Phonons with siesta™

Roberta Farris, 13th November 2024

ICN2 10 YEARS
Institut Català de Nanociència i Nanotecnologia
BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación
CSIC
Consejo Superior de Investigaciones Científicas
MAX DRIVING THE EXASCALE TRANSITION
Ministerio de Ciencia, Innovación y Universidades
Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
AGENCIA ESTATAL DE INVESTIGACIÓN

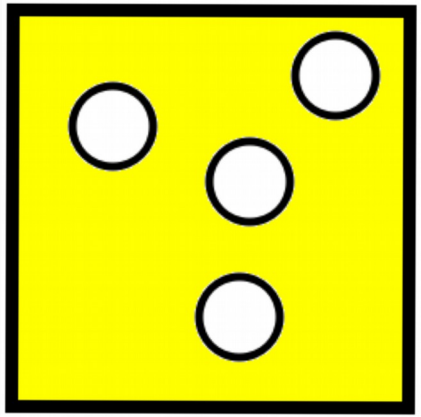
What are phonons?

What are phonons?

Definition: A phonon, in physics, is a collective excitation in an atomic periodic system.

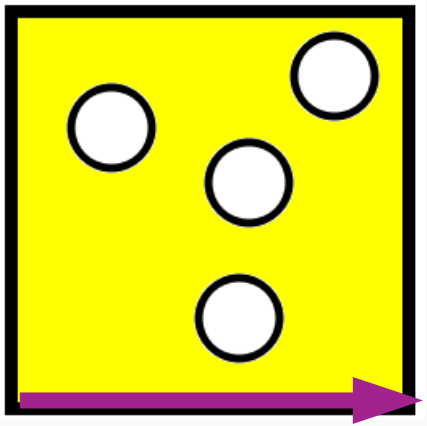
Lattice dynamics: general case

Unit cell



Lattice dynamics: general case

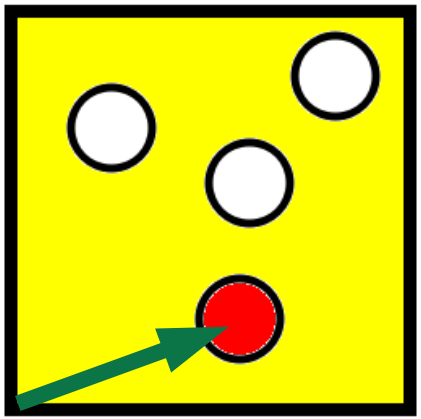
Unit cell



$\vec{R}^a$ **Vector defining the position of unit cell**

Lattice dynamics: general case

Unit cell



$\vec{R}^{a}$ **Vector defining the position of unit cell**

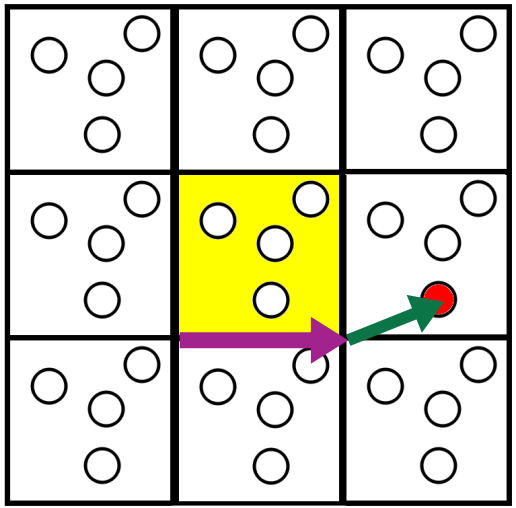
$\vec{R}_{\kappa}$ **Position of atom κ within the unit cell**

Greek characters (k) refers to atoms within the unit cell

Latin characters (a) refers to the different replicas of the unit cell

Lattice dynamics: general case

Assuming periodic boundary conditions...

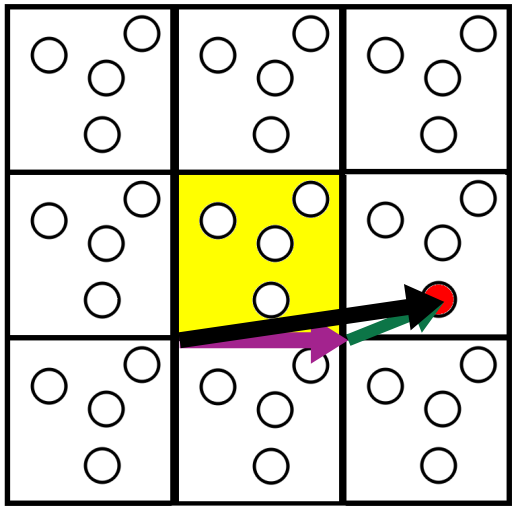


$\vec{R}^{a}$ **Vector defining the position of unit cell**

$\vec{R}_{\kappa}$ **Position of atom κ within the unit cell**

Lattice dynamics: general case

Assuming periodic boundary conditions...



Each atomic position in the supercell can be defined as:

$$\vec{R}^{a}_{\kappa} = \vec{R}_{\kappa} + \vec{R}^{a}$$

Lattice dynamics: general case

Ions can oscillate around the mean equilibrium positions.

Assuming a small deviation* of an atom from its equilibrium position $\vec{u}_{\kappa}^{a}(t)$

$$\vec{R}_{\kappa}^{a}(t) = \vec{R}_{\kappa}^{a} + \vec{u}_{\kappa}^{a}(t) = \vec{R}_{\kappa} + \vec{R}^{a} + \vec{u}_{\kappa}^{a}(t)$$

*the typical displacement of each ion around the equilibrium position is small compared to the interatomic spacing

Lattice dynamics: total energy

$$E_{i+e}^{harmonic}\left(\vec{R}_{\kappa}^{a}(t)\right)=E_{e+i}\left(\vec{R}_{\kappa}^{a}\right)+\sum_{a\,\kappa\alpha}\sum_{b\,\kappa'\beta}\frac{1}{2}\frac{\partial^{2}E_{e+i}}{\partial u_{\kappa\alpha}^{a}\partial u_{\kappa'\beta}^{b}}u_{\kappa\alpha}^{a}u_{\kappa'\beta}^{b}$$

Lattice dynamics: total energy

- **Adiabatic approximation:** electrons are in their ground state of any instantaneous ionic configuration.

$$E_{i+e}^{harmonic}\left(\vec{R}_{\kappa}^{a}(t)\right)=E_{e+i}\left(\vec{R}_{\kappa}^{a}\right)+\sum_{a\,\kappa\alpha}\sum_{b\,\kappa'\beta}\frac{1}{2}\frac{\partial^{2}E_{e+i}}{\partial u_{\kappa\alpha}^{a}\partial u_{\kappa'\beta}^{b}}u_{\kappa\alpha}^{a}u_{\kappa'\beta}^{b}$$

