# How to compute phonons with siesta

Roberta Farris

Wednesday 4th October 2023

**First steps with SIESTA: from zero to hero**

# How to compute phonons with SIESTA

**We are going to use the VIBRA suit ...**

# How to compute phonons with SIESTA

» Tutorials

Edit on GitLab

## Tutorials

This set of tutorials will guide you in the exploration of Siesta's features.

**Before you do anything else, start here.** You need to set up your local working environment to follow the tutorial.

- Setting up the local working environment for the tutorial exercises

## Basics of Siesta

This section is recommended for all beginners, and also as a refresher for more experienced users.

- A first encounter with Siesta
- First crystals
- Pseudopotentials
- Basis sets
- The real-space grid
- Sampling of the BZ with k-points
- The self-consistent-field cycle
- Structural optimization using forces and stresses
- Vibration modes and phonons
- Magnetism

# How to compute phonons with SIESTA

## Vibration modes and phonons

In this set of exercises we will use the method of finite-differences implemented in Siesta to compute force constants in real space. We will explore the cases of a molecule and of a crystal. In the latter case we will focus on the need of a supercell to represent the real-space force constants.

We will also try the visualization tools available.

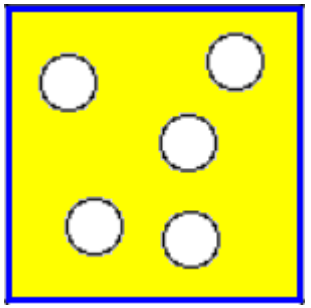
- Modes of vibration of the benzene molecule
- Phonon dispersion of bulk Si

# How to compute phonons with SIESTA

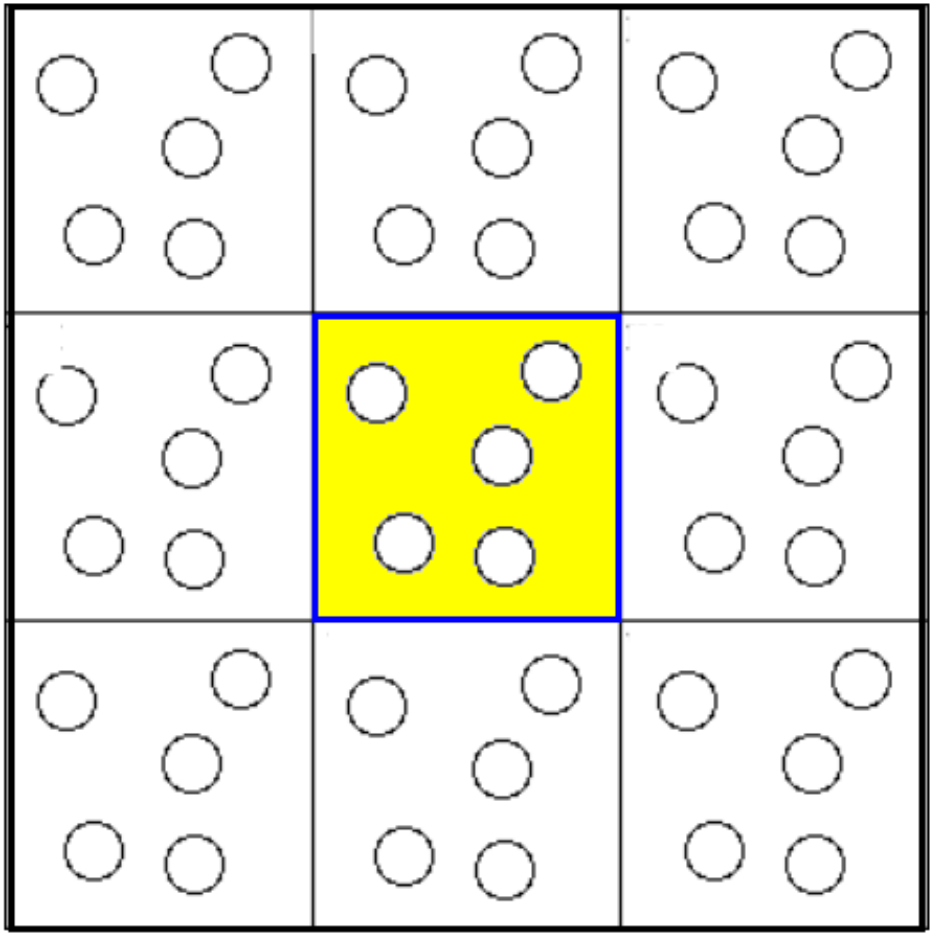
**Step 1**

**Build a supercell**

Unit cell (in real space)



Supercell(in real space)



