

Phonon interpolation and vibrational thermodynamic properties

Andrea Dal Corso

SISSA, CNR-IOM and MaX
Trieste (Italy)

Outline

- 1 Phonon interpolation and dispersions
- 2 q-vectors parallelization: a few ideas
- 3 Thermodynamics
 - Canonical partition function
 - Thermodynamic functions
- 4 Quasi-harmonic approximation

ph.x

The program `ph.x` solves this self-consistent linear system for $3 \times N_{at}$ perturbations at a fixed vector \mathbf{q} . With $\frac{\partial \rho(\mathbf{r})}{\partial \mathbf{u}_{s'\beta}(\mathbf{q})}$ for all perturbations, it calculates the dynamical matrix

$$D_{s\alpha s'\beta}(\mathbf{q})$$

at the given \mathbf{q} as discussed above. Diagonalizing this matrix we obtain $3 \times N_{at}$ frequencies $\omega_{\mathbf{q},\nu}$. By repeating this procedure for several \mathbf{q} we could plot $\omega_{\mathbf{q},\nu}$ as a function of \mathbf{q} and display the phonon dispersions. However, it is more convenient to adopt a different approach that requires the calculation of the dynamical matrix in a small set of vectors \mathbf{q} .

Phonon dispersions

The dynamical matrix of the solid:

$$D_{s\alpha s'\beta}(\mathbf{q}) = \frac{1}{\sqrt{M_s M_{s'}}} \sum_{\nu} \frac{\partial^2 E_{tot}}{\partial \mathbf{u}_{\mu s\alpha} \partial \mathbf{u}_{\nu s'\beta}} e^{i\mathbf{q}(\mathbf{R}_{\nu} - \mathbf{R}_{\mu})} \quad (1)$$

is a periodic function of \mathbf{q} with $D_{s\alpha s'\beta}(\mathbf{q} + \mathbf{G}) = D_{s\alpha s'\beta}(\mathbf{q})$ for any reciprocal lattice vector \mathbf{G} . Furthermore, due to the translational invariance of the solid, it does not depend on μ . Eq.1 is a Fourier expansion of a three dimensional periodic function. We have Fourier components only at the discrete values \mathbf{R}_{ν} of the Bravais lattice and we can write:

$$\frac{1}{\sqrt{M_s M_{s'}}} \frac{\partial^2 E_{tot}}{\partial \mathbf{u}_{\mu s\alpha} \partial \mathbf{u}_{\nu s'\beta}} = \frac{\Omega}{(2\pi)^3} \int d^3 \mathbf{q} D_{s\alpha s'\beta}(\mathbf{q}) e^{-i\mathbf{q}(\mathbf{R}_{\nu} - \mathbf{R}_{\mu})}. \quad (2)$$

Discrete Fourier transform

We can use the properties of the discrete Fourier transform and sample the integral in a uniform mesh of \mathbf{q} vectors. This will give the inter-atomic force constants only for a certain range of values of \mathbf{R}_ν neighbors of \mathbf{R}_μ .

In order to recall the main properties of the discrete Fourier transform, let us consider a one dimensional periodic function $f(x+a) = f(x)$ with period a . This function can be expanded in a Fourier series and will have a discrete set of Fourier components at $k_n = \frac{2\pi}{a}n$, where n is an integer (positive, negative or zero).

$$f(x) = \sum_n c_n e^{ik_n x}$$

where the coefficients of the expansion are:

Discrete Fourier transform - II

$$c_n = \frac{1}{a} \int_0^a f(x) e^{-ik_n x} dx.$$

In general, if $f(x)$ is a sufficiently smooth function, $c_n \rightarrow 0$ at large n . Now suppose that we discretize $f(x)$ in a uniform set of N points $x_j = j\Delta x$ where $\Delta x = a/N$ and $j = 0, \dots, N-1$, then we can calculate:

$$\tilde{c}_n = \frac{1}{N} \sum_{j=0}^{N-1} f(x_j) e^{-i\frac{2\pi}{N}nj},$$

\tilde{c}_n is a periodic function of n and $\tilde{c}_{n+N} = \tilde{c}_n$. So, if N is sufficiently large that $c_n = 0$ when $|n| \geq N/2$, \tilde{c}_n is a good approximation of c_n for $|n| < N/2$ and the function

