

An introduction to density functional theory for experimentalists

Tutorial 4.1

We create a new folder:

```
$ cd ~/scratch/summerschool ; mkdir tutorial-4.1 ; cd tutorial-4.1
```

In this tutorial we will learn how to calculate the vibrational frequencies of molecules and solids, phonon dispersion relations, LO-TO splitting, IR activity, and low-frequency dielectric constants.

Stretching frequency of a diatomic molecule

We start from the simplest possible system, the diatomic molecule Cl_2 studied in Tutorial 2.1.

We copy the setup from T2.1:

```
$ cp ../tutorial-2.1/cl2.in ./
$ cp ../tutorial-2.1/Cl.pz-bhs.UPF ./
$ cp ../tutorial-2.1/pw.x ./
```

We now modify the input file in order to make sure that we are using the optimized geometry and convergence parameters from Tutorial 2.1:

```
$ more cl2.in
```

```
&control
  calculation = 'scf'
  prefix = 'Cl2',
  pseudo_dir = './',
  outdir = './'
/
&system
  ibrav = 1,
  celldm(1) = 20.0,
  nat = 2,
  ntyp = 1,
  ecutwfc = 100,
/
&electrons
  conv_thr = 1.0d-8
/
ATOMIC_SPECIES
  Cl 1.0 Cl.pz-bhs.UPF
ATOMIC_POSITIONS bohr
  Cl 0.000 0.00 0.00
  Cl 3.725 0.00 0.00
K_POINTS gamma
```

As usual we perform a test run to make sure that everything goes smoothly. In this case it is important to set `-npool 1` in the call to `pw.x`.

In Lecture 4.1 we have seen that the vibrational frequency of a diatomic molecule can be calculated using:

$$\omega = \sqrt{\frac{2K}{M}}, \quad K = \left. \frac{\partial^2 U}{\partial d^2} \right|_{d_0}$$

where M is the mass of the Cl nucleus, U is the total potential energy surface, d is the Cl-Cl distance, and d_0 is the equilibrium bond length.

By approximating the second derivative using finite differences we have:

$$\hbar\omega \simeq \hbar \sqrt{\frac{2}{M} \frac{U(d_0 + \delta) - 2U(d_0) + U(d_0 - \delta)}{\delta^2}}$$

where δ is a small number, say $\delta = 0.001$ bohr.

We now calculate $U(d_0)$, $U(d_0 + \delta)$, and $U(d_0 - \delta)$ by creating two new input files where the coordinates of the second Cl atom are modified.

We can do this as usual using `vi`. Alternatively we can use the following direct strategy:

```
$ sed "s/3.725/3.726/g" c12.in > c12_plus.in
$ sed "s/3.725/3.724/g" c12.in > c12_minus.in
```

It is convenient to extract the corresponding total energies from the output files on the fly. This can be done as follows:

```
mpirun -np 12 pw.x -npool 1 < c12_plus.in | grep "\!" > U_plus.txt
mpirun -np 12 pw.x -npool 1 < c12_minus.in | grep "\!" > U_minus.txt
mpirun -np 12 pw.x -npool 1 < c12.in | grep "\!" > U0.txt
```

In these expressions the character `|` 'pipes' the output from the command on the left (`pw.x`) into the input of the following command (`grep "\!"`). The result is then written (`>`) into the file on the right.

After completing our batch job on the cluster we should see the following:

```
$ more U*.txt

:
:
:
U0.txt
:
:
! total energy = -59.99059545 Ry
:
:
U_minus.txt
:
:
! total energy = -59.99059541 Ry
:
:
U_plus.txt
:
:
! total energy = -59.99059506 Ry
```

At this point we can combine our results, considering that the mass of Cl is 35.45 amu (1 amu = 1822.8885 m_e). We find:

$$\hbar\omega = 70.2 \text{ meV}$$

to be compared to the experimental value of 66.7 meV.

Stretching frequency of a diatomic molecule, using DFPT

The calculation method of the previous section is very general and widely used, however there exists a faster alternative based on density-functional perturbation theory (DFPT).

In DFPT the vibrational frequency is calculated directly by working with the equilibrium structure, using perturbation theory.

In the Quantum Espresso package DFPT for vibrations is implemented in a code named `ph.x`. In order to use this code we need to go back to the root directory `summerschool/espresso-5.4.0`, and execute:

```
$ make ph
$ cp bin/ph.x ../tutorial-4.1/
```

We can build a simple input file for Cl_2 as follows:

```
$ cat > ph.in << EOF
vibrations of Cl2
&inputph
  prefix = 'Cl2',
  amass(1) = 35.45,
  outdir = './',
  fildyn = 'cl2.dyn',
/
0.0 0.0 0.0
EOF
```

Here the first line is just a comment field; the file 'cl2.dyn' will contain the dynamical matrix. `amass` is the atomic mass in amu (atomic mass units). The last line specifies that we want a calculation at the Γ point, that is $\mathbf{q} = (0, 0, 0)$. This is appropriate since we are considering an isolated molecule. Note that `prefix` must be the same as that used by `pw.x`.

In order to execute `ph.x` we first need to calculate the ground state properties of the system using `pw.x`. In this case we must modify the input file `cl2.in` as follows:

```
$ more cl2.in

...
K_POINTS tpiba
1
0.0 0.0 0.0 1.0
```

With this modification `pw.x` is still instructed to calculate wavefunctions at Γ , that is $\mathbf{k} = 0$. The difference between this file and the previous version is that now we are instructing `pw.x` to treat wavefunctions as `complex` quantities; in the previous version, the keyword `gamma` was instructing the code to treat wavefunctions as `real` quantities.

The results do not change, but this modification is needed because `ph.x` only recognizes complex wavefunctions.

