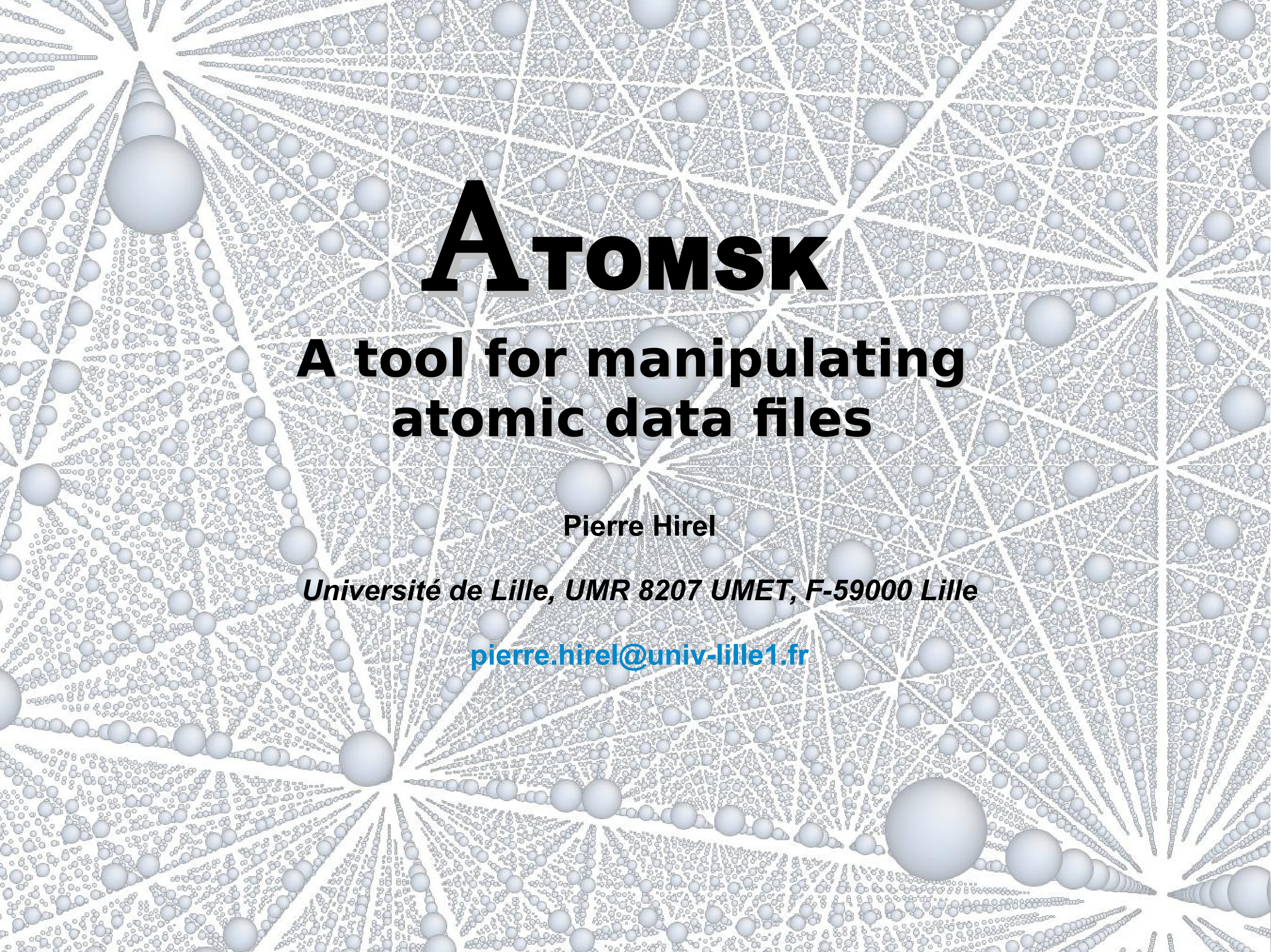


Atomsk

A tool for manipulating atomic data files

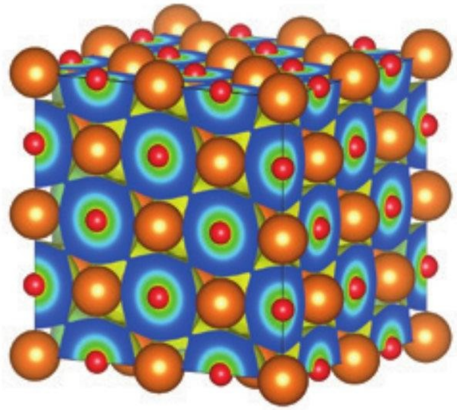
Pierre Hirel

Université de Lille, UMR 8207 UMET, F-59000 Lille

pierre.hirel@univ-lille1.fr

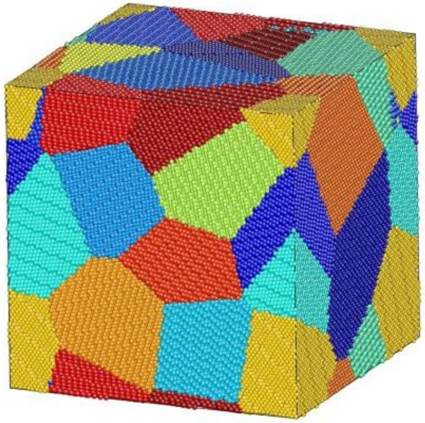
Supported file formats/softwares

Ab initio



Quantum Espresso
SIESTA
VASP

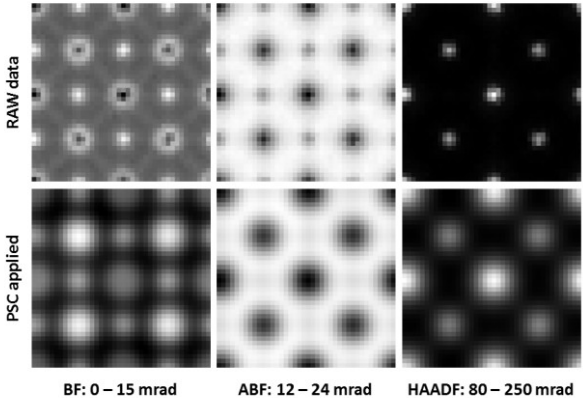
Classical atomistics



DL_POLY
IMD
GULP

LAMMPS
MOLDY
XMD

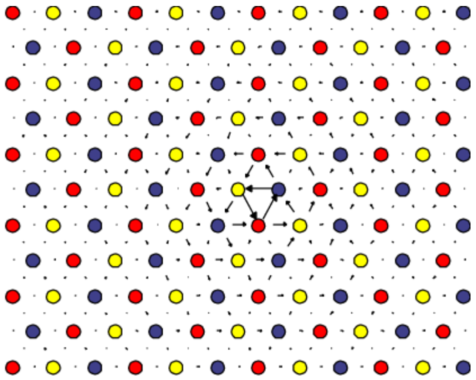
TEM image simulation



Dr Probe
JEMS
QSTEM

Visualization

Atomeye
ddplot
OVITO
XCrySDen
VESTA



Main features:
- Conversion from any to any other format
- Construction of unit cells, supercells
- Transformations: duplicate, cut, shear, rotate...
- Insertion of defects (dislocations, cracks…)
- Complex systems: polycrystals…
- Analysis: RDF, Nye tensor...

What can Atomsk do?

Atomsk can:

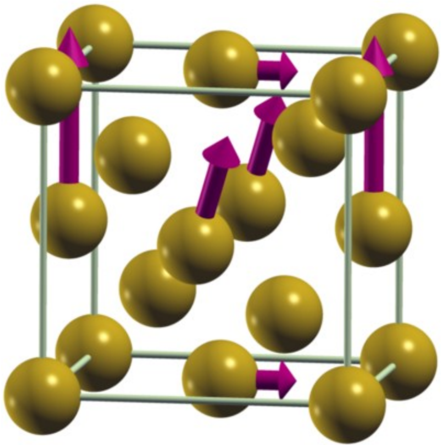
- Manipulate and transform **atom positions**
- Convert and prepare **text files** for simulations or visualization
- Duplicate a system, cut crystal planes, create dislocations...
- Construct bicrystals, polycrystals…
- Merge systems (e.g. system on top of a substrate)
- Run **in command-line** in Windows, Mac OS X, Linux
- Be integrated in scripts (bash, etc.)

Atomsk can not:

- Run simulations
- Visualize atoms, make images or illustrations
- Deal with electronic density
- Run through a nice interface (command-line only!)
- Guess what you want to do

Working with *ab initio* software

VASP



1. Convert your atomic system for VASP

→ Design a system from scratch with Atomsk

→ Convert an existing system into VASP format

```
atomsk  my_system.xsf   POSCAR
```

2. Run VASP

3. Convert VASP output files for visualization

→ Conservation of atomic forces, energies...

```
atomsk  CONTCAR  final.vesta
```

Quantum Espresso / pwscf

1. Convert your atomic system for PWscf

```
atomsk  my_system.xsf   -unit A Bohr  pw
```

2. Run pwscf

3. Convert output file for visualization

→ Convert all intermediate steps

```
atomsk --all-in-one  pw.out   -unit Bohr A  xsf  cfg
```

Working with atomistic simulations

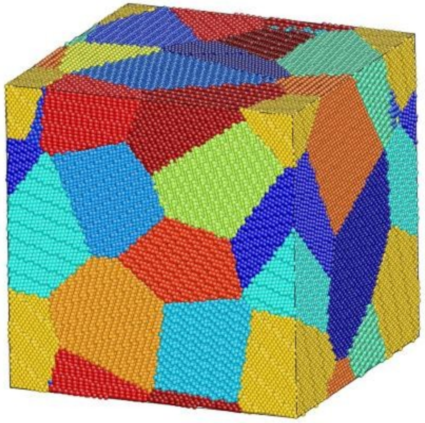
LAMMPS

1. Convert your atomic system for LAMMPS

→ Create a system from scratch with Atomsk

→ Convert from an *ab initio* file

```
atomsk  CONTCAR   lammps
```



2. Run LAMMPS

3. Convert LAMMPS output files for visualization

→ Conservation of atomic forces, energies...

```
atomsk  final.xyz  xsf  cfg
```

Example:
– Build a complex system, far from equilibrium (e.g. grain boundary, inclusion...)
– Perform quick-and-dirty relaxation with force-field (LAMMPS)
→ atoms are closer to their ideal positions
– Convert system for ab initio (VASP) → only a few steps are necessary