Discrete Fourier transform - III

$$f(x) = \sum_{n=-N/2}^{n=N/2} \tilde{c}_n e^{ik_n x}$$

is a good approximation of the function $f(x)$ also on the points x different from x_j . In three dimensions the discretization of Eq. 2 on a uniform mesh of \mathbf{q}_i vectors is:

$$\frac{\partial^2 E_{tot}}{\partial \mathbf{u}_{\mu s \alpha} \partial \mathbf{u}_{\nu s' \beta}} = \frac{1}{N_q} \sum_{i=1}^{N_q} C_{s\alpha s'\beta}(\mathbf{q}_i) e^{-i\mathbf{q}_i(\mathbf{R}_\nu - \mathbf{R}_\mu)},$$

where we defined $C_{s\alpha s'\beta}(\mathbf{q}) = \sqrt{M_s M_{s'}} D_{s\alpha s'\beta}(\mathbf{q})$. Since

$\frac{\partial^2 E_{tot}}{\partial \mathbf{u}_{\mu s \alpha} \partial \mathbf{u}_{\nu s' \beta}}$ depends only on the vector $\mathbf{R} = \mathbf{R}_\nu - \mathbf{R}_\mu$, we can call

q2r.x

$C_{s\alpha s'\beta}(\mathbf{R}) = \frac{\partial^2 E_{tot}}{\partial \mathbf{u}_{\mu s\alpha} \partial \mathbf{u}_{\nu s'\beta}}$ and write the relationship:

$$C_{s\alpha s'\beta}(\mathbf{R}) = \frac{1}{N_q} \sum_{i=1}^{N_q} C_{s\alpha s'\beta}(\mathbf{q}_i) e^{-i\mathbf{q}_i \mathbf{R}}.$$

The code `q2r.x` reads a set of dynamical matrices obtained for a uniform mesh of \mathbf{q}_i vectors and calculates, using this equation, the inter-atomic force constants for some neighbors of the point $\mathbf{R} = 0$.

matdyn.x

If the dynamical matrix is a sufficiently smooth function of \mathbf{q} , the inter-atomic force constants decay sufficiently rapidly in real space and we can use Eq. 1 limiting the sum over ν to the few neighbors of \mathbf{R}_μ for which we have calculated the inter-atomic force constants. With the present notation Eq. 1 becomes:

$$C_{S\alpha S'\beta}(\mathbf{q}) = \sum_{\mathbf{R}} C_{S\alpha S'\beta}(\mathbf{R}) e^{i\mathbf{q}\mathbf{R}}, \quad (3)$$

a relationship that allows the interpolation of the dynamical matrix at arbitrary \mathbf{q} , by a few inter-atomic force constants. The program `matdyn.x` reads the inter-atomic force constants calculated by `q2r.x` and calculates the dynamical matrices at an arbitrary \mathbf{q} using this equation.

This procedure fails in two cases:

- In metals when there are Kohn anomalies. In this case $D_{s\alpha s'\beta}(\mathbf{q})$ is not a smooth function of \mathbf{q} and the inter-atomic force constants are long range.
- In polar insulators where the atomic displacements generate long range electrostatic interactions and the dynamical matrix is non analytic for $\mathbf{q} \rightarrow 0$. This case, however, can be dealt with by calculating the Born effective charges and the dielectric constant of the material.