# How to compute phonons with SIESTA

```
#
# General system descriptors
#

SystemName      Bulk Silicon in the diamond structure
#               building the supercell to compute the phonons

SystemLabel             Si
NumberOfSpecies         1
NumberOfAtoms           2
%block ChemicalSpeciesLabel
  1  14  Si
%endblock ChemicalSpeciesLabel
```

```
#
# Lattice, coordinates, k-sampling
#

LatticeConstant         5.546406 Ang    # Theor. lattice parameter of bulk Si
%block LatticeVectors
  0.00  0.50  0.50
  0.50  0.00  0.50
  0.50  0.50  0.00
%endblock LatticeVectors

AtomicCoordinatesFormat     Fractional
%block AtomicCoordinatesAndAtomicSpecies
 -0.125  -0.125  -0.125     1    28.086
  0.125   0.125   0.125     1    28.086
%endblock AtomicCoordinatesAndAtomicSpecies

kgrid_cutoff         8.0 Ang
```

```
#
# Options to generate the supercell
#

SuperCell_1    3     # number of shells in which the unit cell is
                     #   repeated in the direction of the first lattice vector.
SuperCell_2    3     # Idem for the second lattice vector.
SuperCell_3    3     # Idem for the third  lattice vector.
```

Input file to run **fcbuild** and generate the supercell

**Variables to define the unit cell in real space**

**Variables to define the supercell in real space**

# How to compute phonons with SIESTA

To generate the supercell run:

```
fcbuild < Si.fcbuild.fdf
```

# How to compute phonons with SIESTA

To generate the supercell run:

```
fcbuild < Si.fcbuild.fdf
```

This code dumps the information of the Supercell in an output file, called FC.fdf, that contains the structural data of the supercell, including:

- **The number of atoms.**
- **The lattice constant.**
- **The lattice vectors.**
- **The atomic coordinates and the atomic species of all the atoms.**

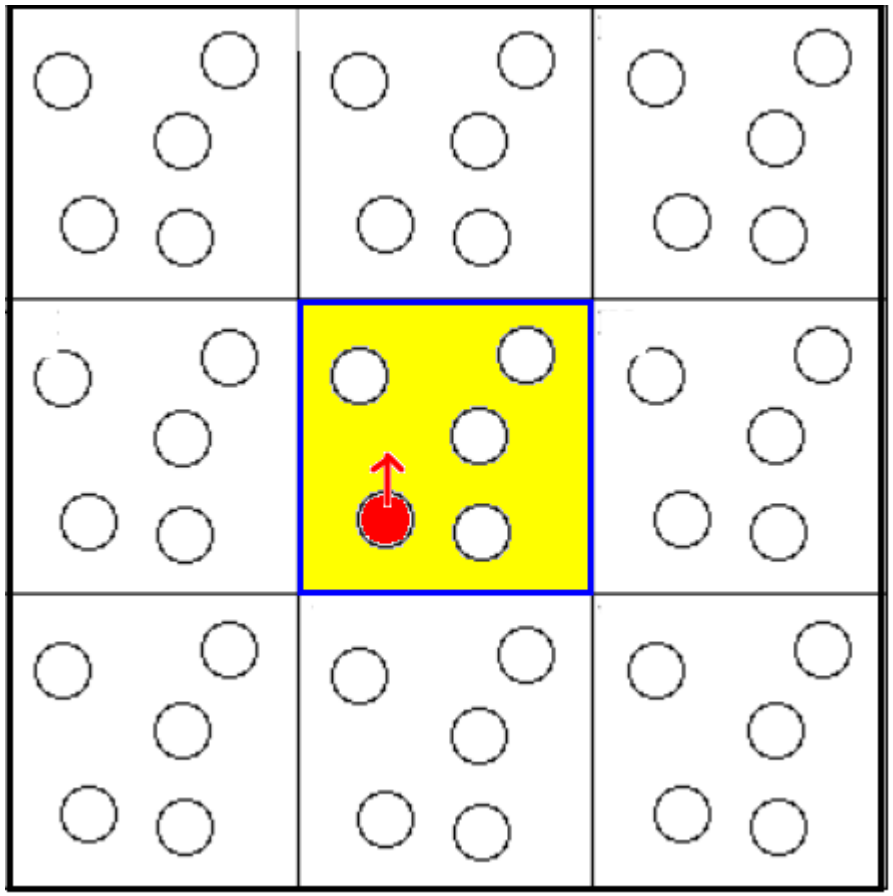
```
NumberOfAtoms         250

LatticeConstant   10.4819139708  Bohr

%block LatticeVectors
   0.0000000000     2.5000000000     2.5000000000
   2.5000000000     0.0000000000     2.5000000000
   2.5000000000     2.5000000000     0.0000000000
%endblock LatticeVectors

AtomicCoordinatesFormat  NotScaledCartesianBohr

%block AtomicCoordinatesAndAtomicSpecies
 -22.2740671880   -22.2740671880   -22.2740671880      1
 -19.6535886953   -19.6535886953   -19.6535886953      1
 -17.0331102026   -17.0331102026   -22.2740671880      1
 -14.4126317099   -14.4126317099   -19.6535886953      1
 -11.7921532172   -11.7921532172   -22.2740671880      1
  -9.1716747245    -9.1716747245   -19.6535886953      1
  -6.5511962318    -6.5511962318   -22.2740671880      1
  -3.9307177391    -3.9307177391   -19.6535886953      1
  -1.3102392464    -1.3102392464   -22.2740671880      1
   1.3102392464     1.3102392464   -19.6535886953      1
 -17.0331102026   -22.2740671880   -17.0331102026      1
 -14.4126317099   -19.6535886953   -14.4126317099      1
 -11.7921532172   -17.0331102026   -17.0331102026      1
  -9.1716747245   -14.4126317099   -14.4126317099      1
  -6.5511962318   -11.7921532172   -17.0331102026      1
  -3.9307177391    -9.1716747245   -14.4126317099      1
  -1.3102392464    -6.5511962318   -17.0331102026      1
   1.3102392464    -3.9307177391   -14.4126317099      1
```