Working with TEM image simulation

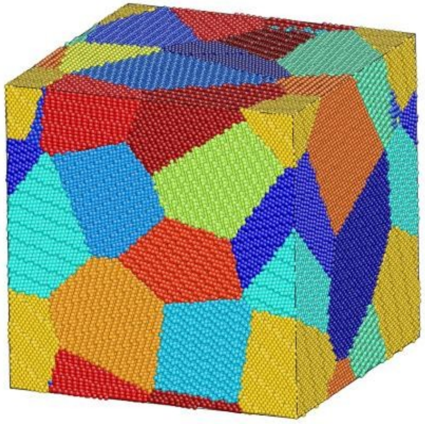
JEMS

1. Generate atomic system for JEMS

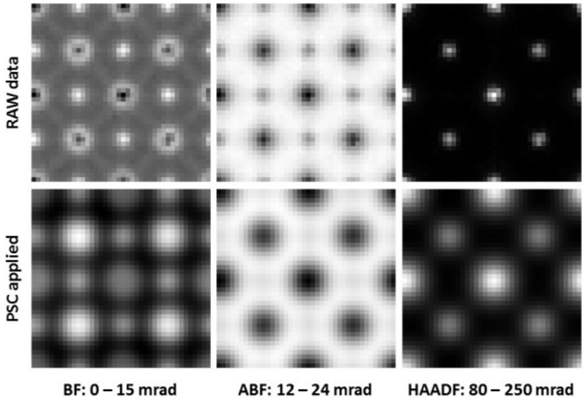
→ Design a system from scratch with Atomsk

→ Convert atomic system from an *ab initio* or force-field simulation

```
atomsk  relax.lmc   jems
```



2. Open file in JEMS and work on it



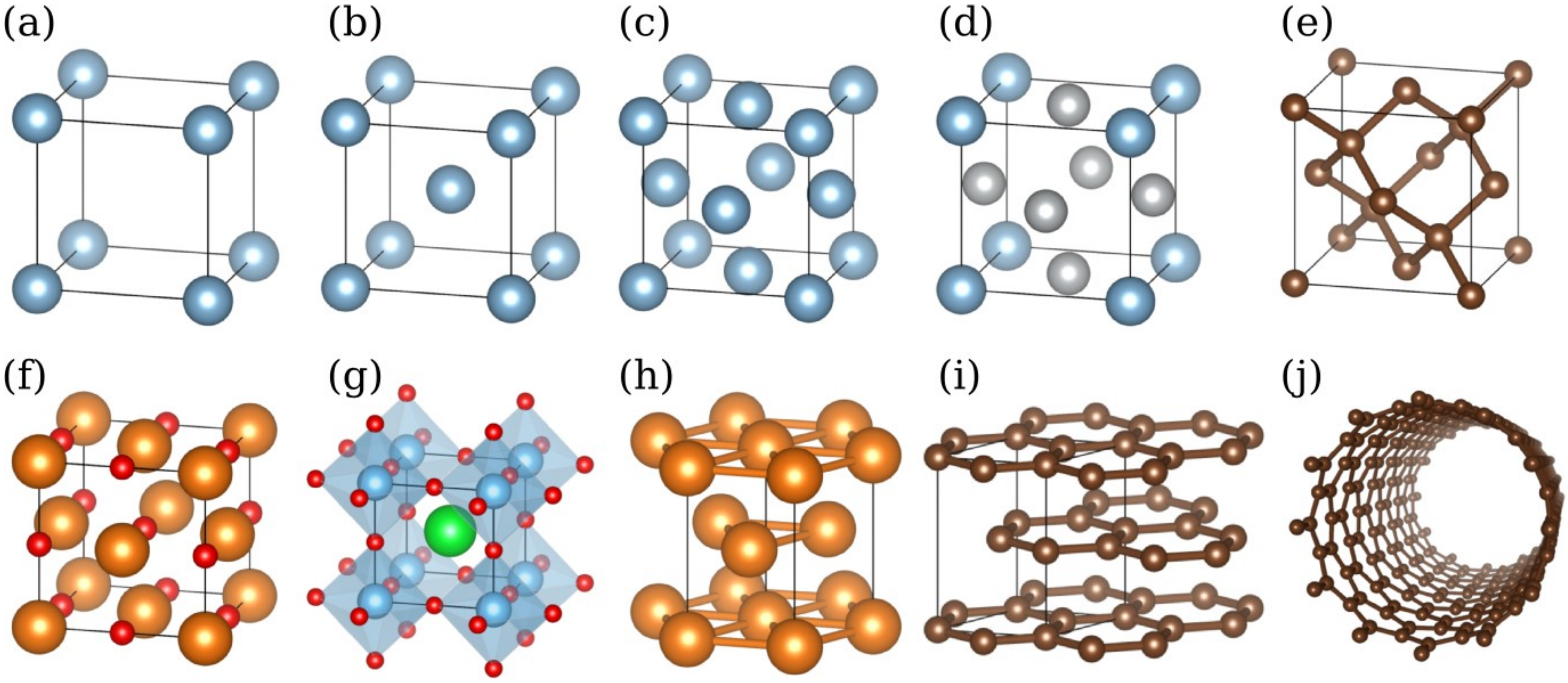
Create unitcells

Lattice type ↓ Lattice constant ↓ Atom species ↓

```
atomsk --create  fcc       4.02  Al        aluminium.xsf
```

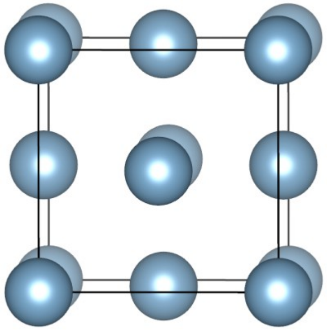
```
atomsk --create  bcc       3.86  Fe        iron.cfg
```

```
atomsk --create  rocksalt 3.86  Mg O    periclase.xsf
```

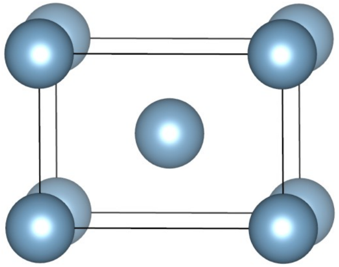


Orient unitcells

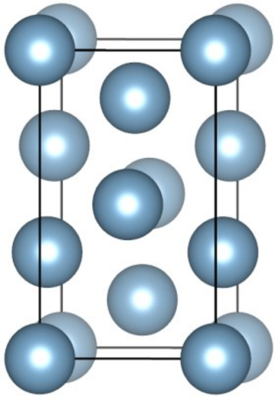
```
atomsk --create  fcc  4.02 Al  orient [100] [010] [001] aluminium.xsf
```



```
atomsk --create  fcc  4.02 Al  orient [001] [110] [1-10] aluminium.xsf
```

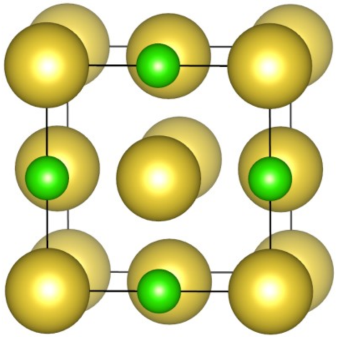


```
atomsk --create  fcc  4.02 Al  orient [0-11] [211] [1-1-1] aluminium.xsf
```

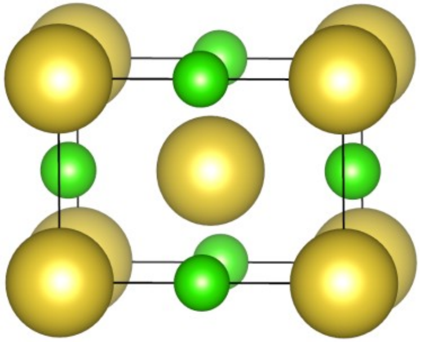


Orient unitcells

All cubic lattices can be oriented:
bcc, fcc, diamond, rocksalt, perovskite...

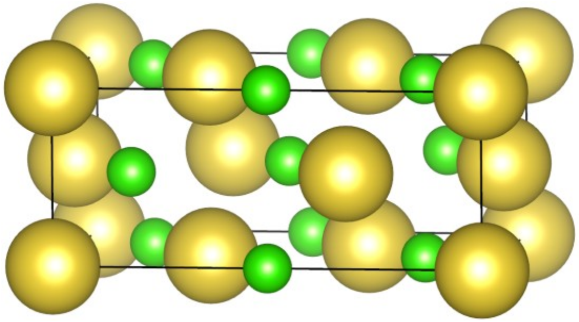


```
atomsk --create rocksalt 4.02 NaCl orient [100] [010] [001] NaCl.xsf
```



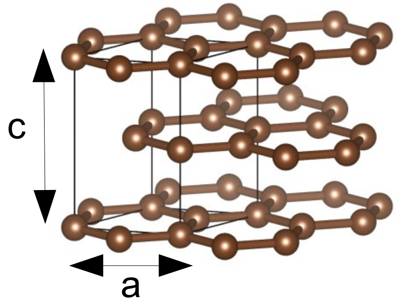
```
atomsk --create rocksalt 4.02 NaCl orient [001] [110] [1-10] NaCl.xsf
```

```
atomsk --create rocksalt 4.02 Na Cl orient [1-1-1] [0-11] [211] NaCl.xsf
```



Generate nanotubes

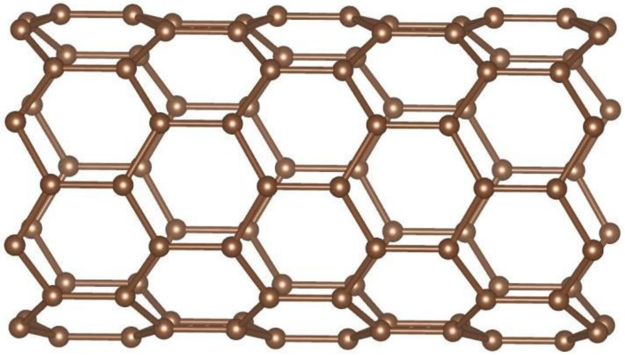
Mode: create



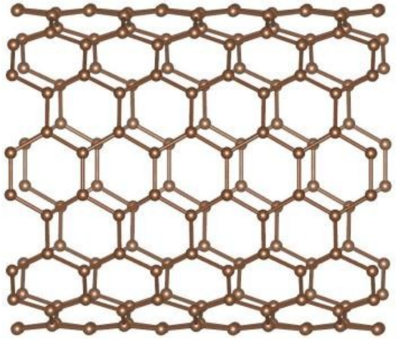
```
atomsk --create nanotube 3.6  6  0  C  carbon_NT.xsf
```

a   m   n   Chemical symbol(s)