Lattice dynamics: total energy

- **Adiabatic approximation:** electrons are in their ground state of any instantaneous ionic configuration.
- **Harmonic approximation:** the terms beyond the quadratic order are negligible.

$$E_{i+e}^{harmonic}\left(\vec{R}_{\kappa}^{a}(t)\right)=E_{e+i}\left(\vec{R}_{\kappa}^{a}\right)+\sum_{a\,\kappa\alpha}\sum_{b\,\kappa'\beta}\frac{1}{2}\frac{\partial^{2}E_{e+i}}{\partial u_{\kappa\alpha}^{a}\partial u_{\kappa'\beta}^{b}}u_{\kappa\alpha}^{a}u_{\kappa'\beta}^{b}$$

Lattice dynamics: total energy

- **Adiabatic approximation:** electrons are in their ground state of any instantaneous ionic configuration.
- **Harmonic approximation:** the terms beyond the quadratic order are negligible.

$$E_{i+e}^{harmonic}\left(\vec{R}_{\kappa}^{a}(t)\right)=E_{e+i}\left(\vec{R}_{\kappa}^{a}\right)+\sum_{a\,\kappa\alpha}\sum_{b\,\kappa'\beta}\frac{1}{2}\frac{\partial^{2}E_{e+i}}{\partial u_{\kappa\alpha}^{a}\partial u_{\kappa'\beta}^{b}}u_{\kappa\alpha}^{a}u_{\kappa'\beta}^{b}$$

Lattice dynamics: Interatomic Force Constants

The second derivatives of the energy are defined as the **interatomic force constants** in real space:

$$C_{\kappa\alpha\kappa'\beta}(a,b) = \frac{\partial^2 E_{e+i}}{\partial u^a_{\kappa\alpha} \partial u^b_{\kappa'\beta}}$$

Lattice dynamics: equation of motion

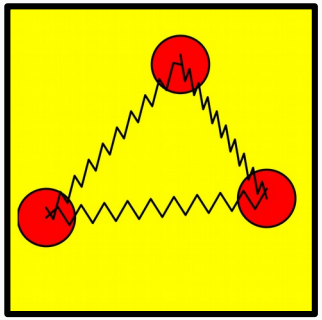
The forces exerted on atom κ in the unit cell *a* is:

$$\vec{F}_{\kappa}^{a} = M_{\kappa}\vec{a}_{\kappa}^{a}$$

Newton's Law

$$\vec{F}_{\kappa}^{a} = -C_{\kappa\alpha}u_{\kappa'\alpha}^{a}$$

Hooke's Law



Lattice dynamics: equation of motion

The forces exerted on atom κ in the unit cell *a* is:

$$\vec{F}^{a}_{\kappa} = M_{\kappa}\vec{a}^{a}_{\kappa}$$

$$\vec{F}^{a}_{\kappa} = -C_{\kappa\alpha}u^{a}_{\kappa'\alpha}$$

$$M_{\kappa}\frac{\partial^2 u^{a}_{\kappa\alpha}}{\partial t^2} = -\sum_{b\,\kappa'\beta}\frac{\partial^2 E_{e+i}}{\partial u^{a}_{\kappa\alpha}\partial u^{b}_{\kappa'\beta}}u^{b}_{\kappa'\beta}$$

For each atoms, **there are three equations of motion** of this type (one for each cartesian direction).

Lattice dynamics: equation of motion

The forces exerted on atom κ in the unit cell *a* is:

$$\vec{F}^{a}_{\kappa} = M_{\kappa}\vec{a}^{a}_{\kappa}$$

$$\vec{F}^{a}_{\kappa} = -C_{\kappa\alpha}u^{a}_{\kappa'\alpha}$$

$$M_{\kappa}\frac{\partial^2 u^{a}_{\kappa\alpha}}{\partial t^2} = -\sum_{b\,\kappa'\beta}\frac{\partial^2 E_{e+i}}{\partial u^{a}_{\kappa\alpha}\partial u^{b}_{\kappa'\beta}}u^{b}_{\kappa'\beta}$$

Lattice dynamics: equation of motion

The forces exerted on atom κ in the unit cell *a* is:

$$\vec{F}^{a}_{\kappa} = M_{\kappa}\vec{a}^{a}_{\kappa}$$

$$\vec{F}^{a}_{\kappa} = -C_{\kappa\alpha}u^{a}_{\kappa'\alpha}$$

$$M_{\kappa}\frac{\partial^2 u^{a}_{\kappa\alpha}}{\partial t^2} = -\sum_{b\,\kappa'\beta}\frac{\partial^2 E_{e+i}}{\partial u^{a}_{\kappa\alpha}\partial u^{b}_{\kappa'\beta}}u^{b}_{\kappa'\beta}$$

$$u^{a}_{\kappa\alpha}(t) = \eta_{m\vec{q}}(\kappa\alpha)e^{i\vec{q}\vec{R}_a}e^{-i\omega_m t}$$

Lattice dynamics: equation of motion

The forces exerted on atom κ in the unit cell *a* is:

$$\vec{F}^{a}_{\kappa} = M_{\kappa}\vec{a}^{a}_{\kappa}$$

$$\vec{F}^{a}_{\kappa} = -C_{\kappa\alpha}u^{a}_{\kappa'\alpha}$$

$$M_{\kappa}\frac{\partial^2 u^{a}_{\kappa\alpha}}{\partial t^2} = -\sum_{b\,\kappa'\beta}\frac{\partial^2 E_{e+i}}{\partial u^{a}_{\kappa\alpha}\partial u^{b}_{\kappa'\beta}}u^{b}_{\kappa'\beta}$$

$$u^{a}_{\kappa\alpha}(t) = \eta_{m\vec{q}}(\kappa\alpha)e^{i\vec{q}\vec{R}_a}e^{-i\omega_m t}$$

Time dependence

Lattice dynamics: equation of motion

The forces exerted on atom κ in the unit cell *a* is:

$$\vec{F}^{a}_{\kappa} = M_{\kappa}\vec{a}^{a}_{\kappa}$$

$$\vec{F}^{a}_{\kappa} = -C_{\kappa\alpha}u^{a}_{\kappa'\alpha}$$

$$M_{\kappa}\frac{\partial^2 u^{a}_{\kappa\alpha}}{\partial t^2} = -\sum_{b\,\kappa'\beta}\frac{\partial^2 E_{e+i}}{\partial u^{a}_{\kappa\alpha}\partial u^{b}_{\kappa'\beta}}u^{b}_{\kappa'\beta}$$

$$u^{a}_{\kappa\alpha}(t) = \eta_{m\vec{q}}(\kappa\alpha)e^{i\vec{q}\vec{R}_a}e^{-i\omega_m t}$$

