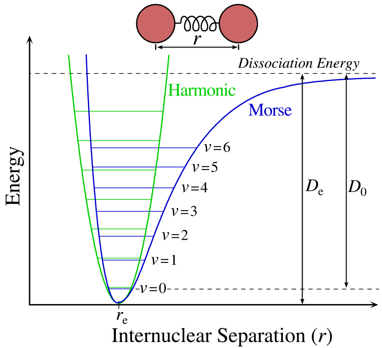
The quasi-harmonic approximation (QHA)

M. Palumbo

19/01/2017
Trieste, Italy

Limitations of the harmonic approximation

$$E_{\rm tot}(R_I, u_I) = E_{\rm tot}(R_I) + \sum_{I,\alpha} \frac{\partial E_{tot}}{\partial u_{I\alpha}} u_{I\alpha} + \frac{1}{2} \sum_{I,\alpha} \frac{\partial^2 E_{tot}}{\partial u_{I\alpha} \partial u_{J\beta}} u_{I\alpha} u_{J\beta} + \ldots$$



The harmonic potential is a good approximation at low energy (temperature), but it becomes inadequate as the temperature increases.

Limitations of the harmonic approximation:

- phonons have infinite lifetime
- phonons do not interact
- no thermal expansion
- no thermal transport

Beyond the harmonic approximation

- More terms in the Taylor expansion of the energy
  - Advantages: capture the full anharmonicy at low and high T
  - Drawbacks: many terms to be considered, difficult to be implemented
- Molecular dynamics simulations (MD)
  - Advantages: capture the full anharmonicy
  - Drawbacks: difficult at low T, one simulation for each temperature
- Quasi-harmonic approximation (QHA)
  - Advantages: relatively simple for isotropic solids (not so for anisotropic)
  - Drawbacks: does not capture the full anharmonicy, not accurate enough at high T

The quasi-harmonic approximation (QHA)

Vibrational Helmoltz energy of a set of harmonic oscillators

$$F^{\mathrm{vib}}(T) = \frac{1}{2}\sum_{\vec{q},\nu} \hbar\omega(\vec{q},\nu) + k_{\mathrm{B}}T\sum_{\vec{q},\nu} \ln[1 - \exp(\frac{-\hbar\omega(\vec{q},\nu)}{k_{\mathrm{B}}T})]$$

Quasi-harmonic "extension"

$$F^{\mathrm{vib}}(X,T) = \frac{1}{2}\sum_{\vec{q},\nu} \hbar\omega(\vec{q},\nu,X) + k_{\mathrm{B}}T\sum_{\vec{q},\nu} \ln[1 - \exp(\frac{-\hbar\omega(\vec{q},\nu,X)}{k_{\mathrm{B}}T})]$$

$X \rightarrow$ volume or lattice paratemers
For a given X, phonons calculations are performed in the harmonic approximation

$$min[F(X,T) = U(X) + F^{\mathrm{vib}}(T,X)]$$

$U(X)$ total energy for a given X

Other thermophysical properties are then obtained from $F_{min}(T,X)$

Isotropic systems

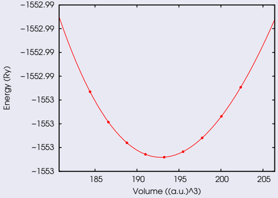
$X = V$

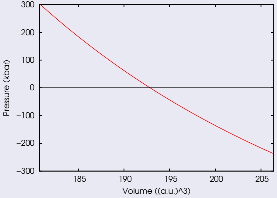
$$F^{\mathrm{vib}}(V,T) = \frac{1}{2}\sum_{\vec{q},\nu}\hbar\omega(\vec{q},\nu,V) + k_{\mathrm{B}}T\sum_{\vec{q},\nu}\ln[1-\exp(\frac{-\hbar\omega(\vec{q},\nu,V)}{k_{\mathrm{B}}T})]$$

Murnaghan Equation of State (EOS)

For a given $T = T_0$

$$F_{tot}(V,T_0) = U(V) + F_{vib}(V,T_0) = \frac{B_T V}{B'_T}\left[\left(\frac{V_0}{V}\right)^{B'_T}\frac{1}{B'_T-1}+1\right] + const.$$





Isotropic systems

For each $T = T_0$, two steps are done:

1. Fit the free energy points (calculated from U and the phonon frequencies for each volume) using Murnaghan EOS
2. Find the minimum of the fitted EOS

Some quantities are immediately obtained from the above two steps:

- **Equilibrium Helmoltz energy** $F_0(V_0, T_0)$
- **Equilibrium volume** $V_0(T_0)$
- **Isothermal bulk modulus** $B_T(T_0)$

Repeating the above two-steps procedure for each temperature from 0 to T_{max} K, the temperature dependence of F_0, V_0 and B_T is obtained. Remember that by definition:
$B_T = -V \left(\frac{\partial P}{\partial V}\right)_T = \frac{1}{K_T}$
Other quantities are obtained from the above quantities and their definitions or textbook thermodynamic equations.