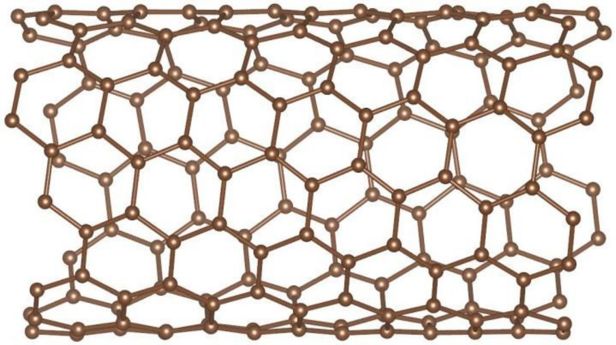
Carbon



(8,0) zigzag nanotube

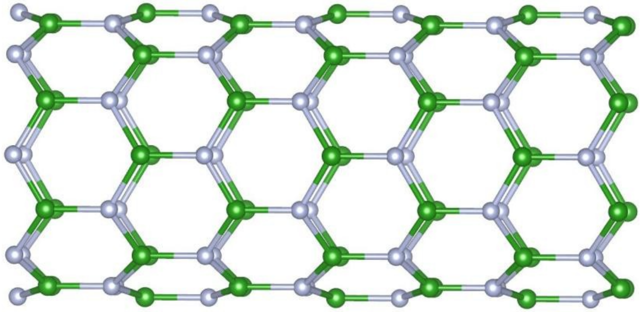


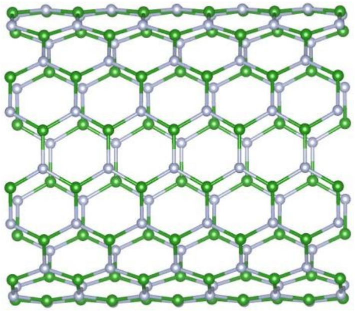
(8,8) armchair nanotube

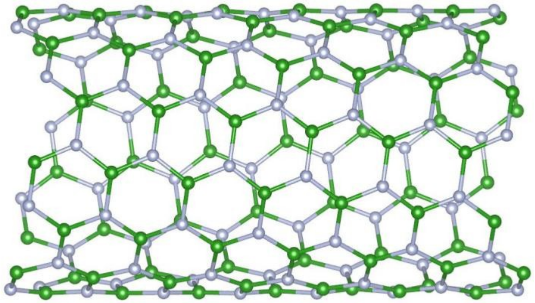


(8,5) chiral nanotube

Boron nitride

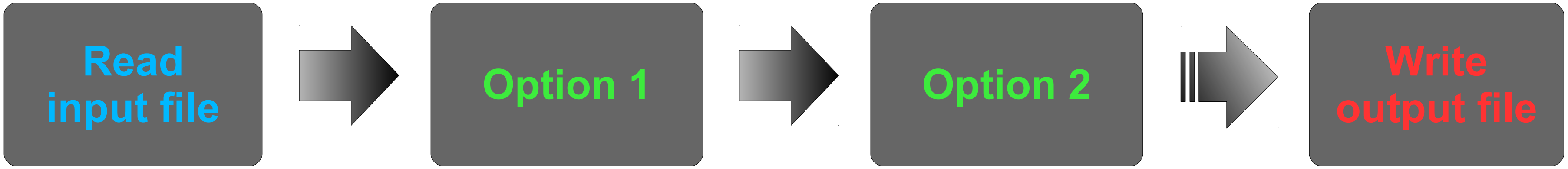






Options

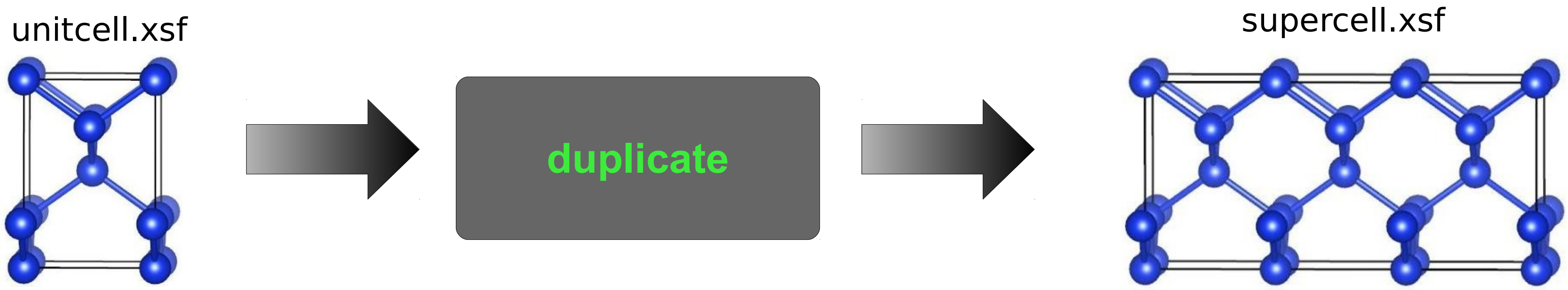
- Options allow to transform a system
- Each option is an **elementary transformation**
- Several options can be applied sequentially to design a system



Create supercells

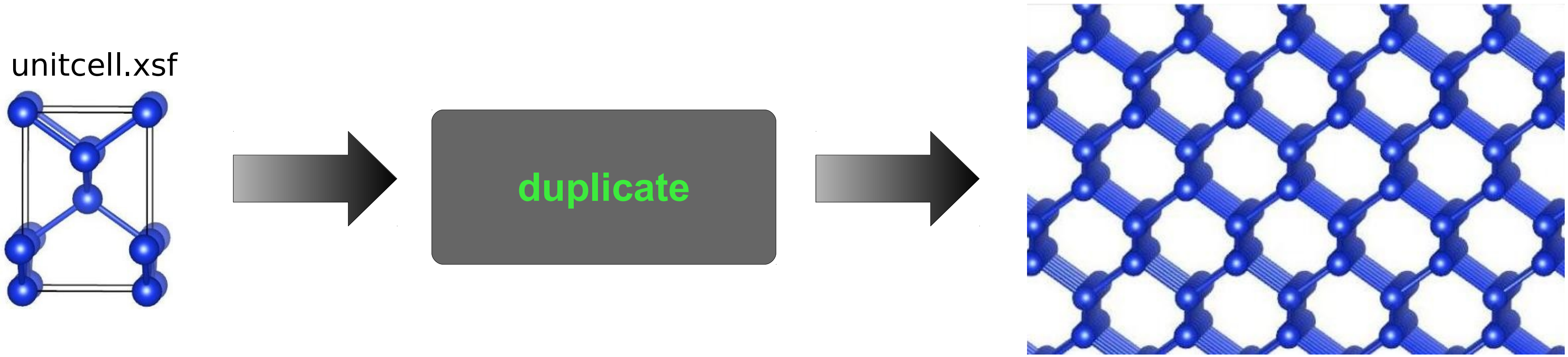
Option: -duplicate Nx Ny Nz

```
atomsk unitcell.xsf -duplicate 3 1 1 supercell.xsf
```



```
atomsk unitcell.xsf -duplicate 10 10 6 supercell.xsf
```

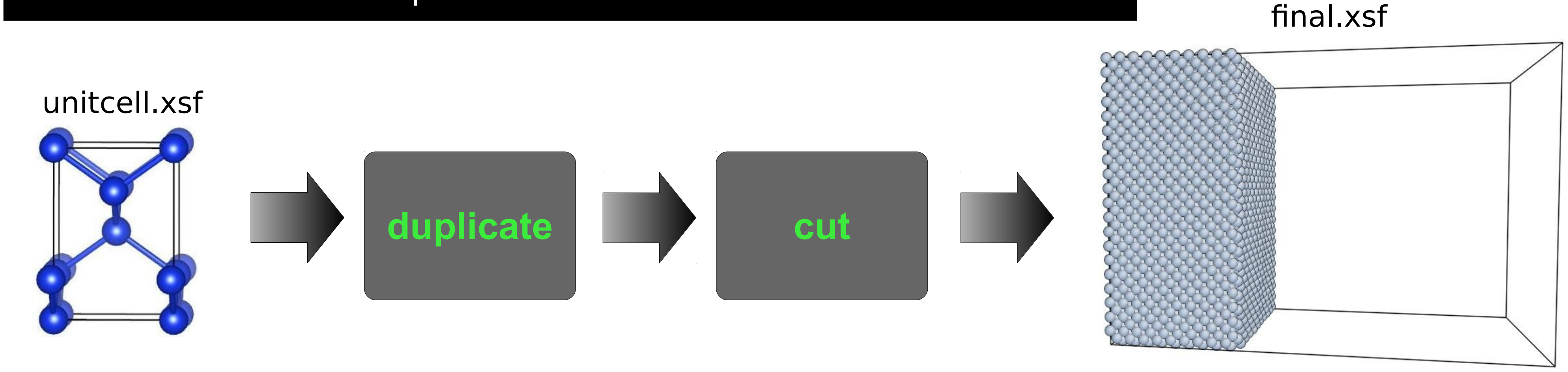
supercell.xsf



Cut atomic planes

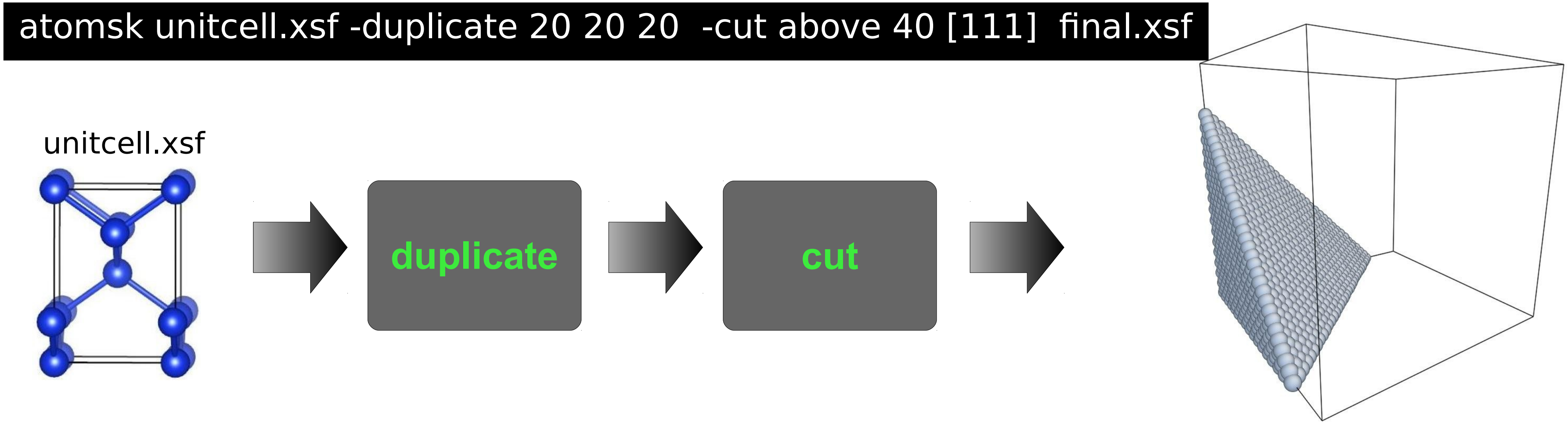
Option: -cut

```
atomsk unitcell.xsf  -duplicate 30 20 10  -cut above 30 x  final.xsf
```



