G. Bellomia
J. Lapajne
N. Ranabhat

Solid Argon Vibrations

2020/01/22

A Lennard-Jones Model for Solid Argon

In the present work we want to present a simple treatment of static and dynamic lattice properties for solid Argon. In its rare-gas phase Argon is typically modeled as a 2-body Lennard Jones interaction model. Regarding the solid-state phase this model leads to some well-known inaccuracies in the description (see e.g. Ref. [2]). Nevertheless we will show that at least the vibrational properties are reproduced quite satisfactorily within the harmonic approximation.

Then, starting from the parametric form of the LJ potential

$$V_{\mathrm{LJ}}(r) = 4\varepsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right],$$

we fit ε and σ imposing the experimental values for the equilibrium distance r_0 and binding energy BE of an Argon dimer:

$$\begin{gathered} 3.758\,\text{Å} = r_0 \equiv r \ \ \textit{such that} \ \ \frac{\mathrm{d}V_{\mathrm{LJ}}(r)}{\mathrm{d}r} = 0 \\ 99.55\,\mathrm{cm}^{-1} = \mathrm{BE} \equiv V_{\mathrm{LJ}}(\infty) - V_{\mathrm{LJ}}(r_0) \\ \Downarrow \\ \varepsilon = \mathrm{BE} = 99.55\,\mathrm{cm}^{-1}, \quad \sigma = 2^{-1/6}\, r_0 = 3.348\,\text{Å}. \end{gathered}$$

The resulting potential curve is shown and briefly discussed in Figure 1.

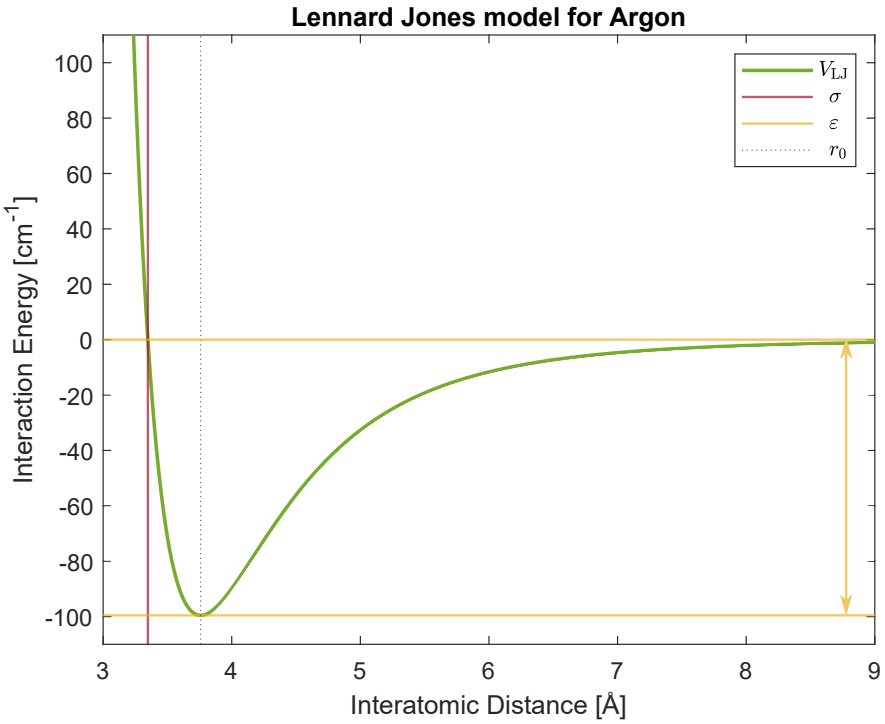
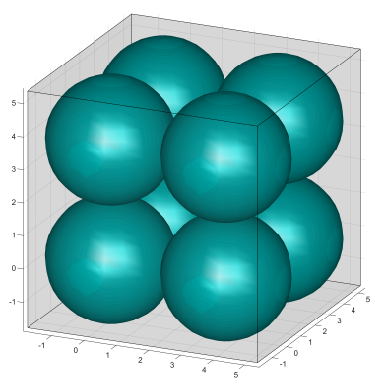


Figure 1: Experimental-data fitted LJ potential for an Argon dimer. The σ parameter can be regarded as the minimum distance that the system can visit without unbinding (*hardcore radius*), ε corresponding to the experimental binding energy.