INSPECT THE OUTPUT FILE BEFORE CONTINUE WITH THE TUTORIAL

# How to compute phonons with SIESTA

**Step 2**

**Displace the atoms in the unit cell and compute the interatomic force constants**

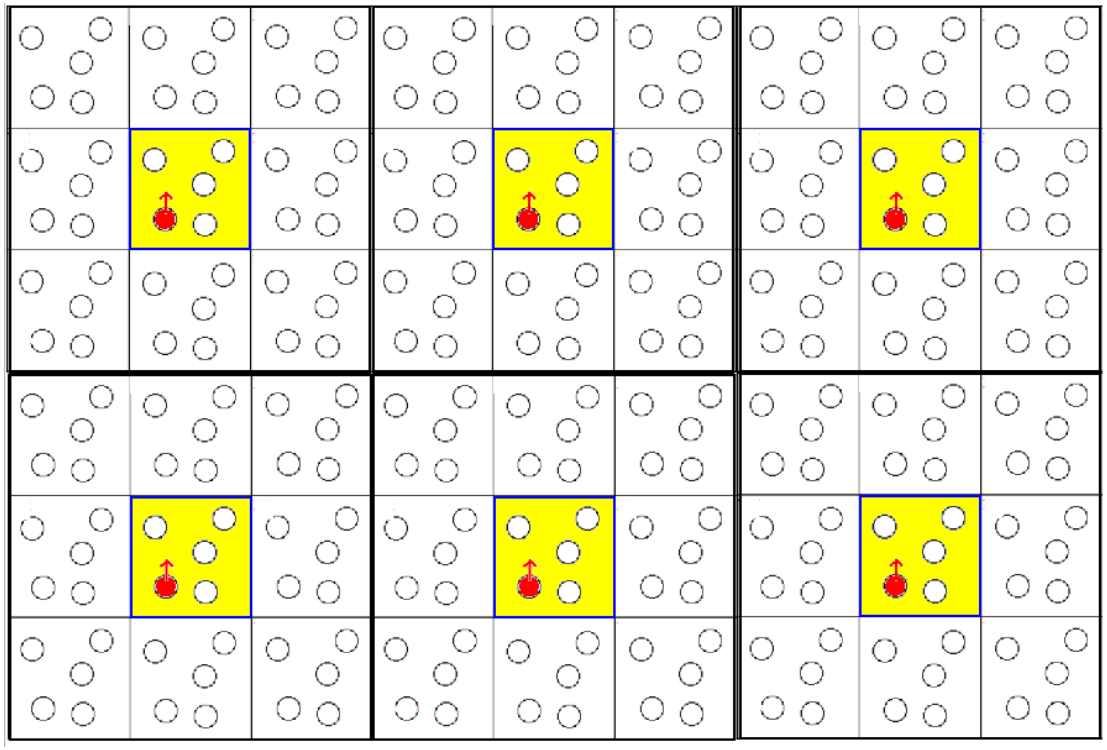


We should displace one atom at the time, but...