Possibility to use Miller notation:

```
atomsk unitcell.xsf -duplicate 20 20 20  -cut above 40 [111]  final.xsf
```



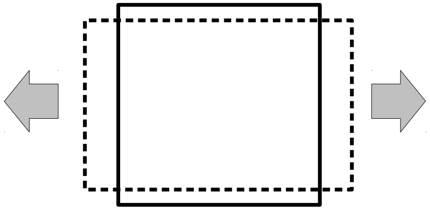
Deform a system

Option: -deform <X|Y|Z> ε ν

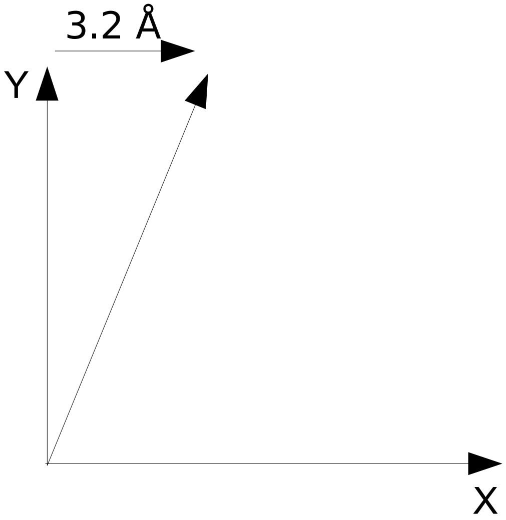
```
atomsk initial.xsf -deform X  0.02  0.3   final.xsf
```

```
atomsk initial.xsf -deform Y  1.5%  0.3   final.xsf
```

```
atomsk initial.xsf -deform Z  -2.1%  0.3   final.xsf
```



Option: -shear <X|Y|Z> ε <X|Y|Z>



```
atomsk initial.xsf -shear  Y  3.2  X   final.xsf
```

```
atomsk initial.xsf -shear  Y  1.5%  Z   final.xsf
```

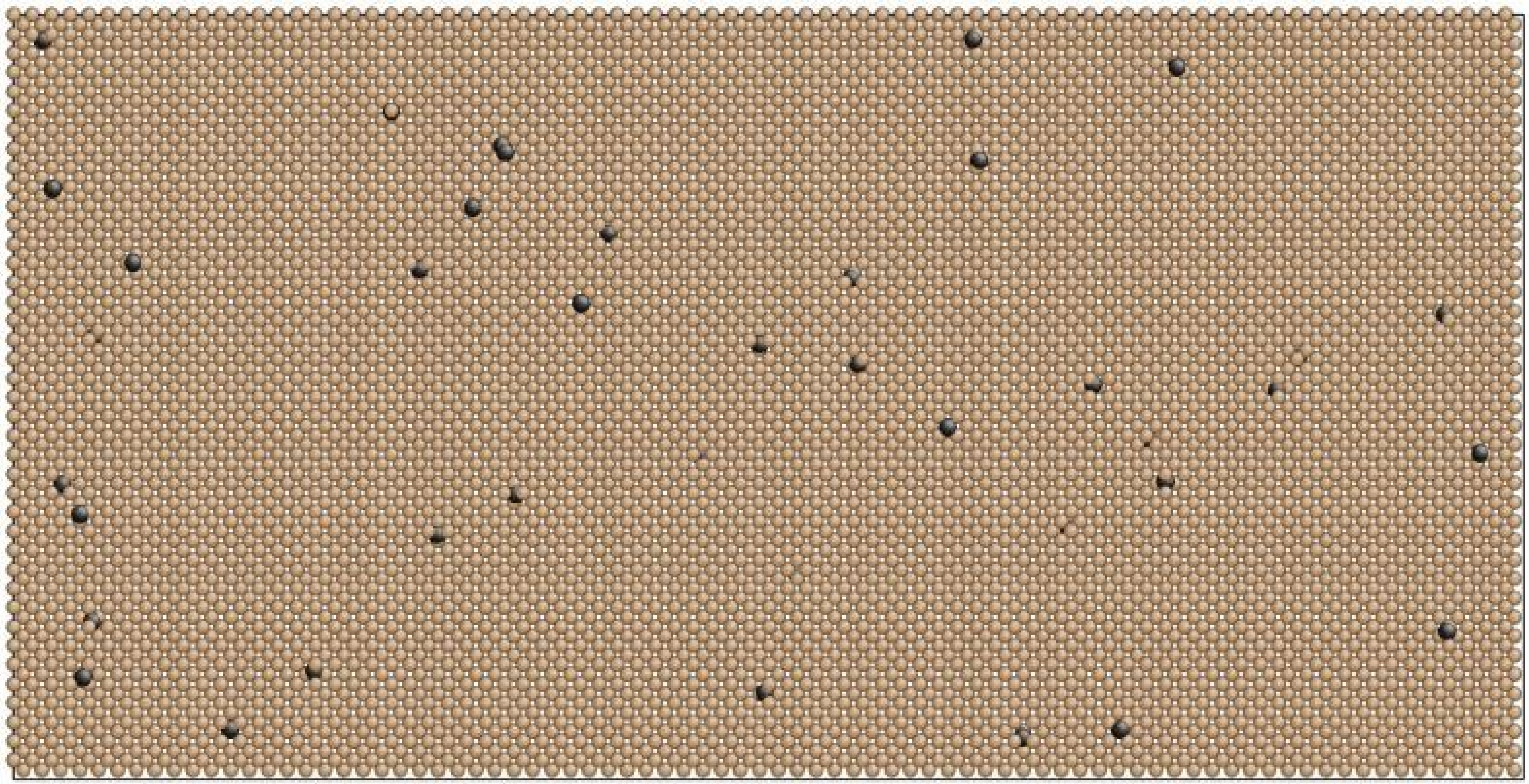
Add atoms

Option: -add-atom

- Add new atoms at random positions

  → Positions adjusted into tetrahedral sites to avoid overlap

```
atomsk  initial.cfg  -add-atom C random 40  final.cfg
```



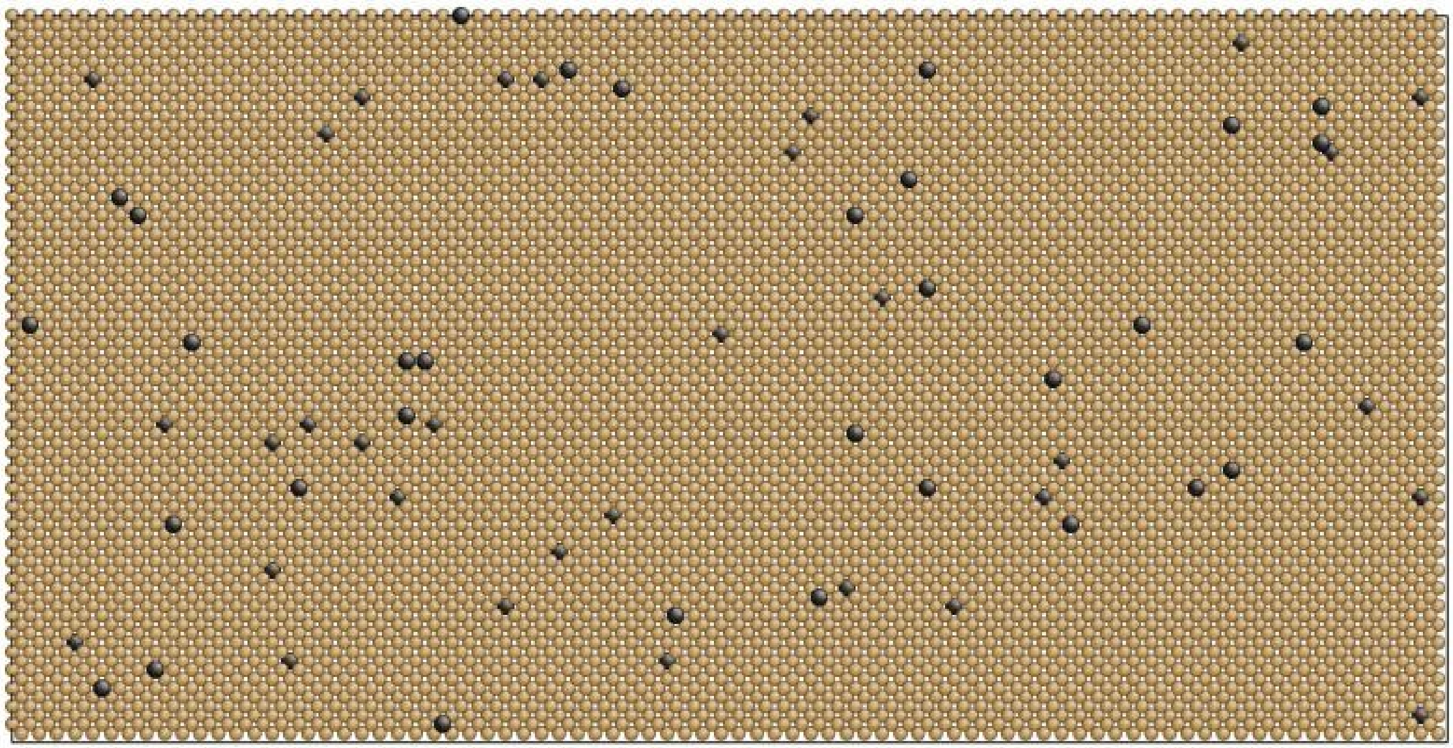
Select atoms

Option: -select

- Select a random set of atoms
  → Following options apply only to selected atoms: remove, substitute, displace, etc.

```
atomsk  initial.cfg  -select random 300 Fe  -substitute Fe C  final.cfg
```

```
atomsk  initial.cfg  -select random  1% Fe  -substitute Fe C  final.cfg
```