We can now insert the following lines into our submission script, and run the job:

```
mpirun -np 12 pw.x -npool 1 < cl2.in > cl2.out
mpirun -np 12 ph.x -npool 1 < ph.in > ph.out
```

After completion of this job we should find the file `cl2.dyn` in our working directory:

```
$ more cl2.dyn
```

```
Dynamical matrix file
```

```
1 2 1 20.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
1 1 'Cl ' 32310.698383589064
1 1 0.0000000000 0.0000000000 0.0000000000
2 1 0.1862500000 0.0000000000 0.0000000000
```

```
Dynamical Matrix in cartesian axes
```

```
q = ( 0.000000000 0.000000000 0.000000000 )

1 1
0.39656662 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
0.00000000 0.00000000 0.00346652 0.00000000 0.00000000 0.00000000
0.00000000 0.00000000 0.00000000 0.00000000 0.00346652 0.00000000

1 2
-0.42782957 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
0.00000000 0.00000000 0.00482745 0.00000000 -0.00000000 0.00000000
0.00000000 0.00000000 -0.00000000 0.00000000 0.00482745 0.00000000

2 1
-0.42783017 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
0.00000000 0.00000000 0.00482799 0.00000000 0.00000000 0.00000000
0.00000000 0.00000000 0.00000000 0.00000000 0.00482799 0.00000000

2 2
0.39662970 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
0.00000000 0.00000000 0.00352411 0.00000000 0.00000000 0.00000000
0.00000000 0.00000000 0.00000000 0.00000000 0.00352411 0.00000000
```

```
Diagonalizing the dynamical matrix
```

```
q = ( 0.000000000 0.000000000 0.000000000 )

*****
freq ( 1) = -3.234445 [THz] = -107.889470 [cm-1]
( -0.707133 -0.000000 -0.000003 -0.000000 -0.000055 -0.000000 )
( -0.707081 -0.000000 -0.000003 -0.000000 0.000028 -0.000000 )
freq ( 2) = -0.668245 [THz] = -22.290250 [cm-1]
( 0.000015 0.000000 -0.662456 0.000000 -0.253313 0.000000 )
( 0.000015 0.000000 0.658488 0.000000 0.251752 0.000000 )
freq ( 3) = -0.667934 [THz] = -22.279866 [cm-1]
( -0.000039 0.000000 -0.253258 0.000000 0.662427 0.000000 )
( -0.000039 0.000000 0.251807 0.000000 -0.658516 0.000000 )
freq ( 4) = 1.669664 [THz] = 55.694000 [cm-1]
( -0.000007 0.000000 0.656044 0.000000 0.258052 0.000000 )
( -0.000008 0.000000 0.659998 0.000000 0.259651 0.000000 )
freq ( 5) = 1.669785 [THz] = 55.698035 [cm-1]
( 0.000009 0.000000 0.258107 0.000000 -0.656073 0.000000 )
( 0.000013 0.000000 0.259596 0.000000 -0.659969 0.000000 )
freq ( 6) = 16.617980 [THz] = 554.316153 [cm-1]
( -0.707081 0.000000 0.000000 0.000000 0.000002 0.000000 )
( 0.707133 0.000000 -0.000000 0.000000 0.000002 0.000000 )
*****
```

Here the blue lines represent the calculated dynamical matrix: we have 2 atoms and 3 Cartesian coordinates, therefore the size of this matrix is 6×6 . The blue lines correspond to precisely 36 numbers, presented as pairs of real and imaginary part.

The numbers in red are the vibrational frequencies obtained by diagonalizing the dynamical matrix.

Here we see that some frequencies are negative. This is only a convention, which is used to indicate that the diagonalization of the dynamical matrix led to a negative eigenvalue: $\omega^2 < 0$. In these cases the code prints the quantity $-\sqrt{|\omega^2|}$, and the minus sign is just a flag to warn us that something is not right. In other codes you may find the imaginary unit i in front of these frequencies, eg 107.889470 i .

In this example we were expecting to obtain $\omega = 0$ for 5 modes (3 translations of Cl_2 and 2 rotations), and one high-frequency stretching mode. However, we should keep in mind that our Cl_2 molecule is in a periodic supercell, therefore a global rotation of all the molecules must involve some small amount of energy. Furthermore, in these calculations space is not exactly 'isotropic', owing to our finite planewaves cutoff. Together these two effect lead to nonzero frequencies in modes 1–5.

These artifacts can be corrected by imposing so-called acoustic sum rules. This procedure corresponds to modifying the dynamical matrix in such a way as to make sure that the molecule will not experience any restoring force when translated or rotated. We can perform this operation by calling a **post-processing** program, `dynmat.x`:

```
$ cp ../espresso-5.4.0/bin/dynmat.x ./
$ cat > c12.dynmat.in << EOF
&input
  fildyn = 'c12.dyn',
  asr = 'zero-dim',
/
EOF
$ ./dynmat.x < c12.dynmat.in
```

Here `asr` is a flag that instructs the code to impose the acoustic sum rule (a.s.r.). Note that we are executing this small program in serial on the current node, without submitting a batch job.

This program will produce the following frequencies:

#	mode	[cm ⁻¹]	[THz]	IR
	1	0.00	0.0000	0.0000
	2	0.00	0.0000	0.0000
	3	0.00	0.0000	0.0000
	4	0.00	0.0000	0.0000
	5	0.00	0.0000	0.0000
	6	554.32	16.6180	0.0000

We see that now the system has only one nonzero vibrational frequency, as expected. The calculated value 68.7 meV ($1 \text{ meV} = 8.0655 \text{ cm}^{-1}$) is close to our result from the previous section, 65.1 meV. The two values are not identical for two reasons: (1) The acoustic sum rule modifies the potential energy surface, and (2) the present calculations correspond to taking the second derivative of U in the limit $\delta \rightarrow 0$.

The documentation about the phonon code `ph.x` can be found at the following link:
http://www.quantum-espresso.org/wp-content/uploads/Doc/INPUT_PH.html

An extensive set of examples on how to use `ph.x` is located inside the the directory:
[espresso-5.4.0/PHonon/examples/](http://www.quantum-espresso.org/PHonon/examples/)

Phonon dispersion relations of diamond

In this section we calculate the phonon dispersion relations of diamond. We begin by setting up the usual input file for diamond, from Tutorial 2.2:

```
$ wget http://www.quantum-espresso.org/wp-content/uploads/upf_files/C.pz-vbc.UPF
$ cat > scf.in << EOF
&control
  calculation = 'scf'
  prefix = 'diamond',
  pseudo_dir = './',
  outdir = './'
/
&system
 ibrav = 2,
  celldm(1) = 6.66405,
  nat = 2,
  ntyp = 1,
  ecutwfc = 100.0,
/
&electrons
  conv_thr = 1.0d-12
/
ATOMIC_SPECIES
  C 1.0 C.pz-vbc.UPF
ATOMIC_POSITIONS crystal
  C 0.00 0.00 0.00
  C 0.25 0.25 0.25
K_POINTS automatic
  4 4 4 1 1 1
EOF
```

Here the lattice constant, the Brillouin-zone sampling, and the planewaves cutoff are set to the same values that we obtained in Tutorial 2.2. We are now using a more stringent threshold for the self-consistent cycle, `conv_thr`, since phonon calculations are quite sensitive to the accuracy of the ground-state DFT calculation.