Other quasi-harmonic quantities

The **volume thermal expansion** β is obtained as numerical derivative of $V_0(T)$ from the definition:

$$\beta(T) = \frac{1}{V_0(T)} \left(\frac{\partial V_0}{\partial T} \right)_P$$

The **isocoric heat capacity** is obtained directly from the phonon frequencies calculated at the equilibrium volume V_0:

$$C_V = k_B \sum_{\vec{q},\nu} \left(\frac{\hbar\omega(\vec{q},\nu)}{k_B T} \right)^2 \frac{\exp(\hbar\omega(\vec{q},\nu)/k_B T)}{\left[\exp(\hbar\omega(\vec{q},\nu)/k_B T) - 1\right]^2}$$

The **isobaric heat capacity** is obtained using the following thermodynamic equation:

$$C_P = C_V + TV\beta^2 B_T$$

Grüneisein parameters

The **Grüneisein mode parameters** are obtained as numerical derivative from the definition:

$$\gamma_{\mathbf{q},\nu,V} = -\frac{V}{\omega_{\mathbf{q},\nu,V}} \frac{\partial \omega_{\mathbf{q},\nu,V}}{\partial V}$$

The **Grüneisein (thermodynamic) parameter** is:

$$\gamma = \frac{V\beta B_T}{C_V} = \frac{V\beta B_S}{C_P}$$

but also

$$\gamma = \frac{V}{C_V}\left(\frac{\partial P}{\partial T}\right)_V = V\left(\frac{\partial T}{\partial U}\right)_V\left(\frac{\partial P}{\partial T}\right)_V = V\left(\frac{\partial P}{\partial U}\right)_V$$

Note: $B_S = -V\left(\frac{\partial P}{\partial V}\right)_S$ is the adiabatic bulk modulus

Software tools: thermo_pw/python codes

- Thermo_pw
  - in Fortran90
  - automatic calculations of several properties (convergence tests, phonon frequencies, electronic bands, elastic constants, QHA properties, etc.)
  - directly interfaced with QE (with one additional level of parallelization: images)
  - open source, GNU Licence, available at http://qeforge.qe-forge.org/gf/project/thermo_pw/
- pyqha Python module
  - in python 2/3, using numpy, scipy, matplotlib libraries
  - postprocessing of QHA properties from phonon DOS
  - code independent and easy to use and tinker with
  - open source, MIT Licence, available at https://github.com/mauropalumbo75/pyqha

Running thermo_pw

A typical command to run a thermo_pw calculation is:

mpirun -n np thermo_pw.x -ni ni ... < pw.in > outputfile

-n np → define the number of processor to be used
-ni ni → define the number of images to be used
... → parallelization options as in quantum espresso

3 input files:

- **pw.in** (necessary), the input file as for pw.x
- **ph_control** (optional), the input file as for ph.x
- **thermo_control** (necessary), the input file for thermo_pw

General output file in **outputfile**

Thermo_pw output files

Directory structure with thermo_pw results:

```
\anhar_files
\dynamical_matrices
\elastic_constants
\energy_files
\g1
\g2
\g3
\g4
\g5
\g6
\g7
\gnuplot_files
\phdisp_files
\restart
\therm_files

pw.in
ph_control
thermo_control
Si.pz-vbc.UPF
...
```

Running pyqha

As a Python module, **pyqha can be imported in your own scripts** as:

```
>>> import pyqha
```

or you can just import the functions you need as:

```
>>> from pyqha import read_thermo, gen_TT, compute_thermo_geo, rearrange_thermo
>>> from pyqha import fitFvibV, write_xy, simple_plot_xy, multiple_plot_xy
```

You can also use **pyqha** interactively in your Python shell.

The QHA for f.c.c. Si: pw.x input file

```
 &control
   calculation = 'scf'
   prefix='silicon',
   pseudo_dir = './',
   outdir='./'
 /
 &system
   ibrav= 2,
   celldm(1)= 10.20,
   nat=2,
   ntyp=1,
   ecutwfc=24.0,
 /
 &electrons
    conv_thr =  1.0d-8
 /
ATOMIC_SPECIES
 Si  28.086  Si.pz-vbc.UPF
ATOMIC_POSITIONS (alat)
 Si 0.00 0.00 0.00
 Si 0.25 0.25 0.25
K_POINTS AUTOMATIC
2 2 2 1 1 1
```

The QHA for f.c.c. Si: ph.x input file

```
 &inputph
  tr2_ph=1.0d-12,
  prefix='silicon',
  fildyn='si.dyn.xml',
  ldisp=.TRUE.
  nq1=4, nq2=4, nq3=4,
/
```

The QHA for f.c.c. Si: thermo_pw input file

```
&INPUT_THERMO
 what='mur_lc_t',
 lmurn=TRUE

 nq1_d=128, nq2_d=128, nq3_d=128,

 ngeo=7
 step_ngeo=0.5

 tmin=1
 tmax=800
 deltat=3.
/
```

Units

Units used in Quantum Espresso:

Energy: Rydberg, symbol **Ry**
Distance: atomic unit, symbol **a.u.**

Useful conversions:

1 Ry $\equiv hcR_\infty$ = 13.60569253 eV = 2.1798720294$\cdot 10^{-18}$ J
1 eV/at = 96.4853365 kJ/mol
1 a.u. (Bohr radius) = 0.052918 nm = 0.52918 Å