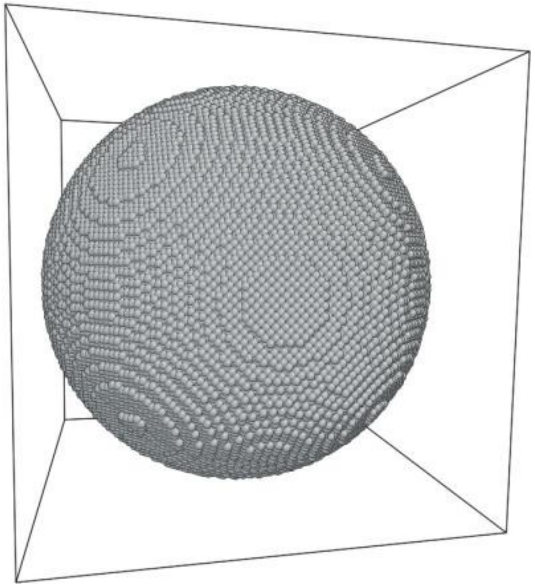


Select atoms

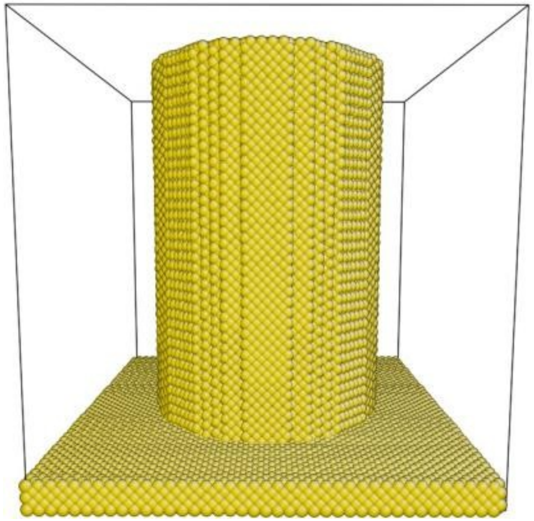
Option: -select

- Select simple geometric shapes: atoms in a box, sphere, cylinder, atoms above or below a crystal plane...
- Following options apply only to selected atoms: remove, substitute, shift, rotate, give velocity...

```
atomsk unitcell.xsf
         -duplicate  40 40 40
         -select out sphere 0.5*box 0.5*box 0.5*box 50
         -remove-atoms  select
         final.xsf
```



```
atomsk unitcell.xsf
         -duplicate  40 40 40
         -select out cylinder Z 0.5*box 0.5*box 45
         -remove-atoms  select
         final.xsf
```



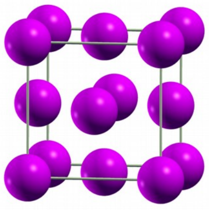
Select atoms

Option: -select

Select atoms according to a finite-element grid:
The grid is automatically mapped to fit the size of the atomic system
Atoms inside "1" elements are selected, and can be transformed (removed, substitute...)

```
atomsk unitcell.xsf  -duplicate 30 30 1  -select  grid  grid_A.txt  -substitute  Al  Au  final.cfg
```

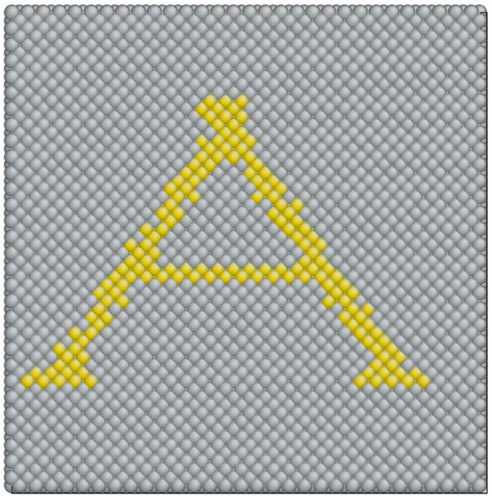
unitcell.xsf



grid_A.txt

```
000000000000000000000000000000000
000000000000000000000000000000000
000000000000000000000000000000000
000000000000000000000000000000000
000000000000011100000000000000000
000000000000001100000000000000000
000000000000011110000000000000000
000000000000110011000000000000000
000000000001100001100000000000000
000000000011000000110000000000000
000000000110000000011000000000000
000000001100000000001100000000000
000000011111111111111110000000000
000000110000000000000011000000000
000001100000000000000001100000000
000011000000000000000000110000000
000110000000000000000000011000000
011111000000000000000000111110000
000000000000000000000000000000000
000000000000000000000000000000000
000000000000000000000000000000000
000000000000000000000000000000000
000000000000000000000000000000000
```

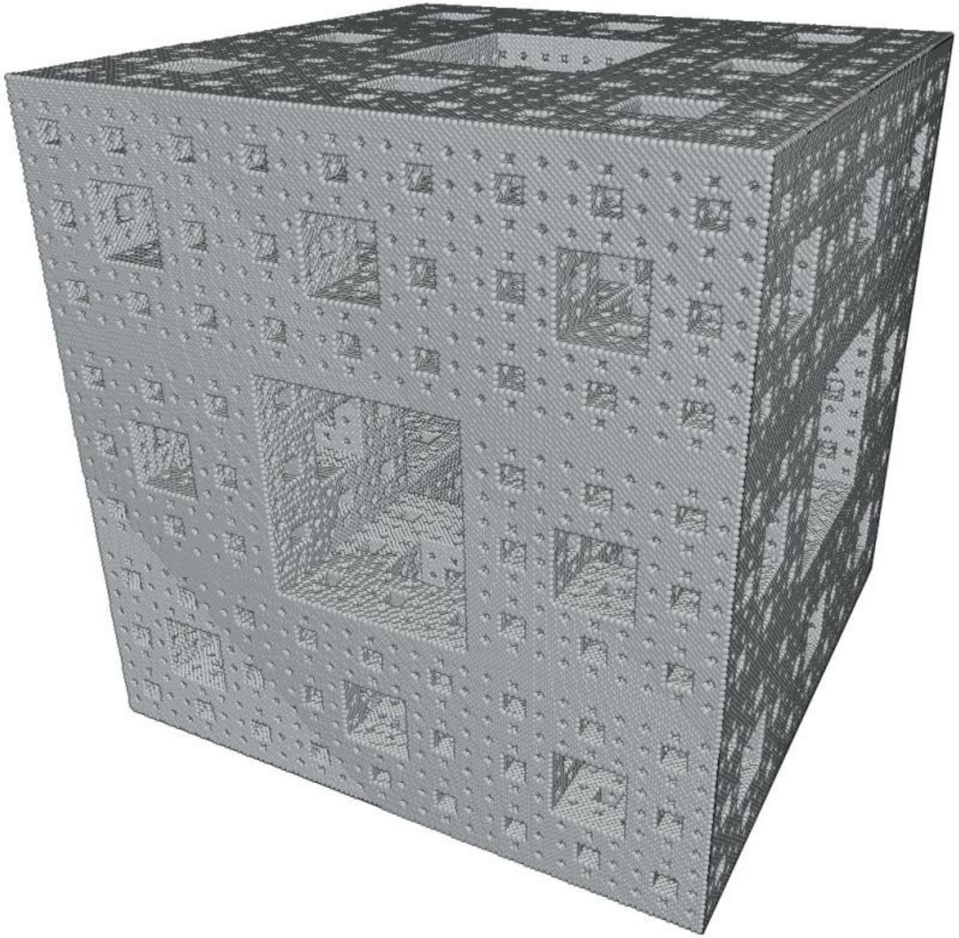
final.cfg

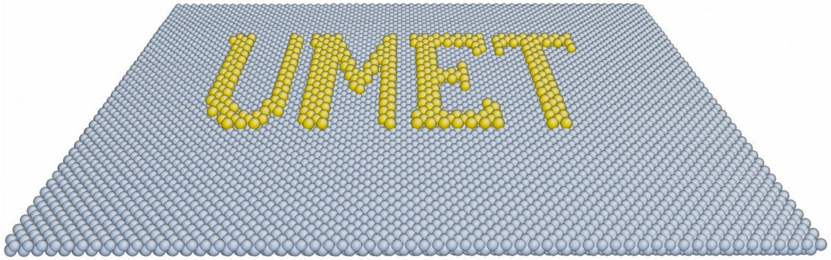


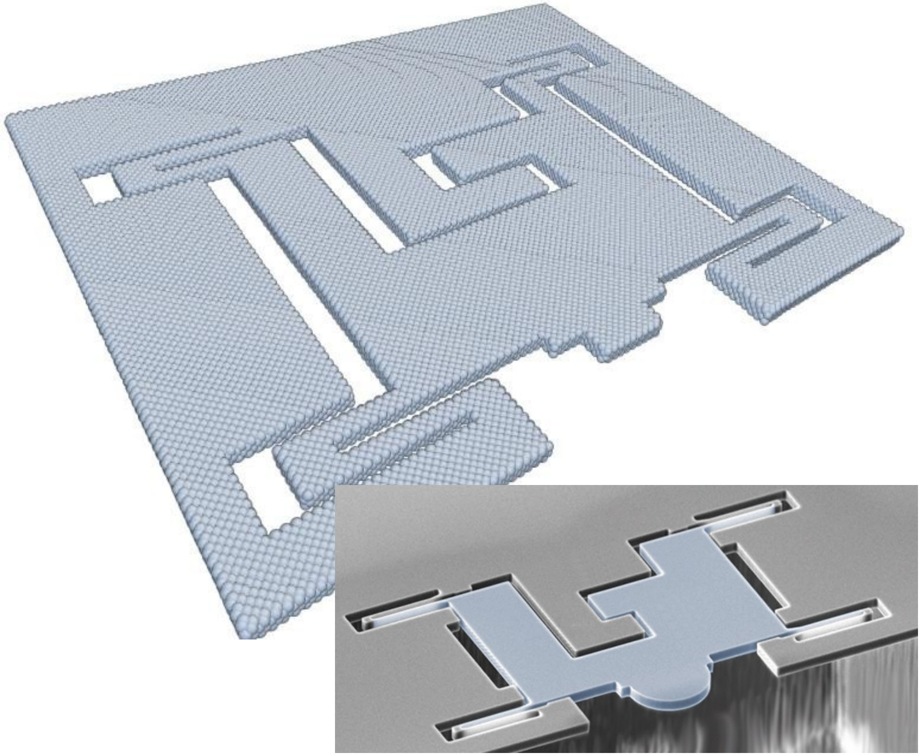
Select atoms

Option: -select

Grid selection allows to build complex shapes







Push-to-pull (Hysitron®)

Insert dislocations

Option: -dislocation

I. Isotropic elasticity

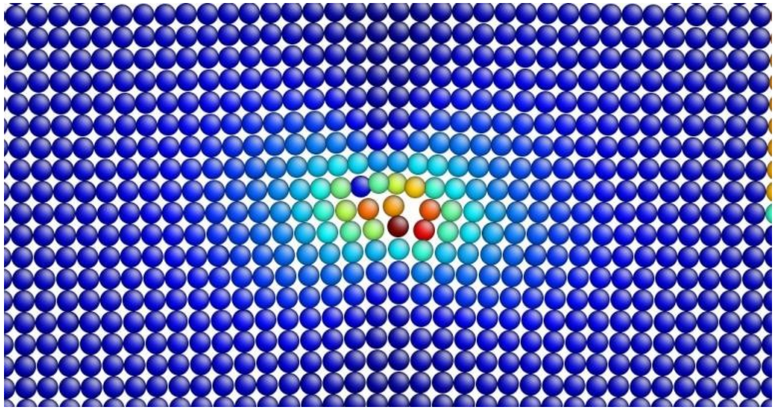
Edge

$$\begin{cases} u_x = \dfrac{b}{2\pi}\left[\arctan\left(\dfrac{y}{x}\right) + \dfrac{xy}{2(1-\nu)\left[x^2+y^2\right]}\right] \\ u_y = \dfrac{-b}{2\pi}\left[(1-2\nu)\ln\dfrac{x^2+y^2}{4(1-\nu)} + \dfrac{x^2-y^2}{4(1-\nu)(x^2+y^2)}\right] \end{cases}$$