In order to calculate phonon frequencies along some high-symmetry paths in the Brillouin zone we need to go through three separate steps:

- 1) Calculate the frequencies on a uniform grid of \mathbf{q} -points;
- 2) Calculate the corresponding interatomic force constants in real space;
- 3) Calculate the frequencies along the chosen path of \mathbf{q} -points, using a Fourier interpolation.

The **first step** is performed using `ph.x`:

```
$ cat > ph.in << EOF
-
&inputph
  prefix = 'diamond',
  ldisp = .true.
  amass(1) = 12.0107,
  fildyn = 'dyn',
  nq1 = 2,
  nq2 = 2,
  nq3 = 2,
  tr2_ph = 1.0d-14,
/
EOF
```

Here the flag `ldisp = .true.` specifies that we are requesting a calculation on a uniform grid. The size of this grid is specified by the variables `nq1`, `nq2`, and `nq3`. Standard grids are of the order of $4 \times 4 \times 4$ to $8 \times 8 \times 8$ points; here we use a modest $2 \times 2 \times 2$ grid only to save time.

This calculation can be performed by using the following lines in our job submission script:

```
mpirun -np 12 pw.x -npool 4 < scf.in > scf.out
mpirun -np 12 ph.x -npool 4 < ph.in > ph.out
```

The **second step** is performed using a program called `q2r.x`. This is a small post-processing program which is found in the directory `../espresso-5.4.0/bin`. The input file is very simple, and we can execute this program locally (ie without submitting to the queue):

```
$ cp ../espresso-5.4.0/bin/q2r.x ./
$ cat > q2r.in << EOF
&input
  fildyn = 'dyn',
  flfrc = 'diam.fc'
/
EOF
$ ./q2r.x < q2r.in
```

At the end of the execution the file `diam.fc` will contain the interatomic force constants.

For the **third step** we need a program called `matdyn.x`. This is also a small post-processing program located in `../espresso-5.4.0/bin`.

```
$ cp ../espresso-5.4.0/bin/matdyn.x ./
```

The input file is on the next page, and also this program can be executed locally:

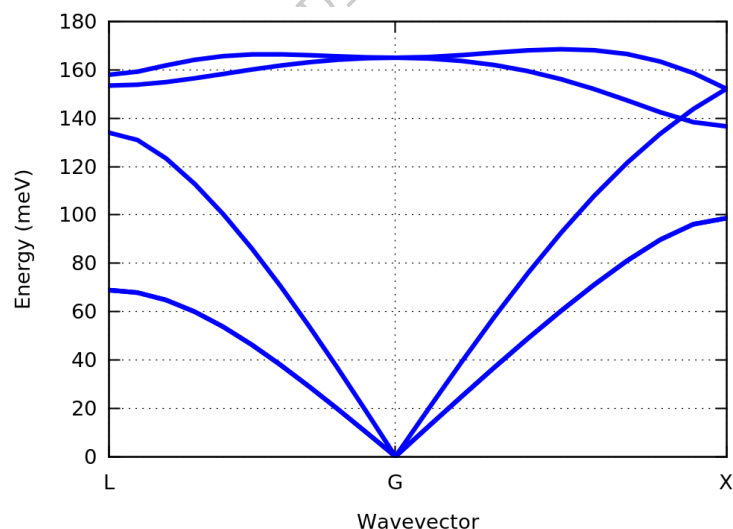
```

$ cat > matdyn.in << EOF
&input
  asr = 'simple',
  flfrc = 'diam.fc',
  flfrq = 'diam.freq'
/
21
0.500 0.500 0.500
0.450 0.450 0.450
0.400 0.400 0.400
0.350 0.350 0.350
0.300 0.300 0.300
0.250 0.250 0.250
0.200 0.200 0.200
0.150 0.150 0.150
0.100 0.100 0.100
0.050 0.050 0.050
0.000 0.000 0.000
0.100 0.000 0.000
0.200 0.000 0.000
0.300 0.000 0.000
0.400 0.000 0.000
0.500 0.000 0.000
0.600 0.000 0.000
0.700 0.000 0.000
0.800 0.000 0.000
0.900 0.000 0.000
1.000 0.000 0.000
EOF
$ ./matdyn.x < matdyn.in

```

In this file we are specifying that we want the code to calculate vibrational frequencies for 21 \mathbf{q} -points. The Cartesian coordinates of these points are specified in units of $2\pi/a$. In this example we have 21 points uniformly distributed along the path $L \rightarrow \Gamma \rightarrow X$. L is $(1/2, 1/2, 1/2)2\pi/a$, X is $(1, 0, 0)2\pi/a$, and Γ is $(0, 0, 0)$.

The calculated frequencies can be found in the file `diam.freq.gp`. A plot of these data using `gnuplot` gives the following phonon dispersion relations:



LO-TO splitting, IR activity, and dielectric constant of GaAs

In this section we consider GaAs as an example of **polar** semiconductor. The atoms of polar semiconductors exhibit nonzero Born effective charges. The main consequences of nonzero Born charges are: i) The vibrational frequencies of longitudinal and transverse optical phonons at long wavelength ($\mathbf{q} \rightarrow 0$) do not coincide. This is called LO-TO splitting. ii) The system exhibits infrared (IR) activity. iii) The ionic vibrations provide an additional contribution to the dielectric constant at low frequency.