Use of symmetry

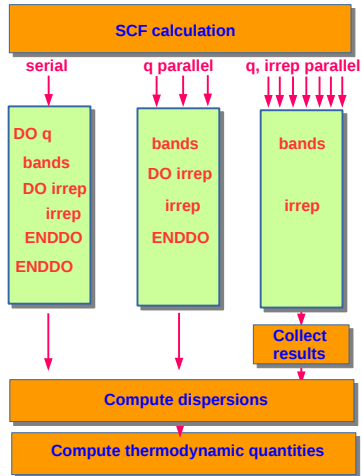
Phonon dispersions require the DFPT calculation on a uniform mesh $N_{q_1} \times N_{q_2} \times N_{q_3} = N_q$ of \mathbf{q} vectors. The CPU time can be roughly estimated as

$$N_q \times 3 \times N_{\text{at}} \times T_{\text{scf}}$$

where T_{scf} is the CPU time of a single self-consistent calculation. Using symmetry the \mathbf{q} -vector mesh is reduced to a set of \bar{N}_q non equivalent \mathbf{q} vectors. The calculation of the dynamical matrix at each \mathbf{q} vector requires an amount of CPU time roughly proportional to the size of its star of \mathbf{q} vectors. So low symmetry \mathbf{q} vectors require much more CPU time than high symmetry \mathbf{q} vectors mainly because `ph.x` uses only the symmetries of the small group of \mathbf{q} to reduce the \mathbf{k} points.

Use of symmetry - II

On the other hand, from the dynamical matrix at \mathbf{q} we can obtain, for free, the dynamical matrices of the star of \mathbf{q} that is larger for low symmetry \mathbf{q} . Not all the $3 \times N_{\text{at}}$ perturbations have to be calculated simultaneously at each \mathbf{q} . Choosing displacement patterns that transform according to an irreducible representation (irrep) of the small group of \mathbf{q} , the number of patterns that transform among themselves is equal to the dimension of the irreducible representation. For standard point groups the maximum dimension is 3, while for \mathbf{q} at zone border and nonsymmorphic point groups the maximum dimension could be larger, up to 6.



q-vectors parallelization

The phonon code is parallelized as the `pw.x` code: reciprocal lattice vectors \mathbf{G} can be distributed among CPUs and/or \mathbf{k} vectors used for the integration over the Brillouin zone can be distributed among pools of CPUs. In addition, it is possible to calculate in parallel the contribution to the dynamical matrix of each irrep for each \mathbf{q} vector.

Different \mathbf{q} vectors might require quite different CPU time so the dynamical matrix of each \mathbf{q} vector is calculated independently. The two input parameters `start_q` and `last_q` allow the choice of the \mathbf{q} vector in the list of \bar{N}_q vectors. To parallelize only on \mathbf{q} vectors it is convenient to send many runs on different CPUs with different `outdir` directories. Preliminarily, the `outdir` produced by `pw.x` has to be copied in all the `outdir` directories where `ph.x` will run.

q vectors and irreps parallelization

q vectors and irreps parallelization is slightly more complex because each run must use the same displacement patterns. Therefore, after running `pw.x`, a preliminary run of `ph.x` with the two flags `start_irr=0` and `last_irr=0` calculates and saves on disk the displacement patterns for all q vectors and irreps. Then `ph.x` can run separately on different machines or on different CPUs. Each run calculates one or more irrep as specified by the four variables `start_q`, `last_q`, `start_irr`, `last_irr`. The `outdir` produced by `pw.x` and by the preliminary run of `ph.x` has to be copied in all the `outdir` directories where `ph.x` will run.

Collecting the results

After running `ph.x` for all **q** vectors and all irreps the results are collected in a single `outdir` directory and `ph.x` is run again to produce the dynamical matrices. The files with the contributions of the irreps to the dynamical matrix are in the `outdir/_phprefix.phsave` directories and are called `data-file.xml.#iq.#irr`. All these files have to be copied in one `outdir/_phprefix.phsave` directory and `ph.x` must be run on this `outdir` without any flag. Note that if the file `data-file.xml.#iq.#irr` for some **q**-vector or for some irreps is missing `ph.x` will recalculate the missing contribution to the dynamical matrix.

Image parallelization

It is possible to run `ph.x` using the image parallelization. The code is run with the option `-ni=number of images` and the total number of processors is divided into `ni` groups, each group running a certain number of `irrep`. The division of `irrep` among images is done at the beginning of the run by the driver and the images run independently without talking to each other. The number of processors must be a multiple of the number of images. Finally you need to run `ph.x` another time to collect the results and obtain the frequencies. The last run of `ph.x` as well as the initial run of `pw.x` must be done with a number of processors equal to the number of processors used with images divided by the number of images. The number of pools must be kept invariant.