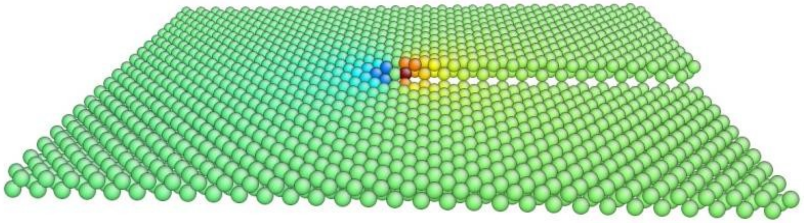
Screw

$$u_z = \frac{b}{2\pi}\arctan\left(\frac{y}{x}\right)$$

```
atomsk unitcell.xsf
            -duplicate  40 40 1
            -dislocation 30 30 edge Z Y 2.8 0.0
            final.xsf
```



```
atomsk unitcell.xsf
            -duplicate  40 40 1
            -dislocation 30 30 screw Z Y 2.8 0.0
            final.xsf
```



Insert dislocations

Option: -dislocation

II. Anisotropic elasticity

1) Write elastic tensor in a text file

tensor.txt

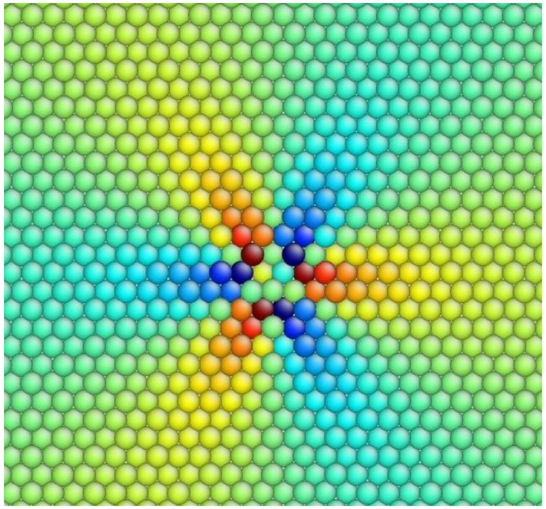
```
elastic  Voigt
243.30 243.30 243.30
145.00 145.00 145.00
116.10 116.10 116.10
```

2) Run Atomsk!

- Atomsk will solve equations of elasticity and apply displacements:

$$u_k = \Re\left\{ \frac{-1}{2\pi i} \sum_{n=1}^{3} A_k(n) D(n) \ln\left(x + py\right) \right\}, \quad k = 1, 3$$

```
atomsk unitcell.xsf
            -duplicate  40 40 1
            -properties  tensor.txt
            -dislocation 30 30 screw Z Y 2.8 0.0
            final.xsf
```

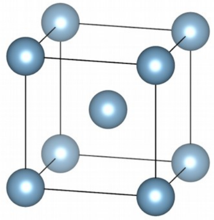


1/2[111] screw dislocation in Fe

Generate polycrystals

Mode: polycrystal

I. Polycrystal with N random grains



unitcell.xsf

1) Prepare a seed

- unit cell (or any arbitrary system)

2) Write a text file

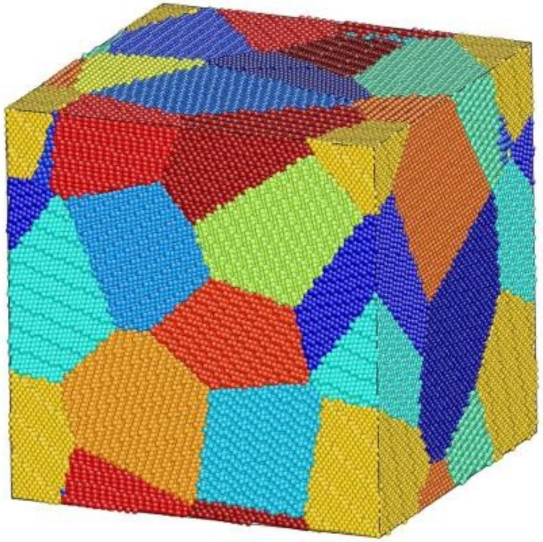
- box dimensions Lx, Ly, Lz
- keyword “random”, followed by N

file.txt

```
box  200  200  200
random  12
```

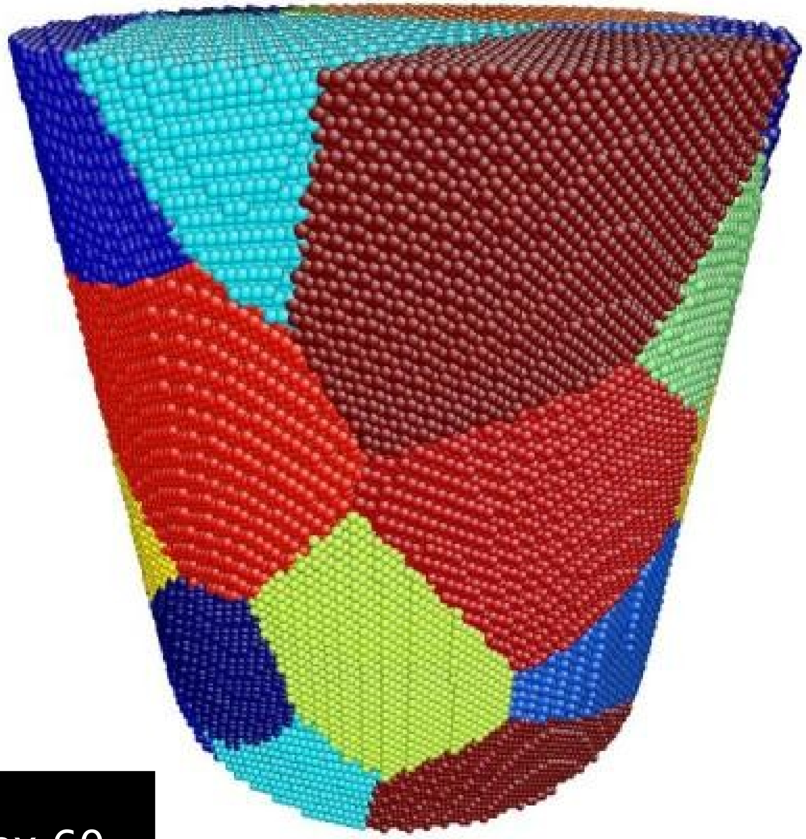
3) Run Atomsk!

```
atomsk --polycrystal unitcell.xsf  file.txt  poly.cfg
```



Generate polycrystals

Possibility to cut / shape polycrystal afterwards



```
atomsk --polycrystal unitcell.xsf  file.txt
          -select out cylinder Z 0.5*box 0.5*box 0.5*box 60
          -remove-atom select
          poly.cfg
```

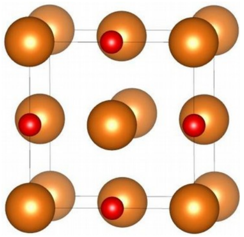
Generate polycrystals

Mode: polycrystal

II. Bicrystals and Grain Boundaries

1) Prepare a seed

- unit cell (or any arbitrary system)



unitcell.xsf

2) Write a text file

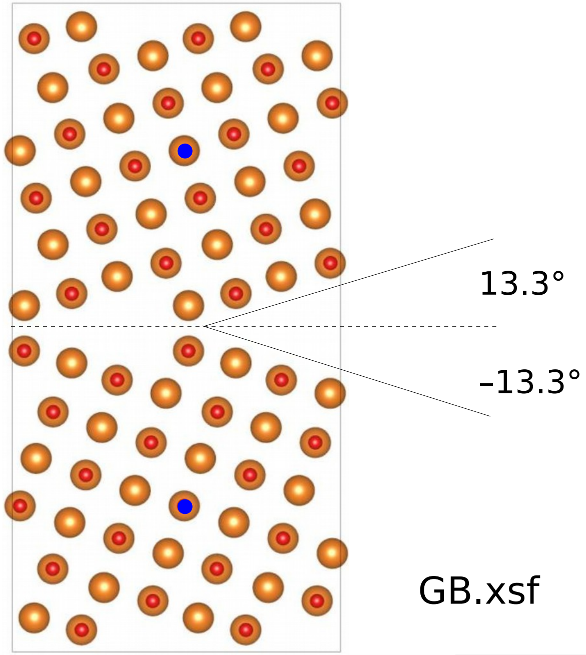
- box dimensions Lx, Ly, Lz
- two **nodes** at Y=0.25 and Y=0.75

file.txt

```
box  20  20  0
node 0.5*box 0.25*box 0   0° 0°  -13.3°
node 0.5*box 0.75*box 0   0° 0°   13.3°
```