Time dependence

Displacements written in terms of a plane wave with respect to cell coordinates

Lattice dynamics: equation of motion

The forces exerted on atom κ in the unit cell *a* is:

$$\vec{F}^{a}_{\kappa} = M_{\kappa}\vec{a}^{a}_{\kappa}$$

$$\vec{F}^{a}_{\kappa} = -C_{\kappa\alpha}u^{a}_{\kappa'\alpha}$$

$$M_{\kappa}\frac{\partial^2 u^{a}_{\kappa\alpha}}{\partial t^2} = -\sum_{b\,\kappa'\beta}\frac{\partial^2 E_{e+i}}{\partial u^{a}_{\kappa\alpha}\partial u^{b}_{\kappa'\beta}}u^{b}_{\kappa'\beta}$$

$$u^{a}_{\kappa\alpha}(t) = \eta_{m\vec{q}}(\kappa\alpha)e^{i\vec{q}\vec{R}_a}e^{-i\omega_m t}$$

$$M_{\kappa}\omega^2_{m\vec{q}}\,\eta_{m\vec{q}}(\kappa\alpha) = \sum_{k\beta}\tilde{C}_{\kappa\alpha\kappa\beta}\eta_{m\vec{q}}(\kappa\beta)$$

Lattice dynamics

Matrix form:

$$\left(\tilde{C}(\vec{q})\right)\left(\eta(\vec{q})\right) = \omega^2(\vec{q})\left(M_\kappa \delta_{\kappa\kappa'}\delta_{\alpha\beta}\right)\left(\eta(\vec{q})\right)$$

Fourier transform of the interatomic force constants

Phonon eigendisplacements

Phonon frequencies

Mass matrix

Phonon eigendisplacements

Solving the eigenvalue problem

Dynamical matrix:

$$\tilde{D}_{\kappa\alpha\kappa\beta}(\vec{q}) = \frac{1}{\sqrt{M_\kappa M_\kappa}} \tilde{C}_{\kappa\alpha\kappa\beta}(\vec{q})$$

Renormalization of the solution:

$$\underbrace{\left(\tilde{D}(\vec{q})\right)}_{\text{Dynamical matrix}} \underbrace{\left(\gamma(\vec{q})\right)}_{\substack{\text{Phonon} \\ \text{eigenvectors}}} = \underbrace{\omega^2(\vec{q})}_{\substack{\text{Phonon} \\ \text{eigenvalues}}} \underbrace{\left(\gamma(\vec{q})\right)}_{\substack{\text{Phonon} \\ \text{eigenvectors}}}$$

Solving the eigenvalue problem… in real space

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

Solving the eigenvalue problem… in real space

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

Force constant matrix in real space

Solving the eigenvalue problem… in real space

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

Force constant matrix in real space

We should:
displace the atoms one by one in the unit cell in all the 3 cartesian directions and compute the forces using finite differences...

Solving the eigenvalue problem… in real space

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

Force constant matrix in real space

We should:
displace the atoms one by one in the unit cell in all the 3 cartesian directions and compute the forces using finite differences...
… but, the force constant matrix in real space decay with the distance between atoms:
We cut the previous sum and consider only a certain number of atoms

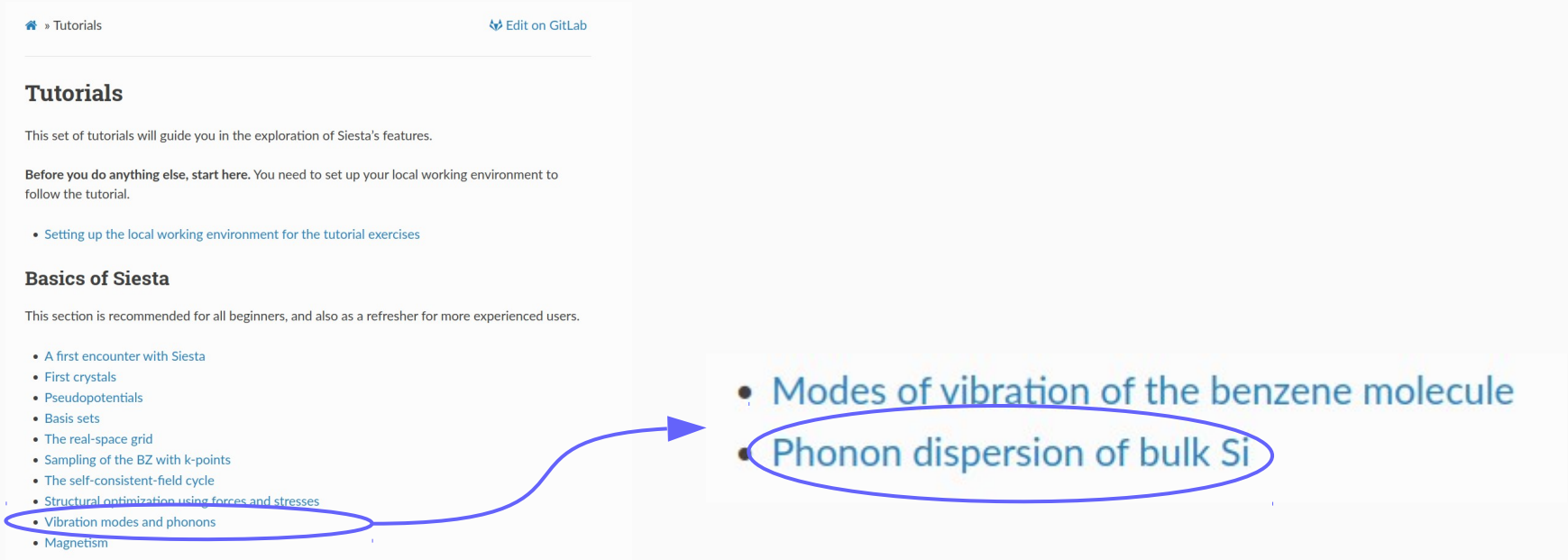
$$\frac{1}{\sqrt{M_\kappa M_\kappa}} \sum_b C_{\kappa\alpha\kappa\beta}(0,b) e^{i\vec{q}\vec{R}_b} \approx \frac{1}{\sqrt{M_\kappa M_\kappa}} \sum_b^{b_{max}} C_{\kappa\alpha\kappa\beta}(0,b) e^{i\vec{q}\vec{R}_b}$$

How do we know if the considered number of atoms is enough to represent the system?

How do we know if the considered number of atoms is enough to represent the system?

TUTORIAL!

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

- Step 1

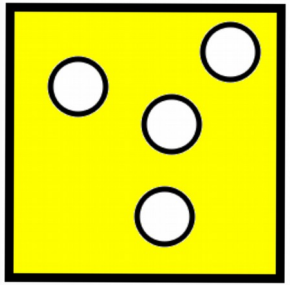
How to compute phonons with SIESTA

- Step 1: **relax the unit cell and the atomic positions** (the tutorial provides a converged structure for simplicity.. BUT in the real cases, **check carefully the convergence** as explained in the previous tutorials!