Let us create a basic input file for `pw.x`, for the case of GaAs:

```
$ wget http://www.quantum-espresso.org/wp-content/uploads/upf_files/Ga.pz-bhs.UPF
$ wget http://www.quantum-espresso.org/wp-content/uploads/upf_files/As.pz-bhs.UPF
cat > scf.in << EOF
&control
  calculation = 'scf'
  prefix = 'gaas',
  pseudo_dir = './',
  outdir = './'
/
&system
 ibrav = 2,
  celldm(1) = 10.4749,
  nat = 2,
  ntyp = 2,
  ecutwfc = 40.0,
/
&electrons
/
ATOMIC_SPECIES
  Ga 1.0 Ga.pz-bhs.UPF
  As 1.0 As.pz-bhs.UPF
ATOMIC_POSITIONS crystal
  Ga 0.00 0.00 0.00
  As 0.25 0.25 0.25
K_POINTS automatic
6 6 6 1 1 1
EOF
```

All the parameters in this input file have been optimized separately. We can perform a test run to make sure that everything is in place: as usual we call `pw.x` from within our submission script:

```
mpirun -np 12 pw.x -npool 12 < scf.in > scf.out
```

Now we calculate vibrational frequencies at $\mathbf{q} = 0$. The input file for `ph.x` is similar to what we have seen in the previous section. The only differences are the two additional flags `epsil` and `zeu`:

```
$ cat > ph.in << EOF
phonons of GaAs
&inputph
  prefix = 'gaas',
  amass(1) = 69.723,
  amass(2) = 74.9216,
  epsil = .true.,
  zeu = .true.,
  fildyn = 'dyn',
  tr2_ph = 1.0d-14,
/
0.0 0.0 0.0
EOF
```

If we visit the documentation page, http://www.quantum-espresso.org/wp-content/uploads/Doc/INPUT_PH.html, and look for these flags we find:

[\[Back to Top\]](#)

epsil	LOGICAL
<i>Default:</i> .false.	

If .true. in a q=0 calculation for a non metal the macroscopic dielectric constant of the system is computed. Do not set epsil to .true. if you have a metallic system or q/=0: the code will complain and stop.

[\[Back to Top\]](#)

zeu	LOGICAL
<i>Default:</i> zeu=epsil	

If .true. in a q=0 calculation for a non metal the effective charges are computed from the dielectric response. This is the default algorithm. If epsil=.true. and zeu=.false. only the dielectric tensor is calculated.

Therefore these flags instruct `ph.x` to also evaluate the high-frequency (electronic) dielectric constant tensor of the system, as well as the Born effective charges. As we have seen in Lecture 4.2, these quantities are needed for calculating the IR activity of each mode and the static dielectric constant.

We execute `ph.x` using this input file from our batch script:

```
mpirun -np 12 ph.x -npool 12 < ph.in > ph.out
```

Towards the end of the output file we will find the following information:

```
Number of q in the star = 1
List of q in the star:
 1  0.000000000  0.000000000  0.000000000

Dielectric constant in cartesian axis

( 11.559429679  0.000000000  0.000000000 )
( 0.000000000  11.559429679  0.000000000 )
( 0.000000000  0.000000000  11.559429679 )

Effective charges (d Force / dE) in cartesian axis

atom 1 Ga
Ex ( 2.03189  0.00000  0.00000 )
Ey ( 0.00000  2.03189  0.00000 )
Ez ( 0.00000  0.00000  2.03189 )
atom 2 As
Ex ( -2.04609  0.00000  0.00000 )
Ey ( 0.00000 -2.04609  0.00000 )
Ez ( 0.00000  0.00000 -2.04609 )

Diagonalizing the dynamical matrix

q = ( 0.000000000  0.000000000  0.000000000 )

*****
freq ( 1) = 0.142309 [THz] = 4.746905 [cm-1]
freq ( 2) = 0.142309 [THz] = 4.746905 [cm-1]
freq ( 3) = 0.142309 [THz] = 4.746905 [cm-1]
freq ( 4) = 8.264350 [THz] = 275.669027 [cm-1]
freq ( 5) = 8.264350 [THz] = 275.669027 [cm-1]
freq ( 6) = 8.264350 [THz] = 275.669027 [cm-1]
*****
```

Here we recognize the high-frequency dielectric constant of GaAs, $\epsilon_\infty = 11.56$, and the Born effective charges of Ga and As, respectively $Z_{\text{Ga}}^* = 2.03$ and $Z_{\text{As}}^* = -2.04$ (in principle these two values should add up to zero, but we have some numerical error).

The calculated dielectric constant is about 6% higher than the experimental value, $\epsilon_\infty^{\text{exp}} = 10.89$. This overestimation is related to the band gap problem of DFT, which will be discussed in Lecture 5.1.

In the previous page we can see that the optical modes exhibit three identical frequencies, while we were expecting two degenerate TO modes and one LO mode at a higher frequency.

The reason why the modes are degenerate is that the calculation performed by `ph.x` missed a contribution, called the 'non-analytical part of the dynamical matrix'. This contribution can be calculated by using the dielectric constant and Born charges. Let us see how:

We create a new input file for `dynmat.x`:

```
$ cat > dynmat.in << EOF
&input
  fildyn = 'dyn',
  asr = 'simple',
  lperm = .true.,
  q(1)=1.0,
  q(2)=0.0,
  q(3)=0.0
/
EOF
```

Here `q(1)`, `q(2)`, and `q(3)` specify the direction along which we approach $\mathbf{q} \rightarrow 0$ (the LO-TO splitting is direction-dependent). When one of these numbers is nonzero, `dynmat.x` understands that it must read the dielectric constant and Born charges, calculate the LO-TO correction, and determine IR activities. The additional flag `lperm` specifies that we also want the static permittivity. After running `dynmat.x` we should obtain the following:

```
$ ./dynmat.x < dynmat.in
...
  IR activities are in (D/A)^2/amu units

# mode   [cm-1]   [THz]   IR
  1       0.00   0.0000   0.0000
  2       0.00   0.0000   0.0000
  3       0.00   0.0000   0.0000
  4      275.63   8.2632   2.6559
  5      275.63   8.2632   2.6559
  6      295.77   8.8670   2.6559

Electronic dielectric permittivity tensor (F/m units)
  11.559430   0.000000   0.000000
  0.000000  11.559430   0.000000
  0.000000   0.000000  11.559430

... with zone-center polar mode contributions
  13.080194  -0.000000   0.000000
  0.000000  13.310576   0.000000
  0.000000   0.000000  13.310576
```

We can see that now we have 2 degenerate TO modes at 34.17 meV, and 1 LO mode at 36.67 meV. The corresponding LO-TO splitting is 2.5 meV, in good agreement with the experimental value of 2.72 meV obtained by [Strauch & Dorner, J. Phys. Condens. Matter 2, 1457 \(1990\)](#).