3) Run Atomsk!

```
atomsk --polycrystal unitcell.xsf  file.txt  GB.xsf
```

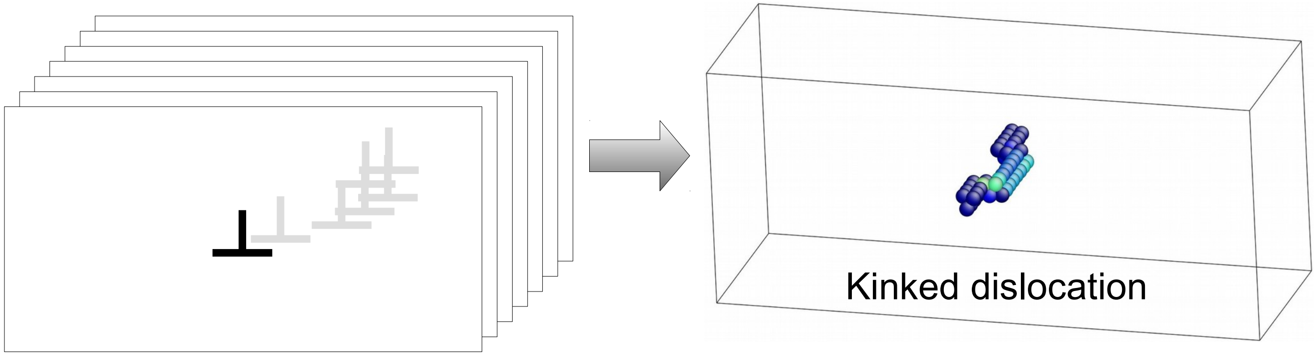


GB.xsf

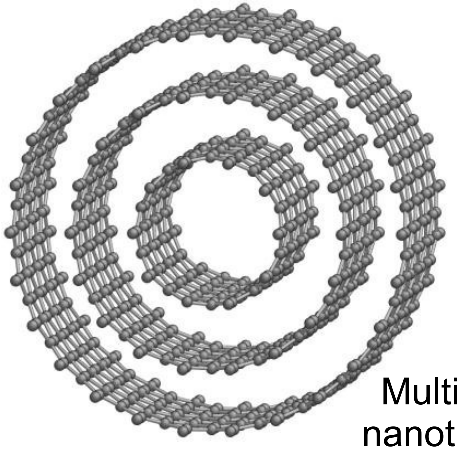
Merge systems together

Mode: merge

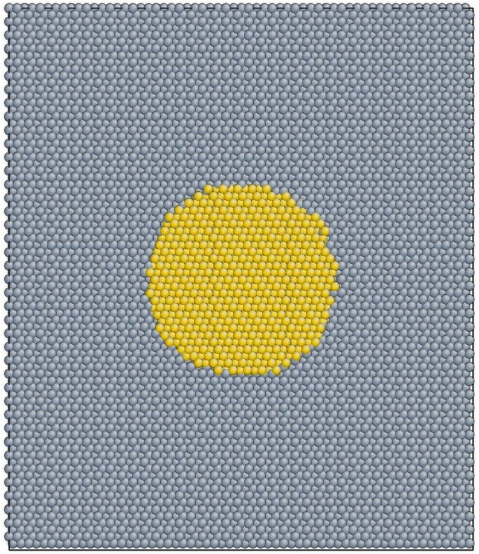
```
atomsk --merge 3  system1.xsf  system2.cfg  system3.xyz  final.xsf
```



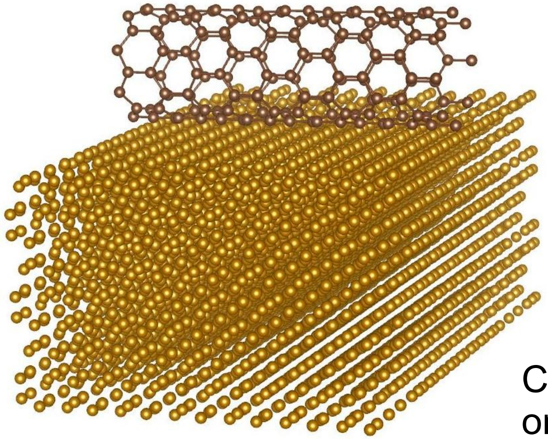
Kinked dislocation



Multi-wall nanotubes



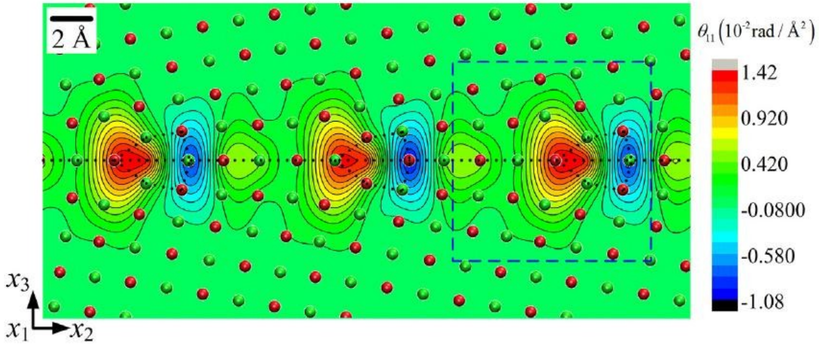
Copper inclusion in Fe matrix

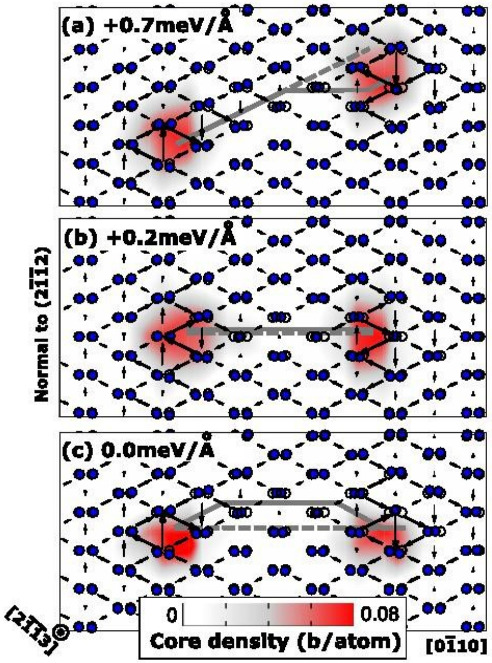


Carbon nanotube on Si substrate

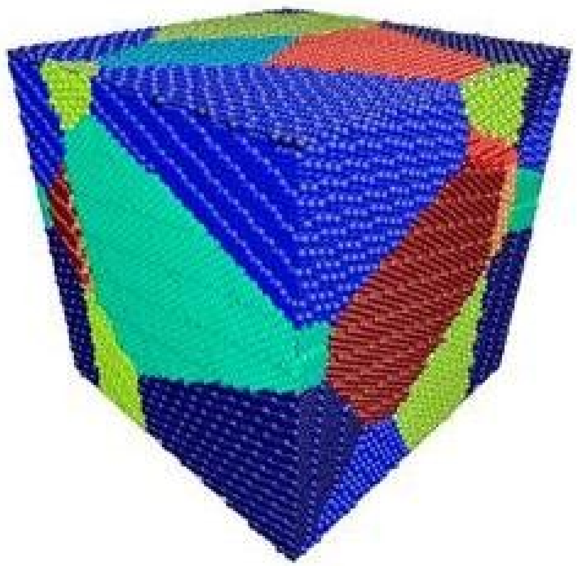
Who uses Atomsk?

Construction of grain boundaries in Cu
X.-Y. Sun et al., *Int. J. Plast.* **77** (2015) 75-89



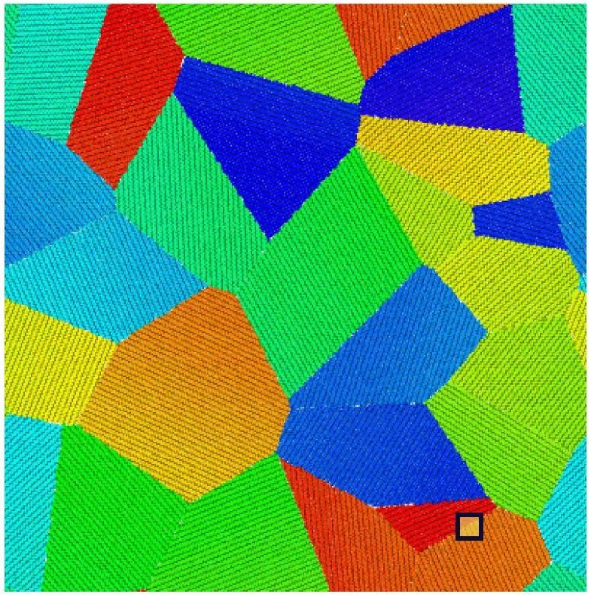


Computation of the Nye tensor
Pyramidal screw dislocations in Mg
Itakura et al.,
Phys. Rev. Lett. **116** (2016) 225501



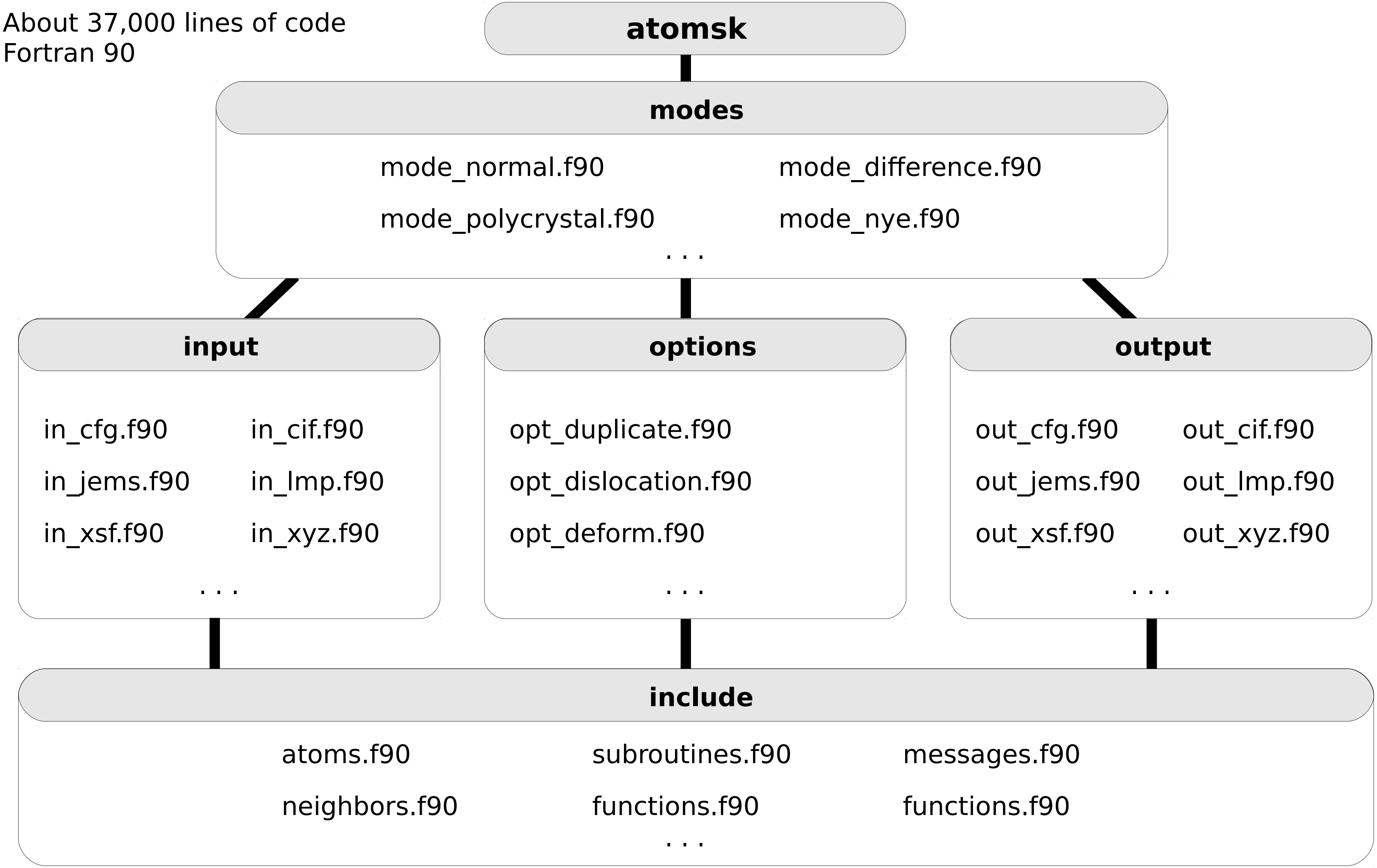
Construction of Li polycrystals
M. Chen et al.,
J. Chem. Theory Comput. **12** (2016) 2950-2963