How to compute phonons with SIESTA

- Step 2: Build a supercell

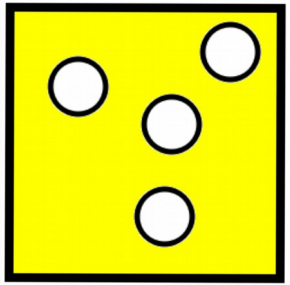
Unit cell (in real space)



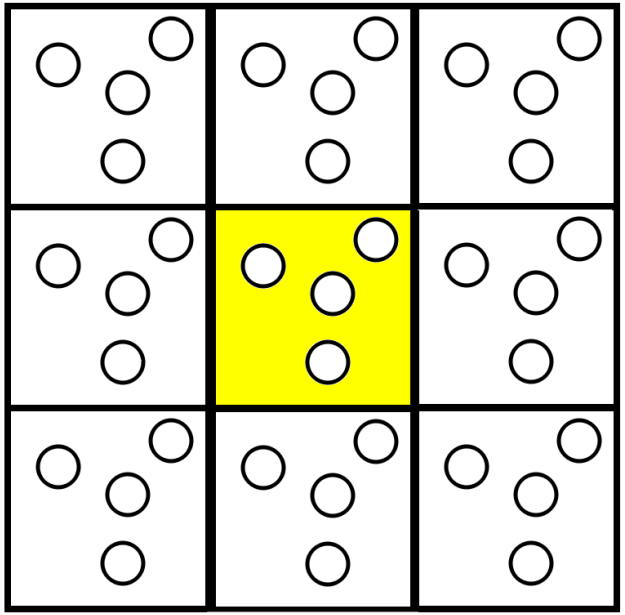
How to compute phonons with SIESTA

- Step 2: Build a supercell

Unit cell (in real space)



Supercell (in real space)



fcbuild

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

```
#
# 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 a supercell

Variables to define the unit cell in real space

Variables to define the supercell in real space

```
#
# 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.
```

How to compute phonons with SIESTA

- Step 2: build the supercell

To generate the supercell run:

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

How to compute phonons with SIESTA

- Step 2: build the supercell

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

```
NumberOfAtoms          54

LatticeConstant   10.4819139708  Bohr

%block LatticeVectors
   0.0000000000     1.5000000000     1.5000000000
   1.5000000000     0.0000000000     1.5000000000
   1.5000000000     1.5000000000     0.0000000000
%endblock LatticeVectors

AtomicCoordinatesFormat  NotScaledCartesianBohr

%block AtomicCoordinatesAndAtomicSpecies
 -11.7921532172   -11.7921532172   -11.7921532172       1
  -9.1716747245    -9.1716747245    -9.1716747245       1
  -6.5511962318    -6.5511962318   -11.7921532172       1
  -3.9307177391    -3.9307177391    -9.1716747245       1
  -1.3102392464    -1.3102392464   -11.7921532172       1
   1.3102392464     1.3102392464    -9.1716747245       1
  -6.5511962318   -11.7921532172    -6.5511962318       1
  -3.9307177391    -9.1716747245    -3.9307177391       1
  -1.3102392464    -6.5511962318    -6.5511962318       1
   1.3102392464    -3.9307177391    -3.9307177391       1
```

How to compute phonons with SIESTA

- Step 2: build the supercell

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.

```
NumberOfAtoms          54

LatticeConstant   10.4819139708  Bohr

%block LatticeVectors
   0.0000000000     1.5000000000     1.5000000000
   1.5000000000     0.0000000000     1.5000000000
   1.5000000000     1.5000000000     0.0000000000
%endblock LatticeVectors

AtomicCoordinatesFormat  NotScaledCartesianBohr

%block AtomicCoordinatesAndAtomicSpecies
 -11.7921532172   -11.7921532172   -11.7921532172       1
  -9.1716747245    -9.1716747245    -9.1716747245       1
  -6.5511962318    -6.5511962318   -11.7921532172       1
  -3.9307177391    -3.9307177391    -9.1716747245       1
  -1.3102392464    -1.3102392464   -11.7921532172       1
   1.3102392464     1.3102392464    -9.1716747245       1
  -6.5511962318   -11.7921532172    -6.5511962318       1
  -3.9307177391    -9.1716747245    -3.9307177391       1
  -1.3102392464    -6.5511962318    -6.5511962318       1
   1.3102392464    -3.9307177391    -3.9307177391       1
```

How to compute phonons with SIESTA

- Step 2: build the supercell

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.

```
NumberOfAtoms          54

LatticeConstant    10.4819139708  Bohr

%block LatticeVectors
   0.0000000000     1.5000000000     1.5000000000
   1.5000000000     0.0000000000     1.5000000000
   1.5000000000     1.5000000000     0.0000000000
%endblock LatticeVectors

AtomicCoordinatesFormat  NotScaledCartesianBohr

%block AtomicCoordinatesAndAtomicSpecies
 -11.7921532172   -11.7921532172   -11.7921532172       1
  -9.1716747245    -9.1716747245    -9.1716747245       1
  -6.5511962318    -6.5511962318   -11.7921532172       1
  -3.9307177391    -3.9307177391    -9.1716747245       1
  -1.3102392464    -1.3102392464   -11.7921532172       1
   1.3102392464     1.3102392464    -9.1716747245       1
  -6.5511962318   -11.7921532172    -6.5511962318       1
  -3.9307177391    -9.1716747245    -3.9307177391       1
  -1.3102392464    -6.5511962318    -6.5511962318       1
   1.3102392464    -3.9307177391    -3.9307177391       1
```

How to compute phonons with SIESTA

- Step 2: build the supercell

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.

```
NumberOfAtoms          54

LatticeConstant    10.4819139708  Bohr

%block LatticeVectors
   0.0000000000     1.5000000000     1.5000000000
   1.5000000000     0.0000000000     1.5000000000
   1.5000000000     1.5000000000     0.0000000000
%endblock LatticeVectors