Thermo_pw

The image features of `ph.x` are further developed in the `thermo_pw` code. This is presently available at:

http://www.qe-forge.org/gf/project/thermo_pw/

This code allows to run asynchronously the different images, so that it is not necessary to divide the work among images at the beginning of the run. Each image runs a new task as soon as it finishes to run the current task until there are tasks to run. Finally the results are collected automatically and `q2r.x` and `matdyn.x` are run after the phonon calculation.

The canonical partition function

The vibrational energy of a solid whose phonon modes have frequencies $\omega_{\mathbf{q},\nu}$ depends on the number of phonons $n_{\mathbf{q},\nu}$ in each mode:

$$E_i = \sum_{\mathbf{q},\nu} \left(n_{\mathbf{q},\nu} + \frac{1}{2} \right) \hbar \omega_{\mathbf{q},\nu},$$

where i indicates the set of integer numbers $n_{\mathbf{q},\nu}$. At a given temperature T the probability that the solid has a certain energy E_i can be calculated by statistical methods and it is:

$$P(E_i) = \frac{1}{Z} e^{-\beta E_i}$$

where Z is the canonical partition function defined as

$$Z = \sum_i e^{-\beta E_i},$$

the sum is over all the possible sets of integers $n_{\mathbf{q},\nu}$ and $\beta = 1/K_B T$ (K_B is the Boltzmann constant). We can write

$$Z = \sum_i e^{-\beta E_i} = \prod_{\mathbf{q},\nu} \left(\sum_{n=0}^{\infty} e^{-(n+1/2)\beta\hbar\omega_{\mathbf{q},\nu}} \right)$$

Making the sum over n and taking the logarithm gives:

$$\begin{aligned} \ln Z &= \sum_{\mathbf{q},\nu} \ln \left[\frac{e^{-\frac{1}{2}\beta\hbar\omega_{\mathbf{q},\nu}}}{1 - e^{-\beta\hbar\omega_{\mathbf{q},\nu}}} \right] = - \sum_{\mathbf{q},\nu} \ln \left[2 \sinh \left(\frac{\beta\hbar\omega_{\mathbf{q},\nu}}{2} \right) \right] \\ &= -\beta \sum_{\mathbf{q},\nu} \frac{\hbar\omega_{\mathbf{q},\nu}}{2} - \sum_{\mathbf{q},\nu} \ln \left[1 - e^{-\beta\hbar\omega_{\mathbf{q},\nu}} \right]. \end{aligned}$$

The thermodynamic functions

The vibrational energy per cell of the solid is:

$$U = \frac{1}{N} \frac{\sum_i E_i e^{-\beta E_i}}{\sum_i e^{-\beta E_i}} = -\frac{1}{N} \frac{\partial \ln Z}{\partial \beta},$$

where N is the number of unit cells of the solid. The Helmholtz free energy per cell ($F = U - TS$ where S is the entropy per cell), is given by:

$$F = -\frac{1}{N} \frac{1}{\beta} \ln Z.$$

Using the expression of $\ln Z$ in terms of the phonon frequencies, we have:

$$U = \frac{1}{N} \sum_{\mathbf{q}, \nu} \frac{\hbar \omega_{\mathbf{q}, \nu}}{2} + \frac{1}{N} \sum_{\mathbf{q}, \nu} \frac{\hbar \omega_{\mathbf{q}, \nu}}{e^{\beta \hbar \omega_{\mathbf{q}, \nu}} - 1}.$$