Construction of black
phosphorous polycrystals
P. Cao et al.,
Nanotechnology **28** (2017) 045702



Source code

About 37,000 lines of code
Fortran 90



atomsk

modes

mode_normal.f90 mode_difference.f90

mode_polycrystal.f90 mode_nye.f90

. . .

input

in_cfg.f90 in_cif.f90

in_jems.f90 in_lmp.f90

in_xsf.f90 in_xyz.f90

. . .

options

opt_duplicate.f90

opt_dislocation.f90

opt_deform.f90

. . .

output

out_cfg.f90 out_cif.f90

out_jems.f90 out_lmp.f90

out_xsf.f90 out_xyz.f90

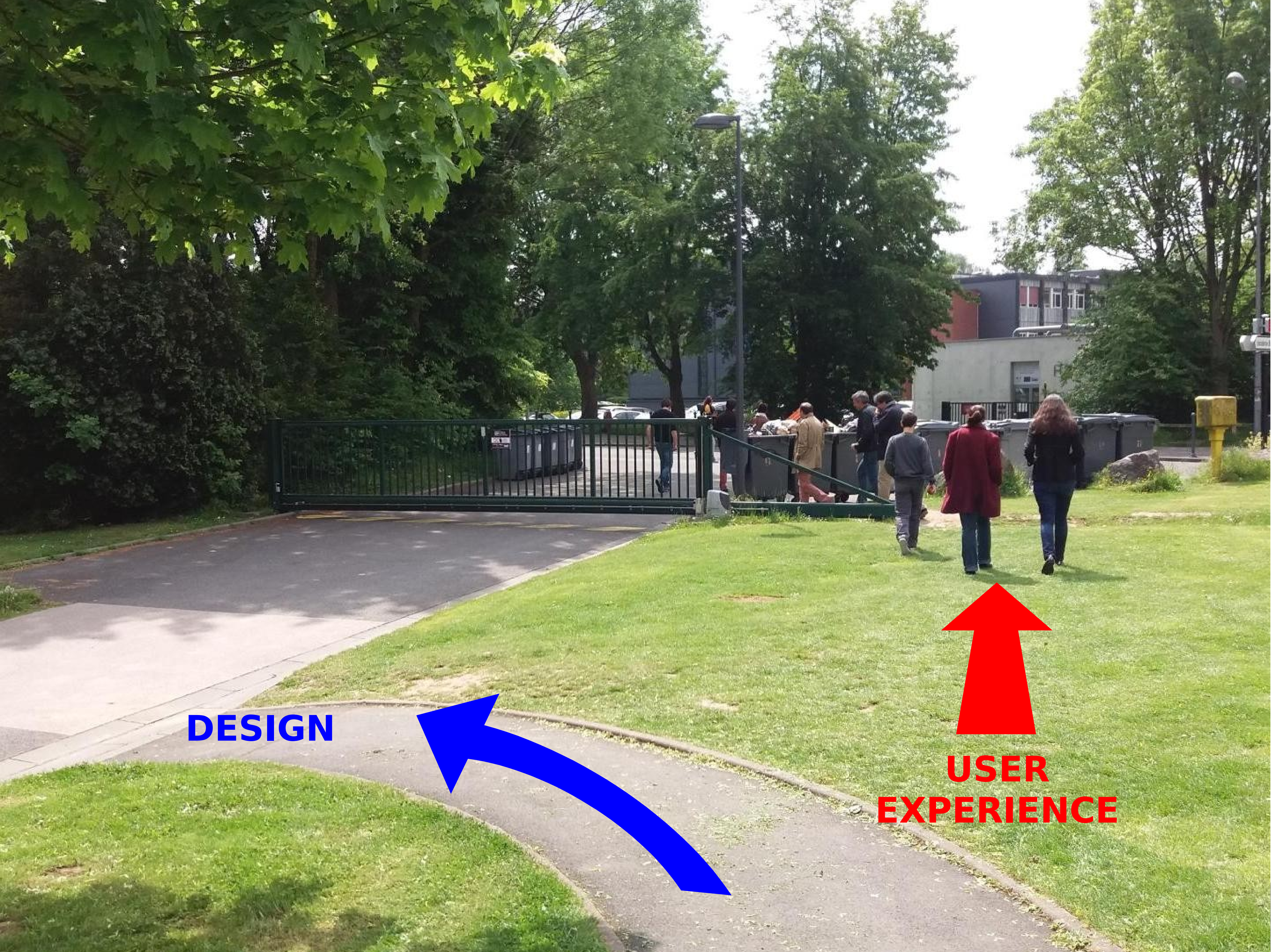
. . .

include

atoms.f90 subroutines.f90 messages.f90

neighbors.f90 functions.f90 functions.f90

. . .

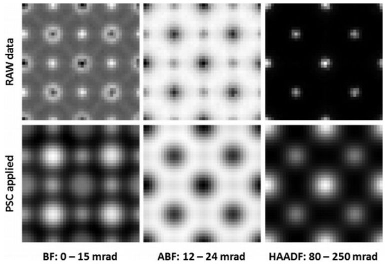


DESIGN

USER EXPERIENCE

Summary

TEM image simulation



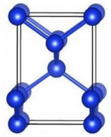
Dr Probe
JEMS
QSTEM

Atomsk

Atomsk

Atomsk

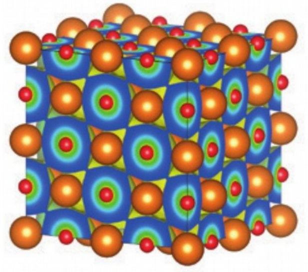
Unit cell



Atomsk

Atomsk

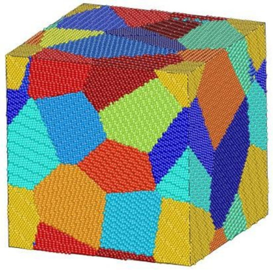
Ab initio



Quantum Espresso
SIESTA
VASP

Atomsk

Classical atomistics



DL_POLY
IMD
GULP

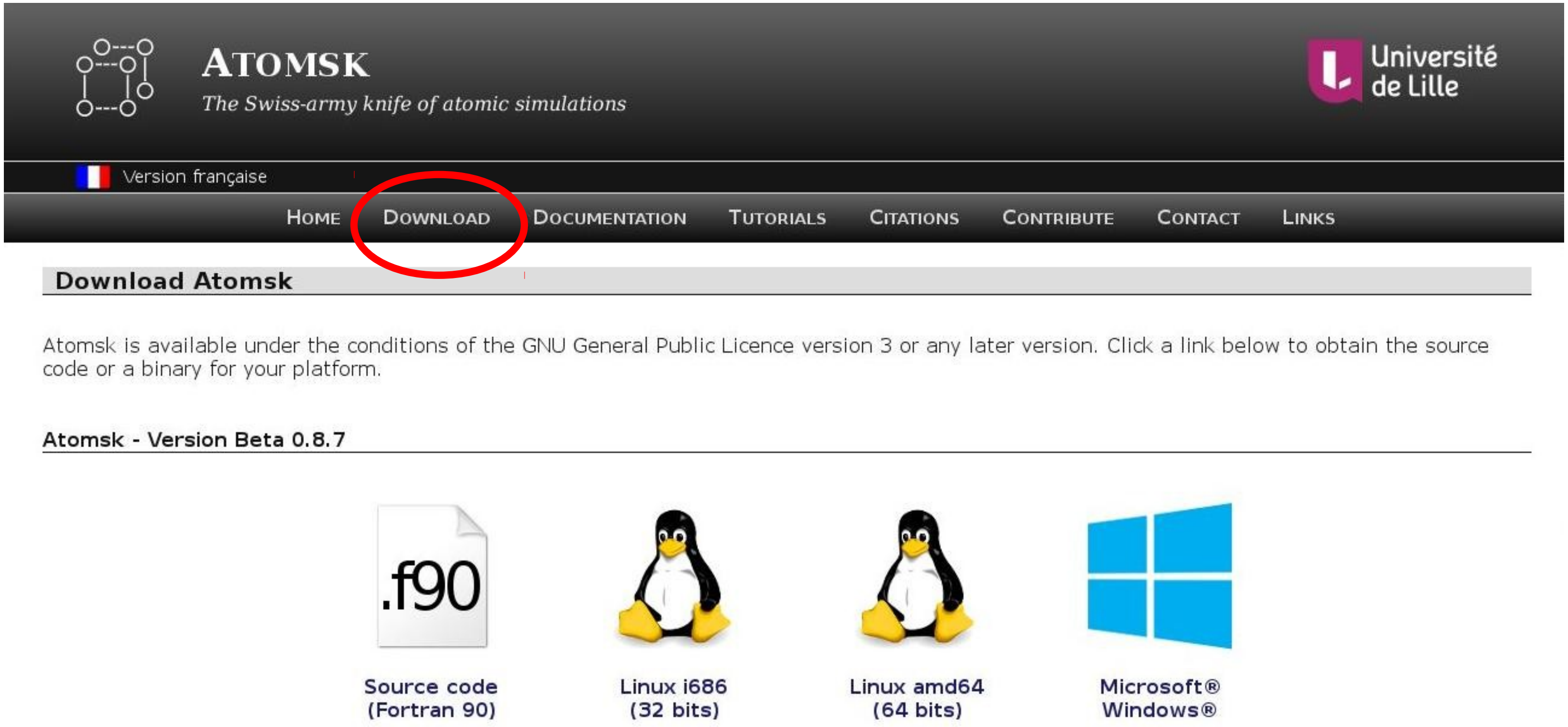
LAMMPS
MOLDY
XMD

Summary

Free, Open Source!

→ You are free to download it, use it, distribute it!
→ Contributions are welcome

Check out the Web site! http://atomsk.univ-lille1.fr



Download Atomsk

Atomsk is available under the conditions of the GNU General Public Licence version 3 or any later version. Click a link below to obtain the source code or a binary for your platform.

Atomsk - Version Beta 0.8.7

Source code (Fortran 90)

Linux i686 (32 bits)

Linux amd64 (64 bits)

Microsoft® Windows®

Cite the paper: P. Hirel, *Comput. Phys. Comm.* **197** (2015) 212-219