The dielectric constants are highlighted in magenta. Here we see that $\epsilon_0 = 13.08$. For comparison the experimental value is $\epsilon_0^{\text{exp}} = 12.9$, therefore the relative deviation is of only 1.4%.

In the output file we also see the IR activities in units of $\text{debye}/\text{\AA}^2/\text{amu}$.

We should be careful in interpreting these data: IR measurements on thick films only detect TO modes; LO modes are seen in thin films at oblique incidence (Berreman effect), and the relative intensities of LO and TO modes depend on the angle of incidence and film thickness. Some illustrative measurements on III-V semiconductors can be found in [Ibáñez et al, J. Appl. Phys. 104, 033544 \(2008\)](#).

Note. The procedure that we used to calculate the LO-TO splitting can be bypassed by performing a direct calculation of phonon frequencies using a small but nonzero wavevector. For example we could simply use:

```
phonons of GaAs near Gamma
&inputph
  prefix = 'gaas',
  amass(1) = 69.723,
  amass(2) = 74.9216,
  fildyn = 'dyn',
  tr2_ph = 1.0d-14,
/
0.01 0.0 0.0
```

This gives two degenerate TO phonons at 34.17 meV and one LO phonon at 36.67 meV, in agreement with our previous calculation.

This alternative procedure is perfectly legitimate, but it does not provide us with Born charges, dielectric constants, and IR activities.

An introduction to density functional theory for experimentalists

Tutorial 4.2

Hands-on session

We create a new folder as usual:

```
> cd ~/scratch/summerschool; mkdir tutorial-4.2 ; cd tutorial-4.2
```

In this tutorial we study the phonon dispersion relations of **diamond**, **GaAs** and **SrTiO₃**.

Exercise 1

► Familiarize yourself with the calculation of the phonon dispersion relations of diamond, following step-by-step the procedure outlined in Tutorial 4.1.

Exercise 2

► Repeat the calculations illustrated in Tutorial 4.1 for GaAs. Note that all these calculations refer to zone-center phonons, $\mathbf{q} \rightarrow 0$.

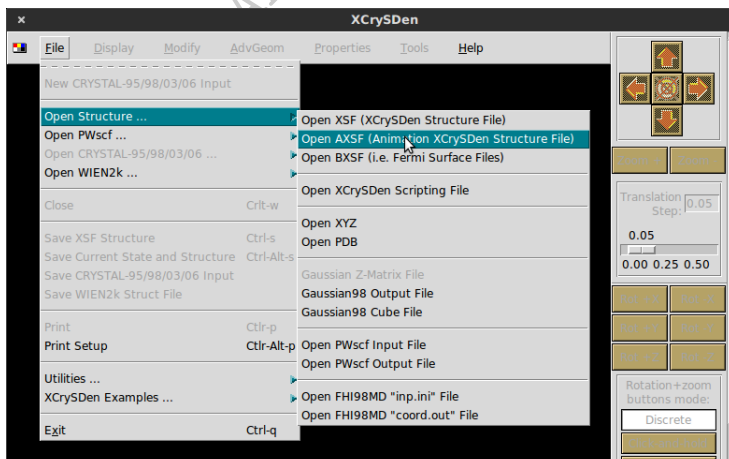
Exercise 3

If you succeeded to complete Exercise 2, you should now see in your working directory the file [dynmat.axsf](#). This file has been produced by the code `dynmat.x` that was invoked at the very end of the exercise.

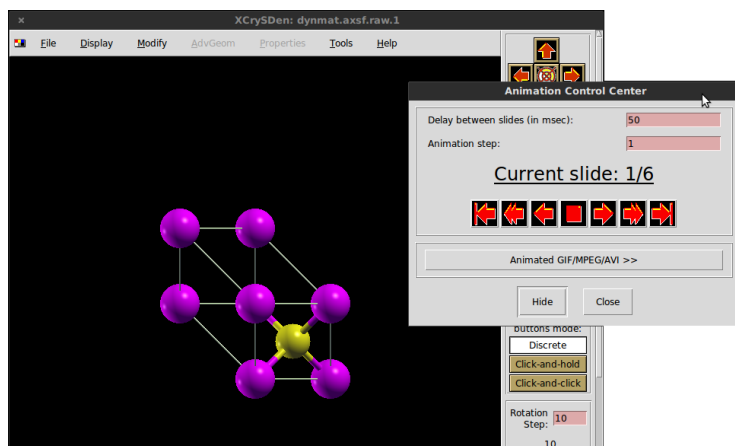
The file `dynmat.axsf` contains the vibrational eignemodes in a format which can be read and visualized by `xcrysdem`. Let us recall that these modes correspond to the atomic displacement patterns associated with a wavevector $\mathbf{q} \rightarrow 0$ along the direction x (this was specified in the input file `dynmat.in`).

► In this exercise we want to visualize these modes.

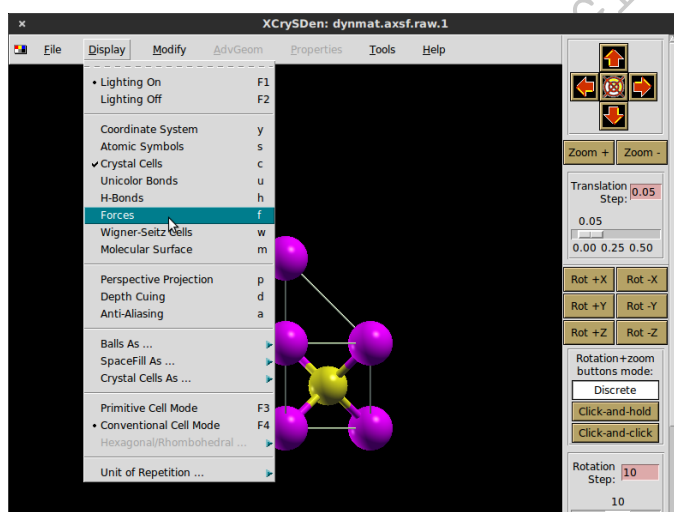
Let us call `xcrysdem` and go through the following steps. We open the `dynmat.axsf` file:



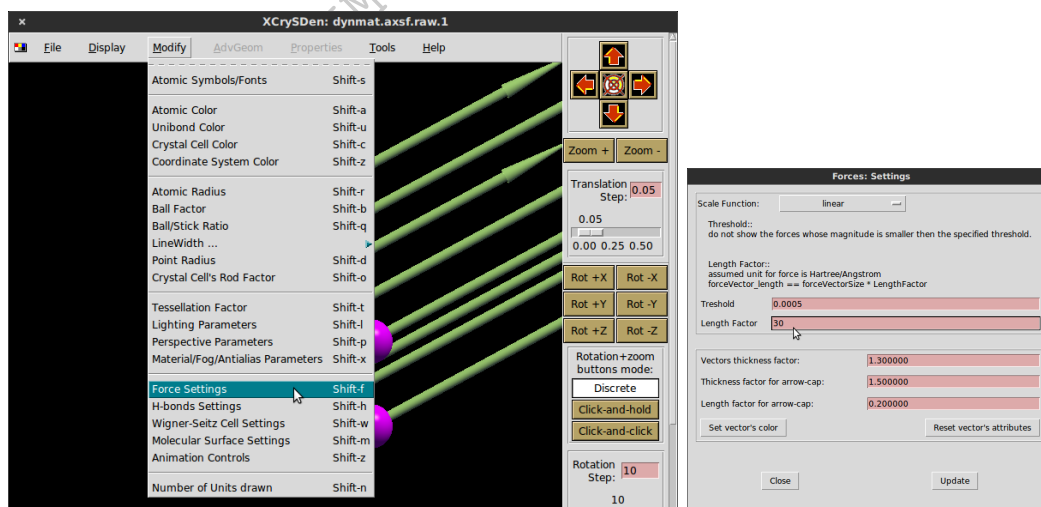
For the time being we ignore the window indicating the 'current slide'. This window will be used later to select the vibrational mode to be visualized.



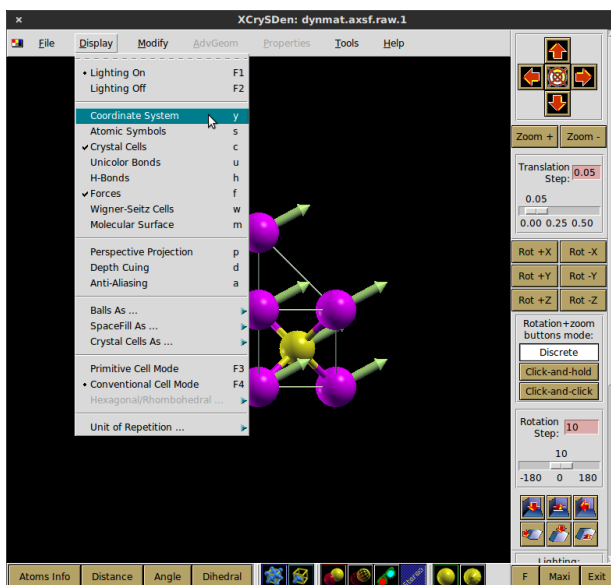
We activate the visualization of 'forces' (in our case the arrows will represent displacements, not forces, but the file format and the naming conventions are the same).



We adjust the length of the arrows by a uniform scaling

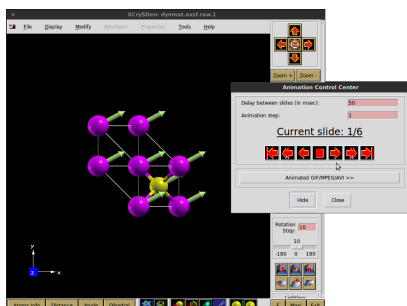


We ask the program to show the Cartesian axes.

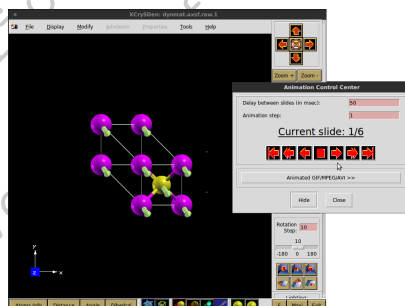


At this point we can use the window entitled 'Current Slide' to inspect each vibrational modes. The result should look similar to the following:

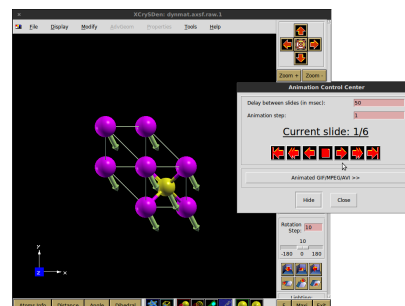
mode 1



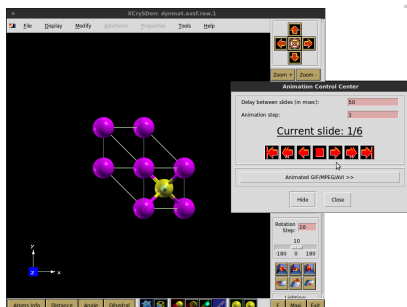
mode 2



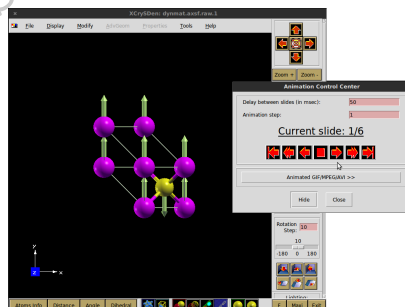
mode 3



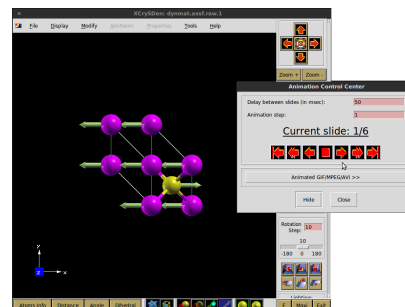
mode 4



mode 5



mode 6



Here we should be able to recognize the three translational modes at $\mathbf{q} = 0$ (modes 1–3), which should have $\omega = 0$. We can also recognize the optical phonons (modes 4–6): in these modes Ga and As atoms move in opposite directions. Furthermore, we see that in modes 4 and 5 the atomic displacements are along y and z , while in mode 6 the atoms displace along x . Since we have \mathbf{q} along the x axis (see `dynmat.in`), we can conclude that mode 6 is LO, while modes 4–5 are TO.

Exercise 4

- ▶ Calculate the phonon dispersion relations of GaAs, including the LO-TO splitting.