# How to compute phonons with SIESTA

**Step 2**

**Displace the atoms in the unit cell and compute the interatomic force constants**



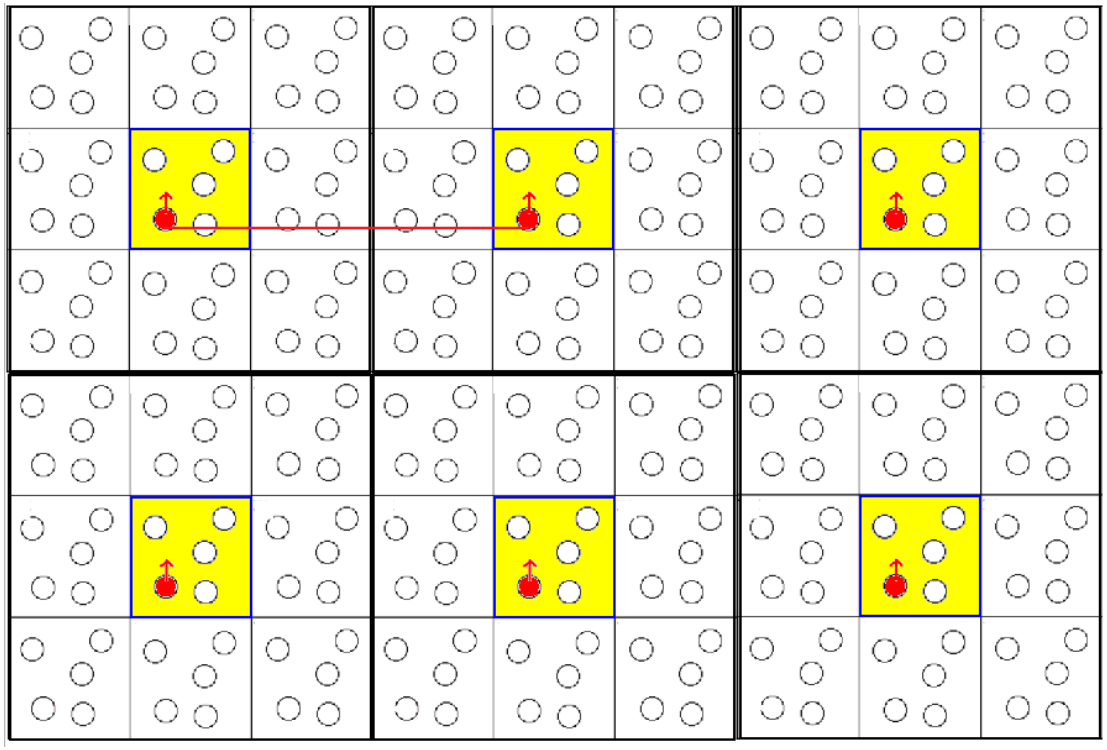
… it is not possible when using periodic boundary conditions…

IT IS IMPORTANT TO CONVERGE THE SIZE OF THE SUPERCELL

# How to compute phonons with SIESTA

**Step 2**

**Displace the atoms in the unit cell and compute the interatomic force constants**



… it is not possible when using periodic boundary conditions…

IT IS IMPORTANT TO CONVERGE THE SIZE OF THE SUPERCELL

# How to compute phonons with SIESTA

```
#
# General system descriptors
#

SystemName      Bulk Silicon in the diamond structure
#               building the supercell to compute the phonons

SystemLabel             Si
NumberOfSpecies         1
NumberOfAtoms                       < FC.fdf
%block ChemicalSpeciesLabel
  1   14   Si
%endblock ChemicalSpeciesLabel

#
# Lattice, coordinates, k-sampling
#

LatticeConstant                     < FC.fdf
LatticeVectors                      < FC.fdf

AtomicCoordinatesFormat             < FC.fdf
AtomicCoordinatesAndAtomicSpecies   < FC.fdf

kgrid_cutoff         8.0 Ang

#
# Grid
#

MeshCutoff           200 Ry

#
# Basis definition
#

PAO.BasisSize   SZ

#
# Options to compute the interatomic force constants in real space
#

MD.TypeOfRun    < FC.fdf   # Compute the interatomic force constants matrix
MD.FCfirst      < FC.fdf   # Index of first atom to displace
MD.FClast       < FC.fdf   # Index of the last atom to displace
MD.FCdispl      < FC.fdf   # Displacement to use for the computation of the
                           #   interatomic force constant matrix
                           #   (Remember that the second derivative of the
                           #   energy with respect the displacement of two
                           #   atoms is computed by means of a
                           #   finite difference derivative of the forces)
```

Input file to run **siesta** and compute the IFCs.

**Si.ifc.fdf**

**The values are taken from the FC.fdf**

# How to compute phonons with SIESTA

The output is a file called Si.FC that contains the force constant matrix

```
Force constants matrix
     -0.0312701      0.0482093      0.0482093
      0.0351977     -0.0006070     -0.0006070
     -0.0321113      0.0487602     -0.0469701
      0.0616160     -0.0701156      0.0616651
     -0.0423482      0.0009719     -0.0000591
      0.0529542     -0.0654082     -0.0593842
     -0.0321113     -0.0469701      0.0487602
      0.0616160      0.0616651     -0.0701156
      0.4098112     -0.1418340     -0.1418340
     -3.9652498      2.9318623      2.9318623
     -0.1379968      0.1480754      0.1571662
      0.0449045      0.0580579     -0.0614154
         ...            ...            ...
```