The phonon density of states

Introducing the phonon density of states per cell:

$$g(\omega) = \frac{1}{N} \sum_{\mathbf{q}, \nu} \delta(\omega - \omega_{\mathbf{q}, \nu})$$

we can write the thermodynamic functions as one dimensional integrals over the frequencies:

$$F = \frac{1}{\beta} \int_0^{\infty} d\omega g(\omega) \ln \left[2 \sinh\left(\frac{\beta \hbar \omega}{2}\right) \right],$$

$$U = \int_0^{\infty} d\omega g(\omega) \frac{\hbar \omega}{2} + \int_0^{\infty} d\omega g(\omega) \frac{\hbar \omega}{e^{\beta \hbar \omega} - 1},$$

$$S = \frac{1}{T} (U - F).$$

The thermodynamic functions

The isochoric heat capacity of one unit cell:

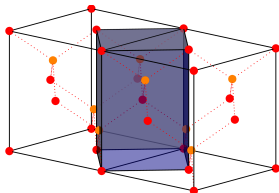
$$\begin{aligned} C_V &= \left. \frac{\partial U}{\partial T} \right|_V = K_B \int_0^\infty d\omega g(\omega) \left[\frac{\beta \hbar \omega / 2}{\sinh(\frac{\beta \hbar \omega}{2})} \right]^2 \\ &= K_B \int_0^\infty d\omega g(\omega) e^{\beta \hbar \omega} \left[\frac{\beta \hbar \omega}{e^{\beta \hbar \omega} - 1} \right]^2. \end{aligned}$$

All these functions are plotted by the `thermo_pw` software after a phonon dispersion calculation. Other small codes that compute these quantities from the phonon density of states are available in the QE package or on the web.

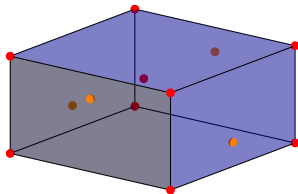
NB: require stable systems where all frequencies are positive.

The α - β structural transition in Tin - I

The structure of Tin in the α phase ($T < 13C$) is the diamond structure (note a centered tetragonal unit cell with $c = \sqrt{2}a$):

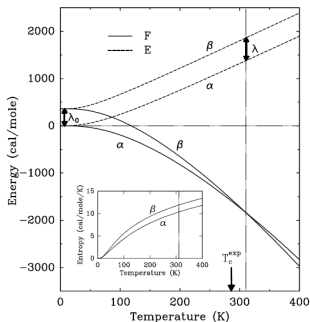


The structure of Tin in the β phase ($T > 13C$) ($c \approx 0.55a$):



The α - β structural transition in Tin - II

Phase stability at zero pressure is studied comparing the Helmholtz free energies at different temperatures:



Equilibrium volume of a solid at constant temperature T

We consider a solid at temperature T and at pressure p . If the volume of the solid changes by $d\Omega$ the solid does a work $pd\Omega$ against the pressure p . To keep the temperature constant the solid absorbs some heat dQ . According to the second law of thermodynamic $dQ = TdS$ where S is the entropy. According to the first law of thermodynamic the change of the internal energy is equal to the adsorbed heat minus the work done $dU = TdS - pd\Omega$. Therefore, at constant temperature:

$$p = - \left(\frac{dU}{d\Omega} - T \frac{dS}{d\Omega} \right) \Big|_T = - \frac{\partial F(\Omega, T)}{\partial \Omega} \Big|_T$$

At temperature T and zero pressure, the solid is in equilibrium at the volume Ω that minimizes the free energy.

The volume as a function of temperature

The total free energy per unit cell of volume Ω is:

$$F(\Omega, T) = U_0(\Omega) + F_{ph}(\Omega, T) + F_{el},$$

where $U_0(\Omega)$ is the DFT total energy, $F_{ph}(\Omega, T) = F$ is the vibrational free energy, and

F_{el} is the electron excitation energy neglected in the following.

Minimizing $F(\Omega, T)$ at each T , we obtain the equilibrium volume as a function of temperature, $\Omega_0(T)$.