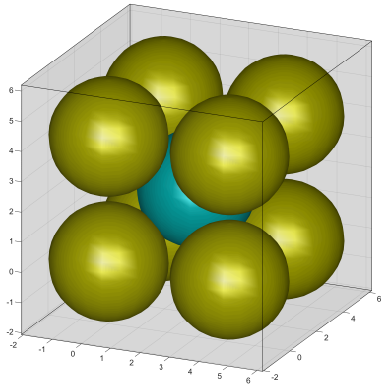
Structural Optimization

The first thing one should investigate, before addressing any dynamical property of a solid material, is what lattice structure it crystallizes into, given the relevant external parameters. For zero temperature and pressure this amounts to determine which structure has lower energy. This is the issue we will address in this section, comparing between simple (SC), body centered (BCC) and face centerd (FCC) cubic lattices and hexagonal close packed (HCP) structure. For their unit- and conventional- cell definitions and for the related notations we follow Ref. [1].

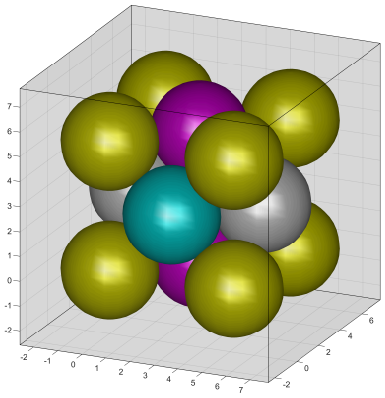
In order to determine which crystal structure for the solid phase of Argon is energetically favored we perform a constrained geometry optimization: we first fix all the periodic structure in terms of one lattice parameter and then minimize the whole *many-body* cohesive energy with respect to it. A more careful approach would be to introduce more variational parameters, e.g. to define three independent lattice parameters a, b and c for the three directions given by the lattice basis vectors or even some angular parameter to allow for deformations of the crystal structure. But since we are interested in bulk properties and being the inter-ionic potential spherically symmetric we expect the optimal structure to be a close packed one, so even for the HCP structure (the only anisotropic one) we fix $c = \sqrt{8/3}\,a$, thus imposing a strict *close-packing* condition.[1]



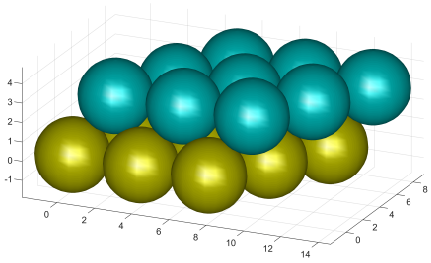
(a) Simple Cubic [Conv. Cell]



(b) Body Centered Cubic [Conv. Cell]



(c) Face Centered Cubic [Conv. Cell]



(d) Hexagonal Close Packed [Arb. View]

Figure 2: The four crystal structures considered for geometry optimization.

To evaluate the bulk cohesion energy one should in principle sum an infinite series of 2-body interaction terms, which can be a difficult task to handle. But since the Lennard-Jones interaction is quite short ranged a truncation of the series should give a controlled result. So we have generated for each crystal structure several supercells, progressively increasing in size, and for each one computed the interaction of the *central atom*[2] with all the others. The resulting thermodynamic limit extrapolation will give an estimate for twice the cohesive energy, which we claim to correspond to a full (double-sum) PBC (*minimum image convention*) calculation for a supercell with double linear dimensions that the ones used in our method: we expect the convergence to the thermodynamic limit to be fast, with good accuracy in the resulting energies values.

[1] For all rare-gas solids the optimal value for the HCP c-parameter should differ from this ideal one by less than 0.1%. (cf. Ref. [2])
[2] We have generated the supercell spanning integer multiples of the unit vectors around it. (cf. Listing 1)

In principle we could not assure the commutation of the thermodynamic limit with the minimization procedure, so we have been careful with a double-check procedure:

1. First we have generated a set of fairly but not exceedingly large supercells (each one made of approximately 2×10^3 atoms), whose structures[3] are reported in Figures 3, 4, 5, 6, respectively for the SC, BCC, FCC and HCP structures. The generation has been done for a handful of lattice parameter values, around the *naively-expected* optimal value (i.e. the one ensuring the nearest neighbors distance to correspond to the isolated dimer equilibrium distance r_0).
2. Then we have minimized the resulting cohesive energy by mean of a polynomial fit of the values computed for the given lattice parameters (see Figure 7). The polynomial has been chosen to be cubic to better fit the evident asymmetry of the computed data around the minimum.
3. Fixing the obtained optimal values for a we have performed a full-scale thermodynamic limit study (up to more than 10^5 atoms), for increasing supercells with the full symmetry of the conventional cells, obtaining a set of explicit convergence curves. (see Figure 9)
4. Finally we have repeated the minimization procedure as described in point 2, but for a supercell size (approximately 5×10^5 atoms) that we consider to be *fully-converged* for what concerns a clear separation of the energies for the different structures. (see Figure 8)