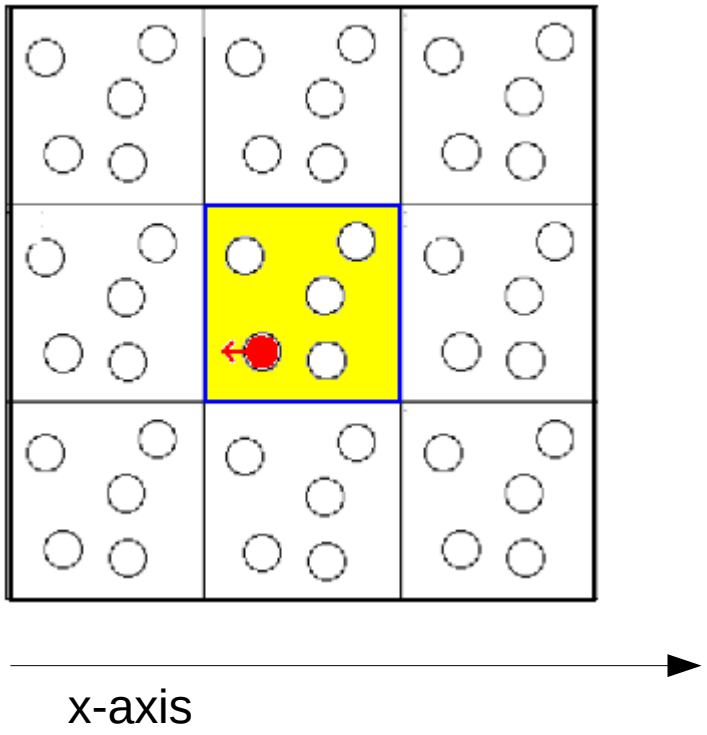
One atom per line

The forces are in eV/Å

How does it work?

# How to compute phonons with SIESTA

The output is a file called Si.FC that contains the force constant matrix




One atom per line

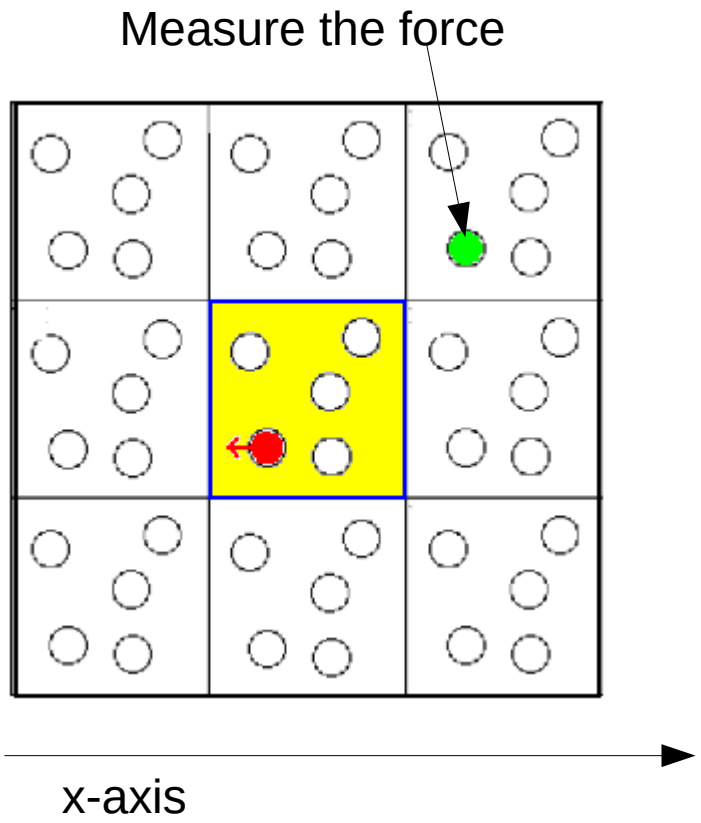
The forces are in eV/Å

How does it work?

1) atom 1 displaced along -x

# How to compute phonons with SIESTA

The output is a file called Si.FC that contains the force constant matrix




One atom per line

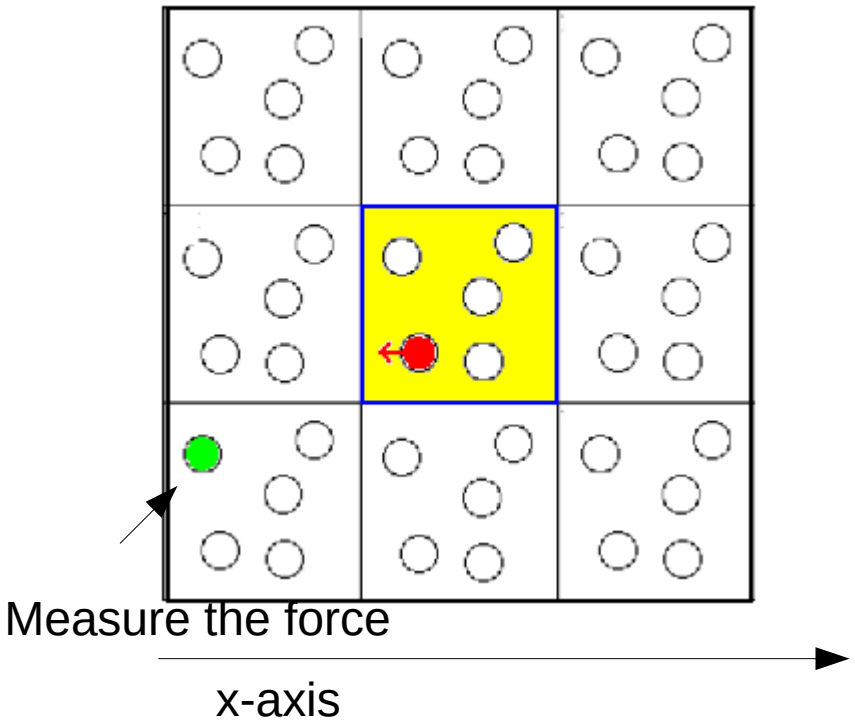
The forces are in eV/Å

How does it work?