AtomicCoordinatesFormat  NotScaledCartesianBohr

%block AtomicCoordinatesAndAtomicSpecies
 -11.7921532172   -11.7921532172   -11.7921532172       1
  -9.1716747245    -9.1716747245    -9.1716747245       1
  -6.5511962318    -6.5511962318   -11.7921532172       1
  -3.9307177391    -3.9307177391    -9.1716747245       1
  -1.3102392464    -1.3102392464   -11.7921532172       1
   1.3102392464     1.3102392464    -9.1716747245       1
  -6.5511962318   -11.7921532172    -6.5511962318       1
  -3.9307177391    -9.1716747245    -3.9307177391       1
  -1.3102392464    -6.5511962318    -6.5511962318       1
   1.3102392464    -3.9307177391    -3.9307177391       1
```

How to compute phonons with SIESTA

- Step 2: build the supercell

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          54

LatticeConstant    10.4819139708  Bohr

%block LatticeVectors
   0.0000000000     1.5000000000     1.5000000000
   1.5000000000     0.0000000000     1.5000000000
   1.5000000000     1.5000000000     0.0000000000
%endblock LatticeVectors

AtomicCoordinatesFormat  NotScaledCartesianBohr

%block AtomicCoordinatesAndAtomicSpecies
 -11.7921532172   -11.7921532172   -11.7921532172       1
  -9.1716747245    -9.1716747245    -9.1716747245       1
  -6.5511962318    -6.5511962318   -11.7921532172       1
  -3.9307177391    -3.9307177391    -9.1716747245       1
  -1.3102392464    -1.3102392464   -11.7921532172       1
   1.3102392464     1.3102392464    -9.1716747245       1
  -6.5511962318   -11.7921532172    -6.5511962318       1
  -3.9307177391    -9.1716747245    -3.9307177391       1
  -1.3102392464    -6.5511962318    -6.5511962318       1
   1.3102392464    -3.9307177391    -3.9307177391       1
```

How to compute phonons with SIESTA

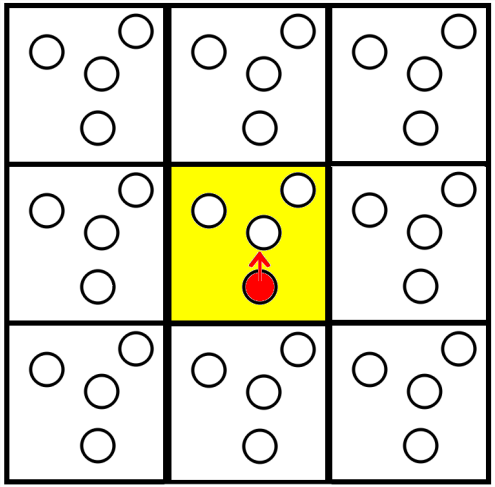
- Step 2: build the supercell

INSPECT THE OUTPUT FILE BEFORE CONTINUE WITH THE TUTORIAL!

```
NumberOfAtoms          54

LatticeConstant   10.4819139708  Bohr

%block LatticeVectors
   0.0000000000     1.5000000000     1.5000000000
   1.5000000000     0.0000000000     1.5000000000
   1.5000000000     1.5000000000     0.0000000000
%endblock LatticeVectors

AtomicCoordinatesFormat  NotScaledCartesianBohr

%block AtomicCoordinatesAndAtomicSpecies
 -11.7921532172   -11.7921532172   -11.7921532172      1
  -9.1716747245    -9.1716747245    -9.1716747245      1
  -6.5511962318    -6.5511962318   -11.7921532172      1
  -3.9307177391    -3.9307177391    -9.1716747245      1
  -1.3102392464    -1.3102392464   -11.7921532172      1
   1.3102392464     1.3102392464    -9.1716747245      1
  -6.5511962318   -11.7921532172    -6.5511962318      1
  -3.9307177391    -9.1716747245    -3.9307177391      1
  -1.3102392464    -6.5511962318    -6.5511962318      1
   1.3102392464    -3.9307177391    -3.9307177391      1
```

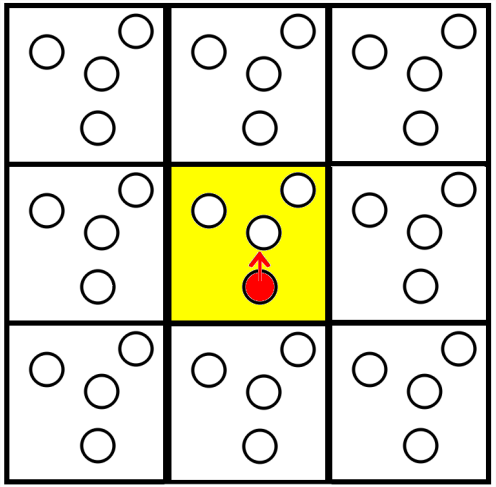
How to compute phonons with SIESTA

- Step 3: Compute the **interatomic force constants**



How to compute phonons with SIESTA

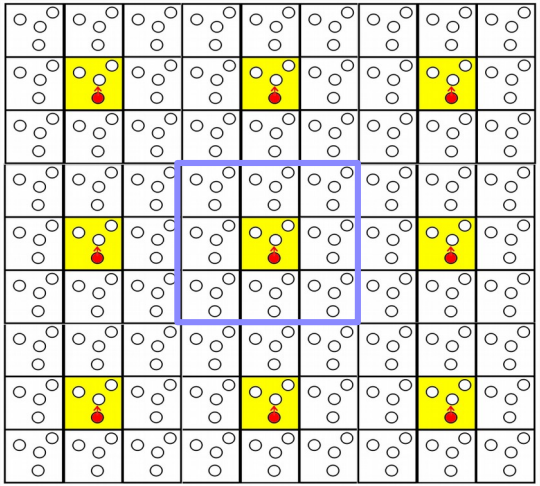
- Step 3: Compute the **interatomic force constants**



We should displace one atom at the time, but…

How to compute phonons with SIESTA

- Step 3: Compute the **interatomic force constants**

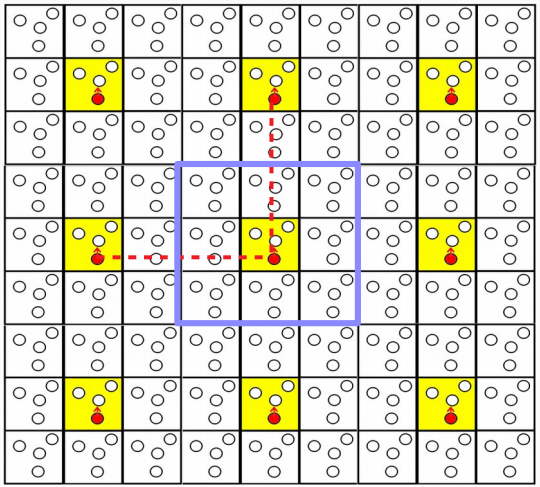


We should displace one atom at the time, but…

We are in periodic boundary conditions

How to compute phonons with SIESTA

- Step 3: Compute the **interatomic force constants**



We should displace one atom at the time, but…

We are in periodic boundary conditions

We need to converge the size of the supercell!