Results and Discussion

Here we present and compare all our results and the theoretical values we have computed following Ref. [2].[4]

	SC	BCC	FCC	HCP
"Small" Supercell				
a^{eq} [Å]	3.573362	4.131401	5.162758	3.650219
U^{eq} [cm^{-1}]	-564.8269	-817.6577	-855.4966	-856.2407
"Large" Supercell				
a^{eq} [Å]	3.572520	4.130540	5.161900	3.649820
U^{eq} [cm^{-1}]	-566.5532	-820.0268	-857.1429	-857.2445
Reference Theory				
a^{eq} [Å]	3.572597	4.130489	5.161733	3.649876
U^{eq} [cm^{-1}]	-566.5348	-820.0224	-857.1454	-857.2320

From the values reported in the table is evident that the bulk equilibrium lattice parameter determination is very robust even for arguably small systems. Instead the corresponding cohesion energy values are significantly more sensitive to supercell dimension, as we could expect since the Lennard Jones contains high powers of the inter-ionic distances.
Actually, inspecting carefully Figure 9, we can see how the difference between FCC and HCP cohesion energies should have even incorrect sign before $\sim 2 \times 10^4$ atoms are accounted for in the calculation. This does not happen in our "small" supercell calculation, probably due to some fortuitous anisotropy effect (the choice of which axis to double the repetition on might lead to different results), sign that $\sim 10^3$ atoms cannot be considered a true bulk for the purpose of discriminating such small energy differences.

[3] These structures may appear somewhat inconsistent: the SC and FCC supercells fully reflect the symmetry of the lattice, while the BCC and the HCP ones introduce some macroscopic anisotropy, being the number of repetition on one of the axes doubled with respect to the others. The reason for this choice is to assure the calculation on different structures to be performed on (almost) the same number of atoms, so to have a meaningful comparison. We claim this anisotropy to be irrelevant for a large enough supercell (i.e. a true bulk).

[4] The reference gives some simple formulas for the equilibrium lattice parameter and the consequent cohesion energy value in terms of:

- Two crystal-structure dependent parameters, L_6 and L_{12}, for which we have assumed the values given there.
- Two inter-ionic potential dependent parameters, corresponding to binding energy and equilibrium distance of a rare-gas dimer. For these we have used the same experimental values we used for all the numerical calculations.

Compare Ref. [2] and Listing 5 for further details.

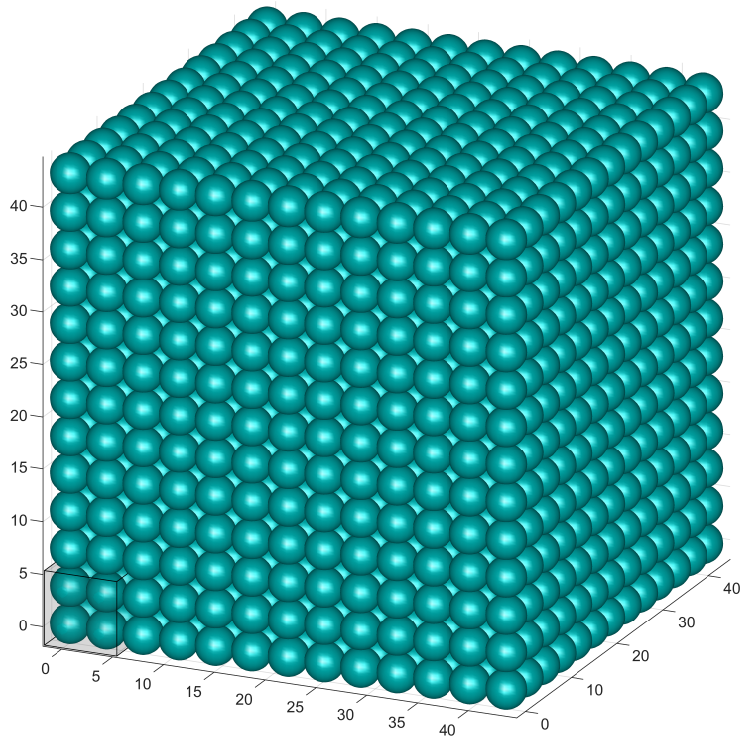


Figure 3: Preliminary supercell for SC optimization. The total number of atoms is $N = 2197$.