1) atom 1 displaced along -x

# How to compute phonons with SIESTA

The output is a file called Si.FC that contains the force constant matrix




One atom per line

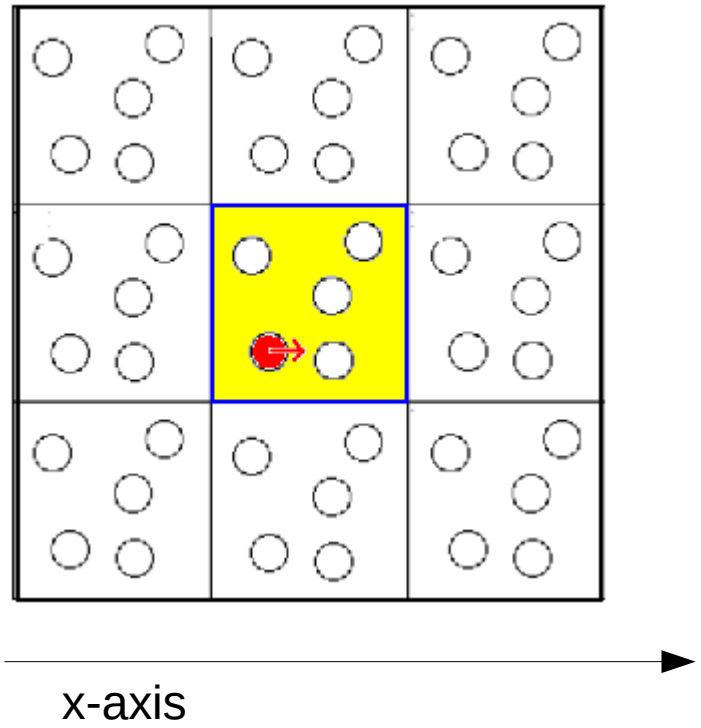
The forces are in eV/Å

How does it work?

1) atom 1 displaced along -x

# How to compute phonons with SIESTA

The output is a file called Si.FC that contains the force constant matrix




One atom per line

The forces are in eV/Å

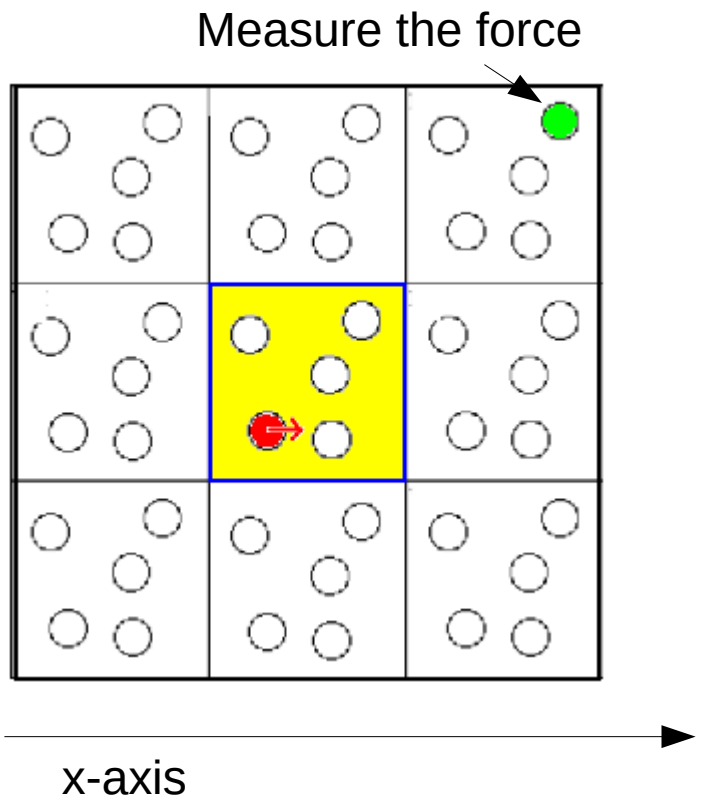
How does it work?