How to compute phonons with SIESTA

- Step 3: Compute the **interatomic force constants**

Input file to compute the IFCs:

Si.ifc.fdf

```
#
# 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
```

How to compute phonons with SIESTA

- Step 3: Compute the **interatomic force constants**

Input file to compute the IFCs:

Si.ifc.fdf

The values are taken from the FC.fdf

```
#
# 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
```

How to compute phonons with SIESTA

- Step 3: Compute the **interatomic force constants**

It generates an output, called Si.FC containing the FC matrix:

```
Force constants matrix. n_atoms, displacement [Ang]: 54 0.2116708843612000E-01
  -0.439679679E-01   0.373549952E-01   0.373549956E-01
   0.487308627E-01   0.112220260E-01   0.112220312E-01
  -0.444853181E-01   0.378480162E-01  -0.596043647E-01
   0.748537455E-01  -0.603656159E-01   0.753213122E-01
  -0.558668532E-01  -0.113062726E-01  -0.119857478E-01
   0.659435968E-01  -0.554118856E-01  -0.491154475E-01
  -0.444853060E-01  -0.596043615E-01   0.378480035E-01
   0.748537420E-01   0.753213069E-01  -0.603656168E-01
   0.398546118E+00  -0.162565673E+00  -0.162565675E+00
  -0.394375945E+01   0.295412432E+01   0.295412434E+01
  -0.155597861E+00   0.144764844E+00   0.150178506E+00
   0.566714190E-01   0.711250209E-01  -0.509770673E-01
  -0.558668527E-01  -0.119857483E-01  -0.113062728E-01
   0.659436026E-01  -0.491154460E-01  -0.554118931E-01
  -0.155597846E+00   0.150178512E+00   0.144764853E+00
   0.566714056E-01  -0.509770640E-01   0.711250128E-01
  -0.298178095E-01  -0.117791111E-01  -0.117790888E-01
```

How to compute phonons with SIESTA

- Step 3: Compute the **interatomic force constants**

It generates an output, called Si.FC containing the FC matrix:

One atom per line, the forces are in eV/Å

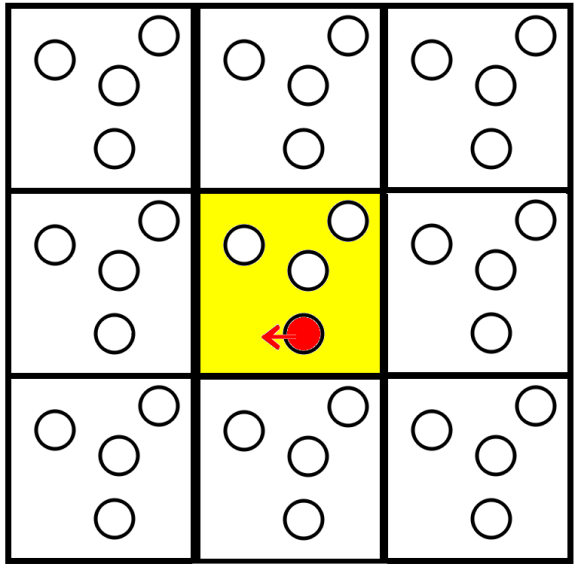
```
Force constants matrix. n_atoms, displacement [Ang]: 54 0.2116708843612000E-01
  -0.439679679E-01   0.373549952E-01   0.373549956E-01
   0.487308627E-01   0.112220260E-01   0.112220312E-01
  -0.444853181E-01   0.378480162E-01  -0.596043647E-01
   0.748537455E-01  -0.603656159E-01   0.753213122E-01
  -0.558668532E-01  -0.113062726E-01  -0.119857478E-01
   0.659435968E-01  -0.554118856E-01  -0.491154475E-01
  -0.444853060E-01  -0.596043615E-01   0.378480035E-01
   0.748537420E-01   0.753213069E-01  -0.603656168E-01
   0.398546118E+00  -0.162565673E+00  -0.162565675E+00
  -0.394375945E+01   0.295412432E+01   0.295412434E+01
  -0.155597861E+00   0.144764844E+00   0.150178506E+00
   0.566714190E-01   0.711250209E-01  -0.509770673E-01
  -0.558668527E-01  -0.119857483E-01  -0.113062728E-01
   0.659436026E-01  -0.491154460E-01  -0.554118931E-01
  -0.155597846E+00   0.150178512E+00   0.144764853E+00
   0.566714056E-01  -0.509770640E-01   0.711250128E-01
  -0.298178095E-01  -0.117791111E-01  -0.117790888E-01
```

How to compute phonons with SIESTA

- Step 3: Compute the **interatomic force constants**

How does it work?

- Atom 1 displaced along -x

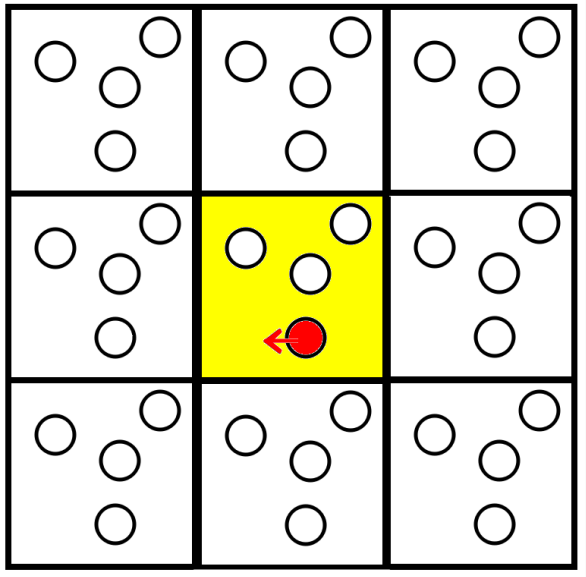


How to compute phonons with SIESTA

- Step 3: Compute the **interatomic force constants**

How does it work?

- Atom 1 displaced along -x
- Compute the forces in all the atoms in the supercell

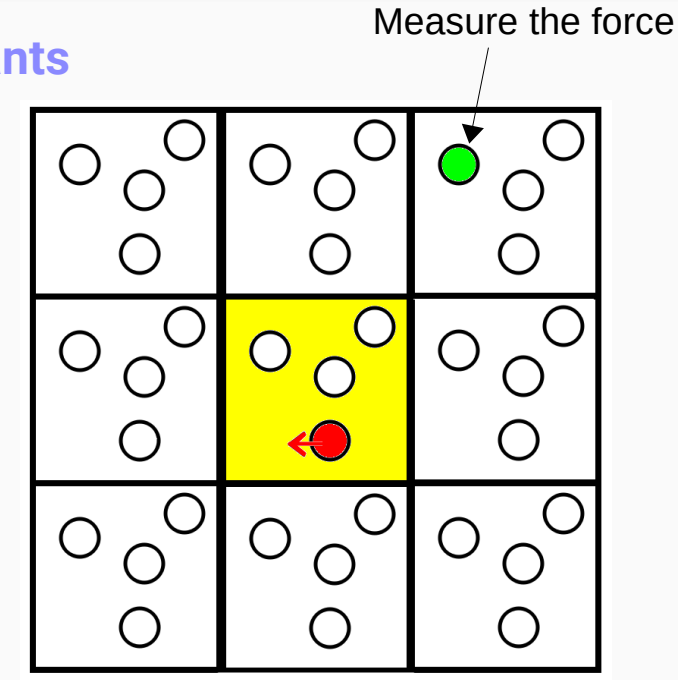


How to compute phonons with SIESTA

- Step 3: Compute the **interatomic force constants**

How does it work?