The volume thermal expansion is defined as:

$$\beta = \frac{1}{\Omega_0(T)} \frac{d\Omega_0(T)}{dT}.$$

Free energy minimization

The bulk modulus B_0^T is the inverse of the compressibility. Calculated at constant temperature it is:

$$\frac{1}{B_0} = K^T = -\frac{1}{\Omega} \left(\frac{\partial \Omega}{\partial p} \right)_T.$$

The Murnaghan equation:

$$F(\Omega) = F(\Omega_0) + \frac{B_0 \Omega}{B'_0} \left[\frac{1}{B'_0 - 1} \left(\frac{\Omega_0}{\Omega} \right)^{B'_0} + 1 \right] - \frac{B_0 \Omega_0}{B'_0 - 1},$$

obtained assuming a constant derivative of the bulk modulus B'_0 with respect to pressure, depends on Ω_0 , B_0 , and B'_0 . By fitting the free energy at each temperature we obtain $\Omega_0(T)$, $B_0(T)$, and $B'_0(T)$.

Thermal expansion from Grüneisen parameters

The Grüneisen parameters are defined by:

$$\gamma_{\mathbf{q},\nu} = -\frac{\Omega}{\omega_{\mathbf{q},\nu}} \frac{d\omega_{\mathbf{q},\nu}}{d\Omega}.$$

Thermal expansion in terms of Grüneisen parameters:

$$\beta = \frac{1}{\Omega(T)} \left. \frac{\partial \Omega(T)}{\partial T} \right|_p = -\frac{1}{\Omega(T)} \left. \frac{\partial \Omega(T)}{\partial p(T)} \right|_T \left. \frac{\partial p(T)}{\partial T} \right|_\Omega = -\frac{1}{B_0(T)} \frac{\partial^2 F(\Omega, T)}{\partial T \partial \Omega}.$$

Only the vibrational part of the free energy depends on T :

$$\beta = \frac{1}{B_0(T)} \sum_{\mathbf{q},\nu} c_{\mathbf{q},\nu} \gamma_{\mathbf{q},\nu},$$

where $c_{\mathbf{q},\nu}$ is given by:

$$c_{\mathbf{q},\nu} = \frac{\hbar\omega_{\mathbf{q},\nu}}{\Omega} \frac{\partial}{\partial T} \left[\frac{1}{e^{\beta\hbar\omega_{\mathbf{q},\nu}} - 1} \right].$$

Isobaric heat capacity, isoentropic bulk modulus

The heat capacity measured at constant pressure can be obtained as:

$$C_P - C_V = T\Omega B_0 \beta^2.$$

The difference between the bulk modulus measured at constant entropy (without heat exchange) or at constant temperature can be obtained as:

$$\frac{1}{B_0^S} - \frac{1}{B_0^T} = -\frac{T\Omega\beta^2}{C_P}.$$

Anisotropic solids: Murnaghan equation

The previous approach is simple to apply to cubic solids. The change of the unit cell volume is equivalent to the change of the crystal parameter a . For anisotropic solids it is still valid, but for each volume there are several crystallographic parameters. For tetragonal, hexagonal, and trigonal solids there are two parameters ($a, c/a$, or $a, \cos\alpha$), for orthorhombic solids three ($a, b/a$, and c/a), for monoclinic solids four ($a, b/a, c/a, \cos\gamma$) and for triclinic solids six ($a, b/a, c/a, \cos\alpha, \cos\beta, \cos\gamma$).

There are two choices:

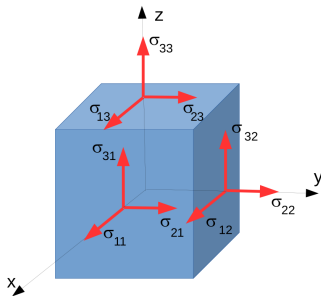
- Change only a and keep constants all angles and ratios.
Simple but approximated.
- Optimize all the parameters at fixed volume.
Less simple and less approximated.