1) atom 1 displaced along -x

2) +x,-y,+y,-z,+z

# How to compute phonons with SIESTA

The output is a file called Si.FC that contains the force constant matrix




One atom per line

The forces are in eV/Å

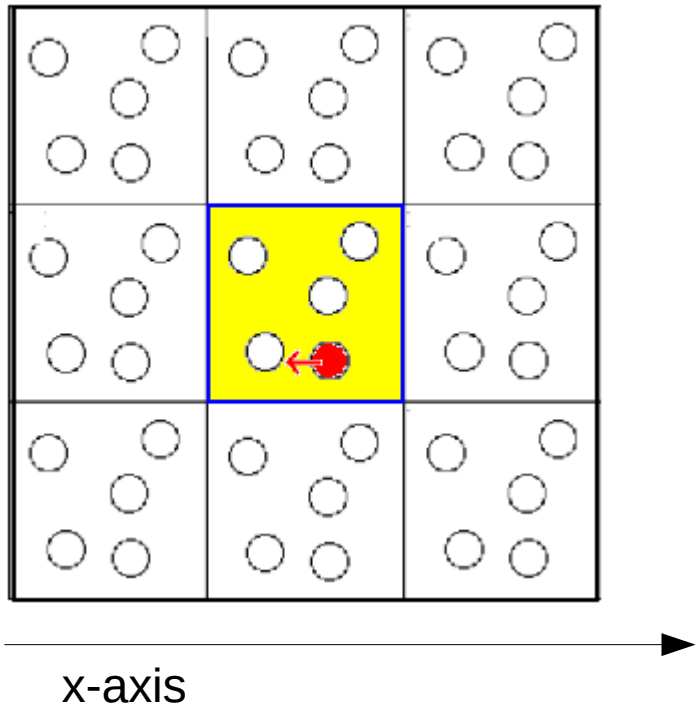
How does it work?

1) atom 1 displaced along -x

2) +x,-y,+y,-z,+z

# How to compute phonons with SIESTA

The output is a file called Si.FC that contains the force constant matrix




One atom per line

The forces are in eV/Å

How does it work?

1) atom 1 displaced along -x

2) +x,-y,+y,-z,+z

3) atom 2 displaced along -x

And so on for all the atoms in the unit cell

# How to compute phonons with SIESTA

**Step 3**

**Compute the dynamical matrix and diagonalize**

$$\tilde{D}_{\kappa\alpha\kappa\beta} \approx \frac{1}{\sqrt{M_\kappa M_\kappa}} \sum_b^{b_{max}} C_{\kappa\alpha\kappa\beta}(0,b)\, e^{i\vec{q}\vec{R}_b}$$

Once the interatomic force constants in real space have been computed, a discrete Fourier transform is performed to compute the dynamical matrix in reciprocal space.

Then, the dynamical matrix is diagonalized and its eigenfrequencies and eigenvectors are computed.

This is done using the vibra code.

# How to compute phonons with SIESTA

vibra < Si.fcbuild.fdf

It generates two outputs:

- Si.bands: mode frequencies (same format as for electronic bandstructure)
- Si.vectors: eigenmodes for each k-point