- Atom 1 displaced along -x
- Compute the forces in all the atoms in the supercell

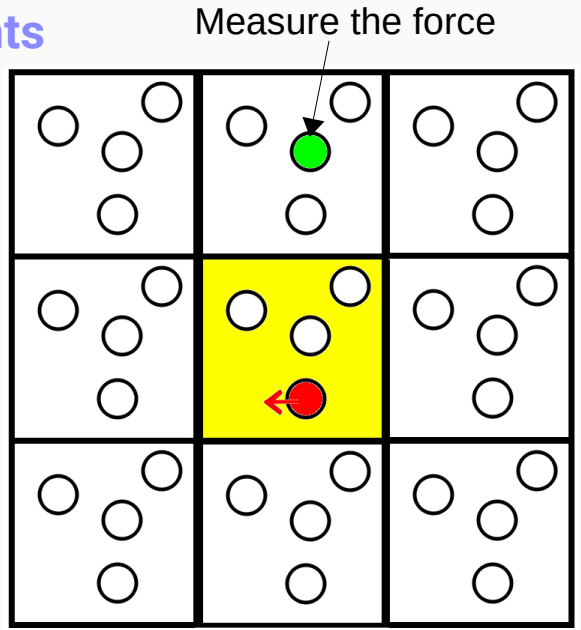


How to compute phonons with SIESTA

- Step 3: Compute the **interatomic force constants**

How does it work?

- Atom 1 displaced along -x
- Compute the forces in all the atoms in the supercell

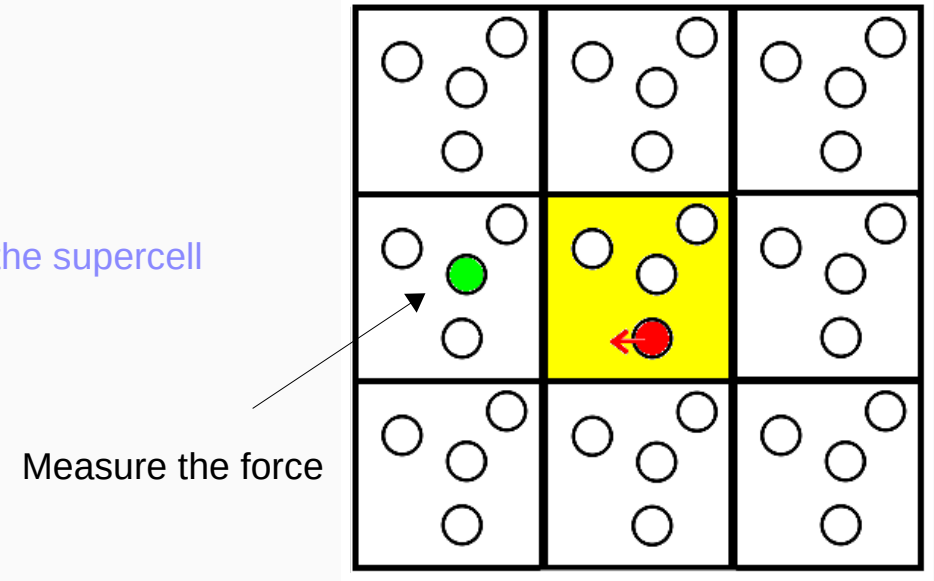


How to compute phonons with SIESTA

- Step 3: Compute the **interatomic force constants**

How does it work?

- Atom 1 displaced along -x
- Compute the forces in all the atoms in the supercell

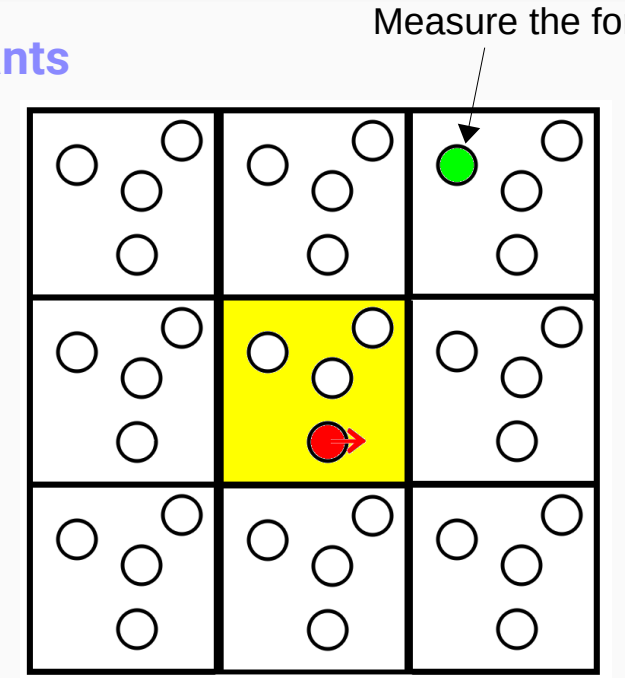


How to compute phonons with SIESTA

- Step 3: Compute the **interatomic force constants**

How does it work?

- Atom 1 displaced along -x
- Compute the forces in all the atoms in the supercell
- Atom 1 displaced along +x

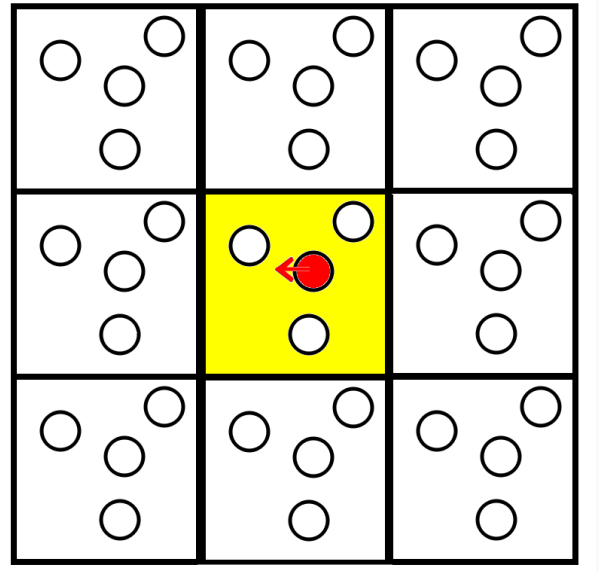


How to compute phonons with SIESTA

- Step 3: Compute the **interatomic force constants**

How does it work?

- Atom 1 displaced along -x
- Compute the forces in all the atoms in the supercell
- Atom 1 displaced along +x
- -y, +y, z, +z
- Atom 2 displaced along -x
- Etc..