Anisotropic solids: Strain

Instead of changing the volume, we change the size and the shape of the unit cell. The strain applied to a solid is described by a symmetric 3×3 tensor $\epsilon_{i,j}$. Calling $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ the primitive vectors of the unperturbed solid, and $\mathbf{a}'_1, \mathbf{a}'_2, \mathbf{a}'_3$ those of the strained solid, we have:

$$\mathbf{a}'_{j,i} = \mathbf{a}_{j,i} + \sum_{k=1}^3 \epsilon_{j,k} \mathbf{a}_{k,i}.$$

Anisotropic solids: Stress



We call stress $\sigma_{i,j}$ the force (\mathbf{f}) per unit area A_j (j indicates the perpendicular to the surface) present in a solid. The work per cell made by a solid with a stress $\sigma_{i,j}$ to change its strain by $d\epsilon_{i,j}$ is $dW = -\Omega \sum_{i,j} \sigma_{i,j} d\epsilon_{i,j}$, where Ω is the unperturbed unit cell volume.

Uniform pressure

A uniform pressure corresponds to a stress:

$$\sigma_{i,i} = -p; \quad p = -\frac{1}{3} \text{tr}(\sigma_{i,j}) = -\frac{1}{3} \sum_{i=1}^3 \sigma_{i,i}.$$

The work made by the solid becomes:

$$dW = -\Omega \sum_{i,j} \sigma_{i,j} d\epsilon_{i,j} = \Omega p \sum_{i=1}^3 d\epsilon_{i,i} = p d\Omega.$$

Equation of state for an anisotropic solid

When the strain of the solid changes of $d\epsilon_{i,j}$, at constant T and stress $\sigma_{i,j}$, the solid makes some work and adsorb some heat to keep the temperature constant so the internal energy of the solid changes as:

$$dU = TdS + \Omega \sum_{i,j} \sigma_{i,j} d\epsilon_{i,j}.$$

From this equation we obtain the equation of state (at constant temperature):

$$\sigma_{i,j} = \frac{1}{\Omega} \left(\frac{\partial U}{\partial \epsilon_{i,j}} - T \frac{\partial S}{\partial \epsilon_{i,j}} \right)_T = \frac{1}{\Omega} \frac{\partial F}{\partial \epsilon_{i,j}} \Big|_T.$$

For a solid with no stress, the strain minimizes the Helmholtz free energy.

Crystal parameters as a function of the temperature and thermal expansion

Considering only strains that do not change the Bravais lattice, we fit the free energy with a quadratic or quartic polynomial of the crystal parameters and minimize it. From the minimum we obtain $a(T)$, $b(T)$, $c(T)$ etc. The thermal expansion measures the strain induced by a change of temperature at constant stress:

$$\alpha_{i,j} = \left. \frac{\partial \epsilon_{i,j}}{\partial T} \right|_{\sigma}$$

Thermal expansion from crystal parameters

In a cubic solid:

$$d\epsilon_{11} = \frac{da}{a}; \quad \alpha_{11} = \frac{1}{a(T)} \frac{da(T)}{dT}.$$

In a tetragonal or hexagonal solid:

$$d\epsilon_{11} = \frac{da}{a}; \quad d\epsilon_{33} = \frac{dc}{c};$$
$$\alpha_{11} = \frac{1}{a(T)} \frac{da(T)}{dT}; \quad \alpha_{33} = \frac{1}{c(T)} \frac{dc(T)}{dT}.$$

In an orthorombic solid:

$$d\epsilon_{11} = \frac{da}{a}; \quad d\epsilon_{22} = \frac{db}{b}; \quad d\epsilon_{33} = \frac{dc}{c};$$
$$\alpha_{11} = \frac{1}{a(T)} \frac{da(T)}{dT}; \quad \alpha_{22} = \frac{1}{b(T)} \frac{db(T)}{dT}; \quad \alpha_{33} = \frac{1}{c(T)} \frac{dc(T)}{dT}.$$

Volume thermal expansion

From the thermal expansion tensor we have:

$$\beta = \frac{1}{\Omega} \frac{\partial \Omega}{\partial T} \Big|_{\rho} = \sum_{i=1}^3 \frac{\partial \epsilon_{i,i}}{\partial T} \Big|_{\sigma} = \sum_{i=1}^3 \alpha_{i,i} = \text{Tr}(\alpha)$$

Elastic constants and compliances

The concepts that generalize the bulk modulus and the compressibility are the elastic constants and compliances. The elastic constants, at constant temperature, are defined as the stress induced by a strain:

$$C_{ij,kl}^T = \left. \frac{\partial \sigma_{i,j}}{\partial \epsilon_{k,l}} \right|_T.$$

The inverse of the elastic constants tensor are the elastic compliances that measure the strain induced by a stress:

$$S_{ij,kl}^T = \left. \frac{\partial \epsilon_{i,j}}{\partial \sigma_{k,l}} \right|_T.$$

Available only at $T = 0$ in thermo_pw.