Note: In this exercise we need to combine what we have learned when we calculated the phonon dispersion relations of diamond, and what we did for calculating the LO-TO splitting in GaAs. The correct input file for `ph.x` is:

```
> cat > ph.in << EOF
phonons of GaAs
&inputph
  prefix = 'gaas',
  amass(1) = 69.723,
  amass(2) = 74.9216,
  epsilon = .true.,
  zeu = .true.,
  fildyn = 'dyn',
  tr2_ph = 1.0d-14,
  ldisp = .true.,
  nq1 = 4,
  nq2 = 4,
  nq3 = 4,
/
EOF
```

In this input file the flags in blue instruct `ph.x` to calculate the electronic dielectric permittivity tensor and the Born effective charges. These quantities are needed in order to correctly describe the LO-TO splitting. The lines in red instruct the code to calculate phonons on a uniform grid of $4 \times 4 \times 4$ **q**-points.

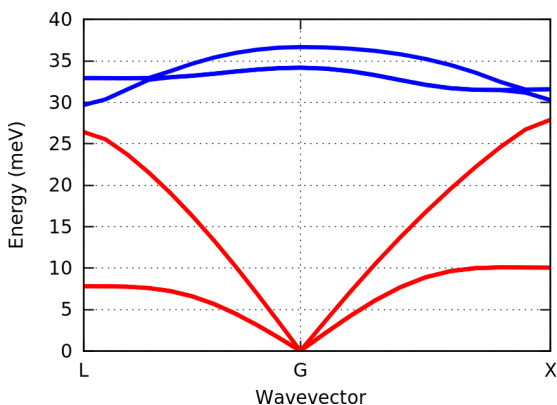
In order to obtain the dispersion relations, you will need to call `pw.x`, `ph.x`, `q2r.x`, and `matdyn.x` as we already did for diamond:

```
mpirun -np 12 pw.x -npool 4 <scf.in > scf.out
mpirun -np 12 ph.x -npool 4 <ph.in > ph.out
./q2r.x < q2r.in
./matdyn.x < matdyn.in
```

The input files `q2r.in` and `matdyn.in` must be prepared in the same way as for diamond. The input file `scf.in` for GaAs is the same as the one that we used in Tutorial 4.1.

- ▶ Plot the phonon dispersion relations of GaAs along the Brillouin zone path $L\Gamma X$ (as we did for diamond in Tutorial 4.1).
- ▶ Verify that the LO-TO splitting at Γ is the same as that calculated in Exercise 2.
- ▶ Compare your phonon dispersion relations with the inelastic neutron scattering data of [Strauch & Dörner, J. Phys. Condens. Matter 2, 1457 \(1990\)](#).

As a sanity check, you should be able to obtain dispersion relations resembling the following:



Exercise 5

In this exercise we want to calculate the phonon dispersion relations of cubic SrTiO₃, including the LO-TO splitting.

► We will use the input file `scf.in` from Tutorial 3.2 (with the optimized parameters given at the beginning of T3.2/Exercise 5), and the pseudopotentials from the same tutorial. Try to perform a test run using this setup, in order to make sure that everything goes smoothly.

► We now adapt to the case of SrTiO₃ the input file `ph.in` prepared in Exercise 4 for GaAs:

```
> cat > ph.in << EOF
phonons of STO
&inputph
prefix = 'sto',
amass(1) = 87.62,
amass(2) = 47.867,
amass(3) = 15.9994,
epsil = .true.,
zeu = .true.,
fildyn = 'dyn',
tr2_ph = 1.0d-14,
ldisp = .true.,
nq1 = 2,
nq2 = 2,
nq3 = 2,
/
EOF
```

Now you can execute `ph.x` using this input file.

You should find out that the execution takes much longer than in the case of GaAs.

Calculations on SrTiO₃ are more time-consuming than for GaAs since we now have 24 electrons per unit cell, and we are using a cutoff of 210 Ry. In these cases it is convenient to perform calculations in two steps:

- 1) We reduce massively the kinetic energy cutoff and the Brillouin zone sampling, and carry out the calculations until we manage to obtain our phonon dispersion relations. These results will be inaccurate and unreliable, but they will allow us to test every step very quickly.
- 2) Once we are confident about the complete procedure, we launch a 'production' calculation with all the optimized convergence parameters. This may take up to 30min on 12 cores, but now we can be confident that the calculation will complete successfully.

► Following this two-step procedure, reduce the cutoff to 50 Ry and the Brillouin zone sampling to 2 2 2 1 1 1, and calculate the phonon dispersion relations of SrTiO₃. The procedure is identical to what was done for GaAs on pag. 4.

Note: In this case we can plot the dispersions along the high-symmetry path $\Gamma XM\Gamma R$. The Cartesian coordinates of these points are $\Gamma : (0, 0, 0)$, $X : (0.5, 0, 0)$, $M : (0.5, 0.5, 0)$, $R : (0.5, 0.5, 0.5)$ in units of $2\pi/a$. The `matdyn.in` file will be:

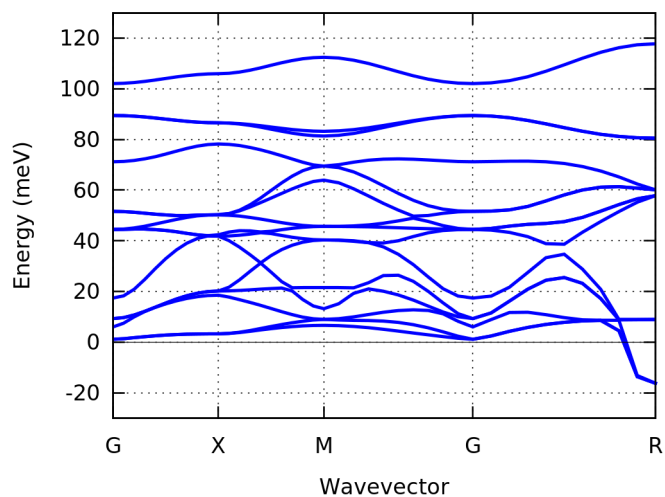
```
&input
asr = 'no',
flfrc = 'sto.fc',
flfrq = 'sto.freq'
/
41
```

```

0.000 0.000 0.000
0.050 0.000 0.000
0.100 0.000 0.000
0.150 0.000 0.000
0.200 0.000 0.000
0.250 0.000 0.000
0.300 0.000 0.000
0.350 0.000 0.000
0.400 0.000 0.000
0.450 0.000 0.000
0.500 0.000 0.000
0.500 0.050 0.000
0.500 0.100 0.000
0.500 0.150 0.000
0.500 0.200 0.000
0.500 0.250 0.000
0.500 0.300 0.000
0.500 0.350 0.000
0.500 0.400 0.000
0.500 0.450 0.000
0.500 0.500 0.000
0.450 0.450 0.000
0.400 0.400 0.000
0.350 0.350 0.000
0.300 0.300 0.000
0.250 0.250 0.000
0.200 0.200 0.000
0.150 0.150 0.000
0.100 0.100 0.000
0.050 0.050 0.000
0.000 0.000 0.000
0.050 0.050 0.050
0.100 0.100 0.100
0.150 0.150 0.150
0.200 0.200 0.200
0.250 0.250 0.250
0.300 0.300 0.300
0.350 0.350 0.350
0.400 0.400 0.400
0.450 0.450 0.450
0.500 0.500 0.500

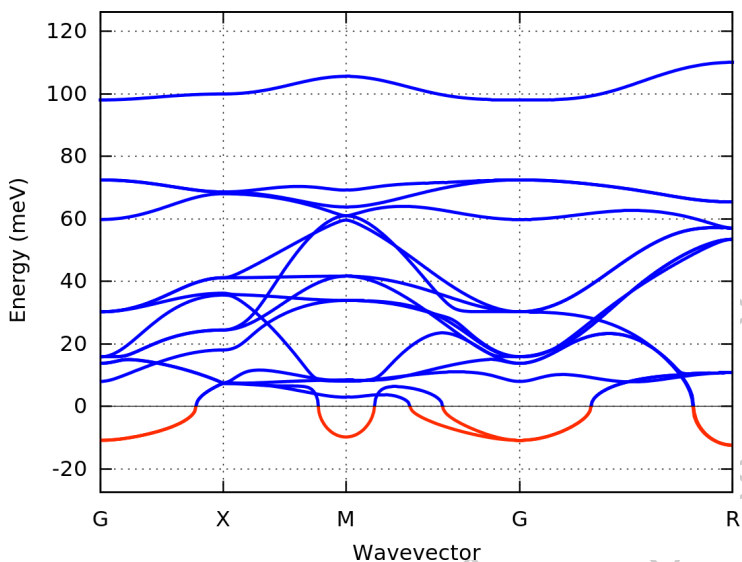
```

You should obtain something like the following:



► Now that we are confident about the entire procedure, we can perform a production run in order to obtain the final dispersion relations: you can repeat the entire procedure by using the optimized parameters `ecutwfc = 210` and `K_POINTS automatic 4 4 4 1 1 1`.

The final result should look as follows (in this plot a denser path of \mathbf{q} -points was used):



► Compare your result with those reported by [Ghosez et al, AIP Conf. Proc. 535, 102 \(2000\)](#) and by [Cancellieri et al, Nat. Commun. 7, 10386 \(2016\)](#).

► In the plot at the top of this page the curves in red denote imaginary frequencies, that is modes for which $\omega^2 < 0$. Since ω^2 represents the curvature of the potential energy surface, negative values imply that the system is in a local **maximum** of the energy surface, and is therefore **unstable** with respect to those vibrational modes.

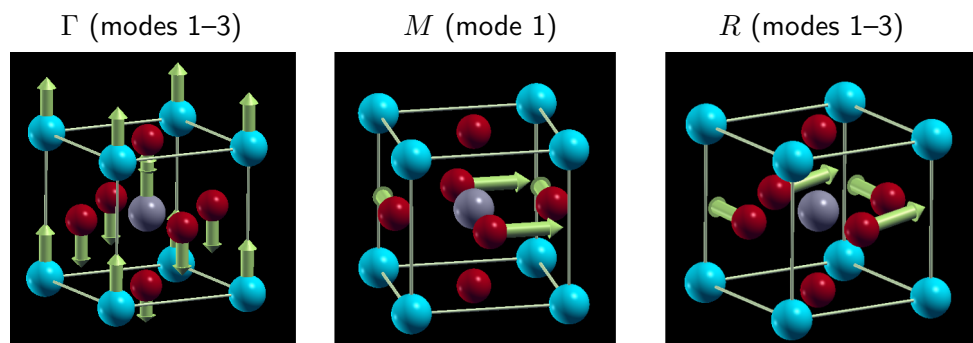
In order to better understand the origin of these instabilities, use `xcrysden` to visualize the displacement patterns of the soft modes at Γ , M , and R .

In order to use `xcrysden` you will need to generate `dynmat.axsf` files as in Exercise 3, using `dynmat.x`. The `dynmat.in` files should look as follows:

```
&input
  fildyn = 'dyn1'
/
```

where `dyn1` is for the Γ point, `dyn3` is for the M point, and `dyn4` is for the R point.

The soft modes, that is the modes with imaginary frequency, should look as follows:



The soft modes at Γ indicate the presence of a ferroelectric (quantum paraelectric) instability, the soft modes at M and R indicate instabilities against rotations of the TiO_6 octahedra. The origin of these soft modes lies in that we are performing ground-state calculations (ie at 0 K) for the cubic structure of SrTiO_3 , but the cubic structure is only stable above 110 K. These soft modes can be taken as an indication of the tendency of the system to lower its symmetry. The soft modes disappear when performing calculations on larger orthorhombic unit cells (containing 20 atoms per cell).

► Calculate the dielectric permittivity and IR activities of cubic SrTiO_3 , using `dynamt.x`.

In this case the appropriate input file is

```
&input
  fildyn = 'dyn1',
  asr = 'no',
  lperm = .true.,
  q(1)=1.0,
  q(2)=0.0,
  q(3)=0.0
/
```

Note that we are instructing `dynamt.x` to read data from `dyn1`, which corresponds to the Γ point. In this case we are also skipping the acoustic sum rule: the use of the sum rule is not meaningful when we have soft modes associated with structural instabilities.

Here we should be careful in interpreting our data: the static permittivity ϵ_0 is **not reliable** since we have soft modes with a large IR activity.

Generally speaking, calculations for the high-temperature cubic phase of SrTiO_3 should include temperature and quantum nuclear effects.