How to compute phonons with SIESTA

- Step 4: Diagonalize the dynamical matrix

  We now compute the dynamical matric and diagonalize:

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

How to compute phonons with SIESTA

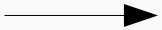
- Step 4: Diagonalize the dynamical matrix

1) The interatomic force constants in real space are computed
2) A discrete Fourier transform is performed
3) We obtain the dynamical matrix in reciprocal space.
4) The dynamical matrix is diagonalized.
5) Its eigenfrequencies and eigenvectors are computed.

How to compute phonons with SIESTA

- Step 4: Diagonalize the dynamical matrix

1) The interatomic force constants in real space are computed
2) A discrete Fourier transform is performed
3) We obtain the dynamical matrix in reciprocal space.
4) The dynamical matrix is diagonalized.
5) Its eigenfrequencies and eigenvectors are computed.



Vibra code

How to compute phonons with SIESTA

- Step 4: Diagonalize the dynamical matrix

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

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

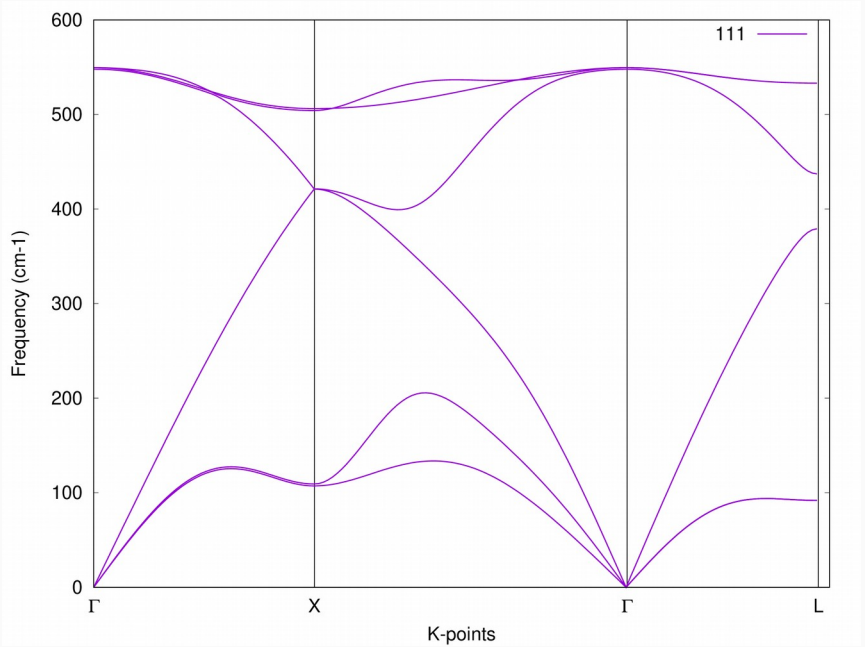
- Step 5: Plot the phonon bandstructure

```
gnubands  < Si.bands > Si.phonon-bands.111.dat
gnuplot
gnuplot> plot "Si.phonon-bands.111.dat" using 1:2 with lines
```

Do not plot the bands in MN5, download the .dat file, and use it locally in your machine

How to compute phonons with SIESTA

- Step 5: Plot the phonon bandstructure



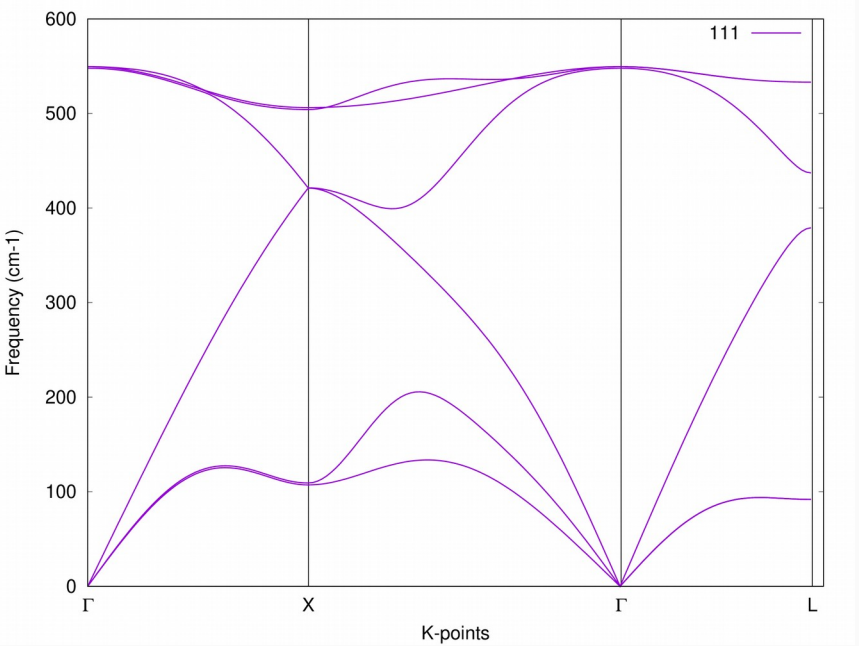
How to compute phonons with SIESTA

- Step 6: Test the convergence of the supercell

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 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 6: Test the convergence of the supercell

In order to do this:

1. We save all the input and output files we used previously, so that we prevent them from being 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
```

2. 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)

```
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.
```

How to compute phonons with SIESTA

- Step 6: Test the convergence of the supercell

We then run SIESTA as before, and store the data. For example, for the 2x2x2 case:

```
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
```

How to compute phonons with SIESTA

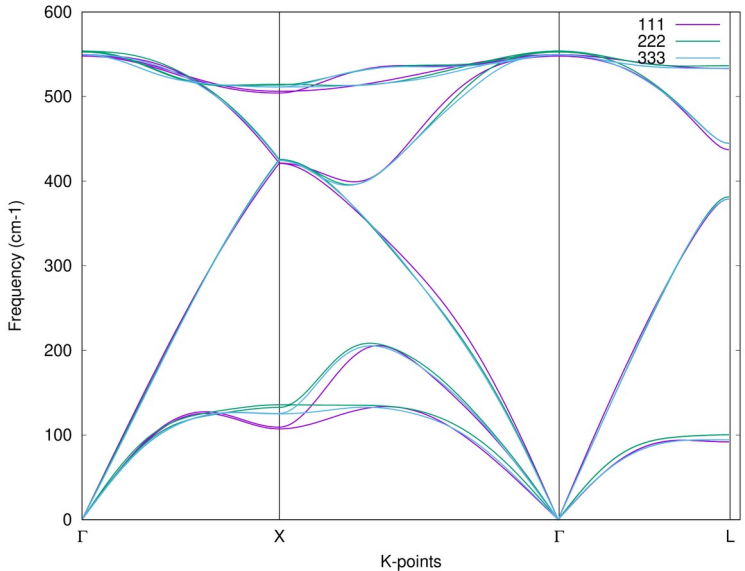
- Step 6: Test the convergence of the supercell

And finally, we can 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

- Step 6: Test the convergence of the supercell



Questions?