Thermal expansion from Grüneisen parameters

The generalization of the Grüneisen parameters are:

$$\gamma_{\mathbf{q},\nu}^{i,j} = -\frac{1}{\omega_{\mathbf{q},\nu}} \frac{d\omega_{\mathbf{q},\nu}}{d\epsilon_{i,j}},$$

and the thermal expansion tensor becomes:

$$\alpha_{i,j} = \left. \frac{\partial \epsilon_{i,j}}{\partial T} \right|_{\sigma} = - \sum_{k,l} \left. \frac{\partial \epsilon_{i,j}}{\partial \sigma_{k,l}} \right|_T \left. \frac{\partial \sigma_{k,l}}{\partial T} \right|_{\epsilon} = -\frac{1}{\Omega} \sum_{k,l} S_{ij,kl}^T \frac{\partial^2 F(\epsilon, T)}{\partial T \partial \epsilon_{k,l}}.$$

Only the vibrational part of the free energy depends on T :

$$\alpha_{i,j} = \sum_{k,l} S_{ij,kl}^T \sum_{\mathbf{q},\nu} c_{\mathbf{q},\nu} \gamma_{\mathbf{q},\nu}^{k,l}.$$

Heat capacity at constant stress, isoentropic elastic constants and compliances

The difference between the constant stress and constant strain heat capacities is:

$$C^\sigma - C^\epsilon = T\Omega \sum_{ij,kl} \alpha_{i,j} C_{ij,kl}^T \alpha_{k,l}.$$

The difference between the isoentropic and isothermal elastic compliances is:

$$S_{ij,kl}^S - S_{ij,kl}^T = -\frac{T\Omega \alpha_{i,j} \alpha_{k,l}}{C^\sigma}.$$

The difference between the isoentropic and isothermal elastic constants is:

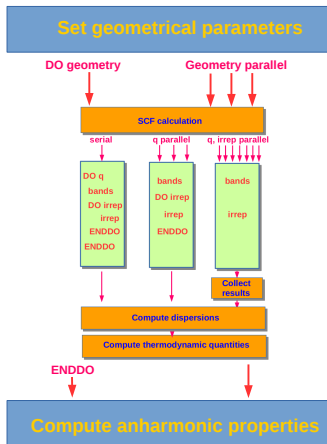
$$C_{ij,kl}^S - C_{ij,kl}^T = \frac{T\Omega b_{i,j} b_{k,l}}{C^\epsilon},$$

where the $b_{i,j}$ are the so called thermal stresses:

$$b_{i,j} = \left. \frac{\partial \sigma_{i,j}}{\partial T} \right|_{\epsilon} = - \sum_{k,l} C_{ij,kl}^T \alpha_{k,l},$$

calculated from the product of the elastic constants and the thermal expansion.

The thermo_pw software



Bibliography

- 1 S. Baroni, S. de Gironcoli, A. Dal Corso, and P. Giannozzi, Rev. Mod. Phys. **73**, 515 (2001).
- 2 R. di Meo, A. Dal Corso, P. Giannozzi, and S. Cozzini, Calculation of phonon dispersions on the GRID using Quantum ESPRESSO, ICTP lecture notes **24**, 163 (2009).
- 3 P. Pavone, S. Baroni, and S. de Gironcoli, Phys. Rev. B **57**, 10421 (1998).
- 4 J.F. Nye, Physical properties of crystals, Oxford University press, Oxford (1957)
- 5 D.C. Wallace, Thermodynamics of crystals, J. Wiley and Sons, (1972), Dover edition (1998).
- 6 A. Dal Corso, thermo.pdf. Notes distributed with the thermo_pw documentation.