# How to compute phonons with SIESTA

**Step 4**

**Plot the bands**

gnubands < Si.bands > Si.phonon-bands.111.dat

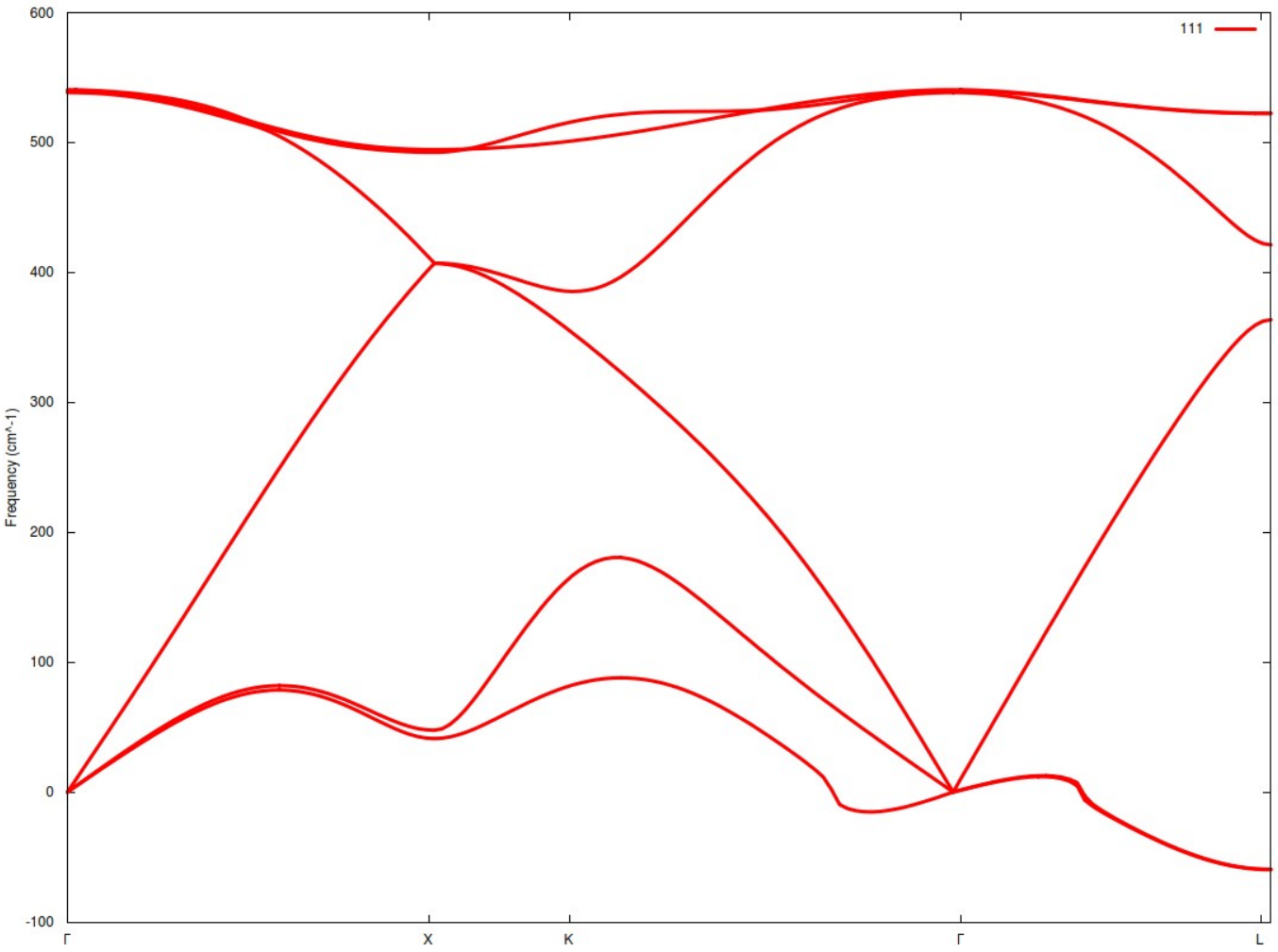
gnuplot

gnuplot> plot “Si.phonon-bands.111.dat” using 1:2 with lines

# How to compute phonons with SIESTA

**Step 4**

**Plot the bands**



# How to compute phonons with SIESTA

**Step 5**

**Test the convergence of the supercell.**

One should always check the convergence of the computed phonon band structure with respect the size of the supercell, to be sure that all the relevant interatomic force constant matrix elements are included.

the simulations for larger cells require more than hour of CPU time to generate the force constant matrix. You can either repeat the procedure explained or directly take the force constant matrix prepared for you, direct output of the proposed simulations. The name of the output files are Si.222.FC and Si.333.FC respectively (**FILES** subfolder)

# How to compute phonons with SIESTA

## Step 5

## Test the convergence of the supercell.

- To do this:
  - First, we save all the input and output files used upto now in order to be overwritten:

```
$ cp Si.fcbuild.fdf Si.fcbuild.111.fdf
$ mv FC.fdf FC.111.fdf
$ mv Si.FC Si.111.FC
$ mv Si.vectors Si.111.vectors
$ mv Si.bands Si.111.bands
```

  - Edit the file Si.fcbuild.fdf and increase the size of the supercell, adding up to 5 periodic repetitions of the unit cell in each direction (named -2, -1, 0, 1, 2)

```
#
# Options to generate the supercell
#

SuperCell_1    2     # number of shells in which the unit cell is
#   repeated in the direction of the first lattice vector.
SuperCell_2    2     # Idem for the second lattice vector.
SuperCell_3    2     # Idem for the third  lattice vector.
```

  - Repeat the previous procedure for SuperCell_1,2,3 = 2:

```
fcbuild < Si.fcbuild.fdf
siesta < Si.ifc.fdf > Si.ifc.222.out
vibra < Si.fcbuild.fdf
gnubands < Si.bands > Si.phonon-bands.222.dat
gnuplot
gnuplot> plot "Si.phonon-bands.222.dat" using 1:2 with lines

$ cp Si.fcbuild.fdf Si.fcbuild.222.fdf
$ mv FC.fdf FC.222.fdf
$ mv Si.FC Si.222.FC
$ mv Si.vectors Si.222.vectors
$ mv Si.bands Si.222.bands
```

  - Repeat the previous procedure for SuperCell_1,2,3 = 3:

```
fcbuild < Si.fcbuild.fdf
siesta < Si.ifc.fdf > Si.ifc.333.out
vibra < Si.fcbuild.fdf
gnubands < Si.bands > Si.phonon-bands.333.dat
gnuplot
gnuplot> plot "Si.phonon-bands.333.dat" using 1:2 with lines

$ cp Si.fcbuild.fdf Si.fcbuild.333.fdf
$ mv FC.fdf FC.333.fdf
$ mv Si.FC Si.333.FC
$ mv Si.vectors Si.333.vectors
$ mv Si.bands Si.333.bands
```

- To compare the results obtained with the three superlattices:

```
$ gnuplot
gnuplot> plot "Si.phonon-bands.111.dat" using 1:2 with lines,
              "Si.phonon-bands.222.dat" using 1:2 w l,
              "Si.phonon-bands.333.dat" u 1:2 with lines
```

# How to compute phonons with SIESTA