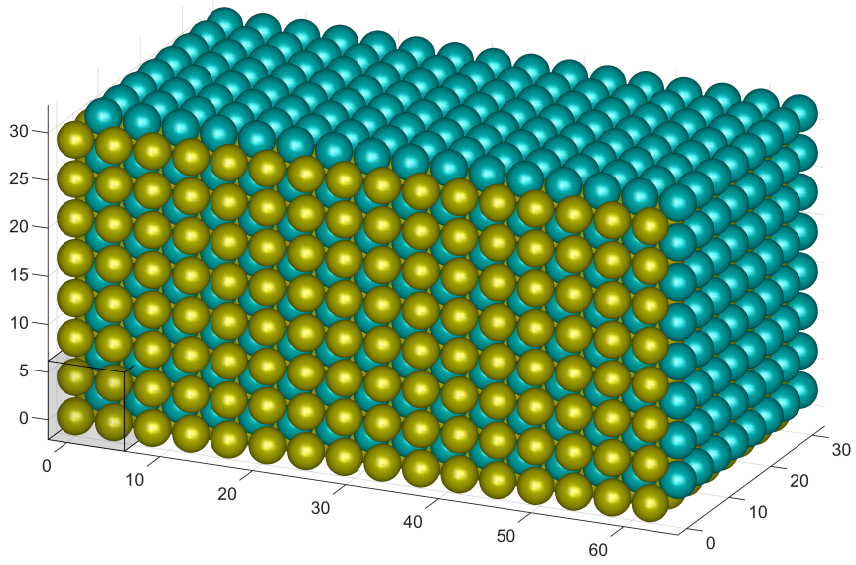


Figure 4: Preliminary supercell for BCC optimization. The total number of atoms is $N = 2048$.

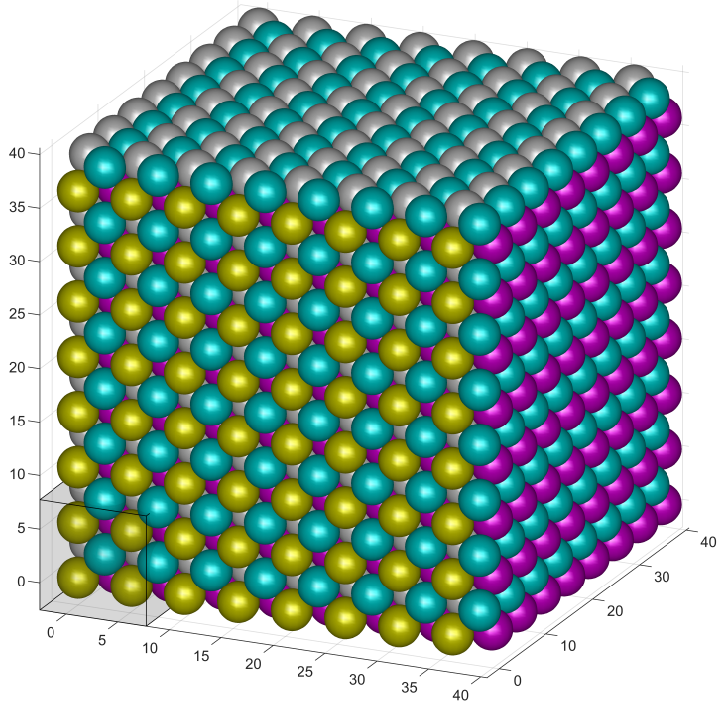


Figure 5: Preliminary supercell for FCC optimization. The total number of atoms is $N = 2048$.

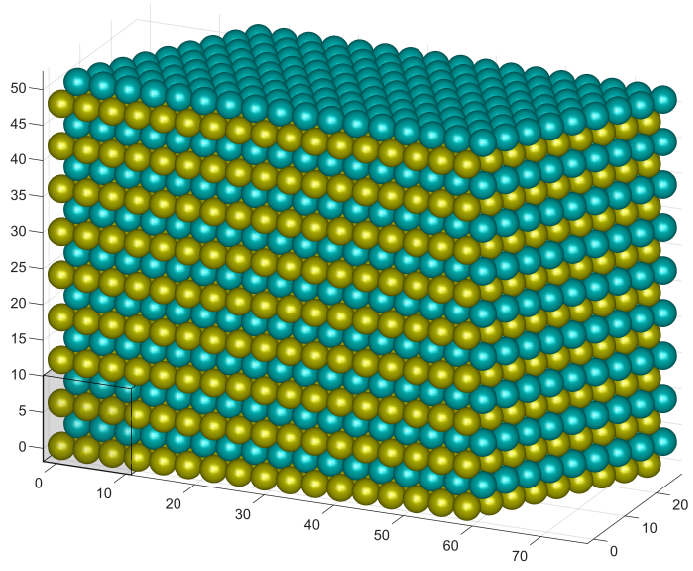


Figure 6: Preliminary supercell for HCP optimization. The total number of atoms is $N = 2048$.

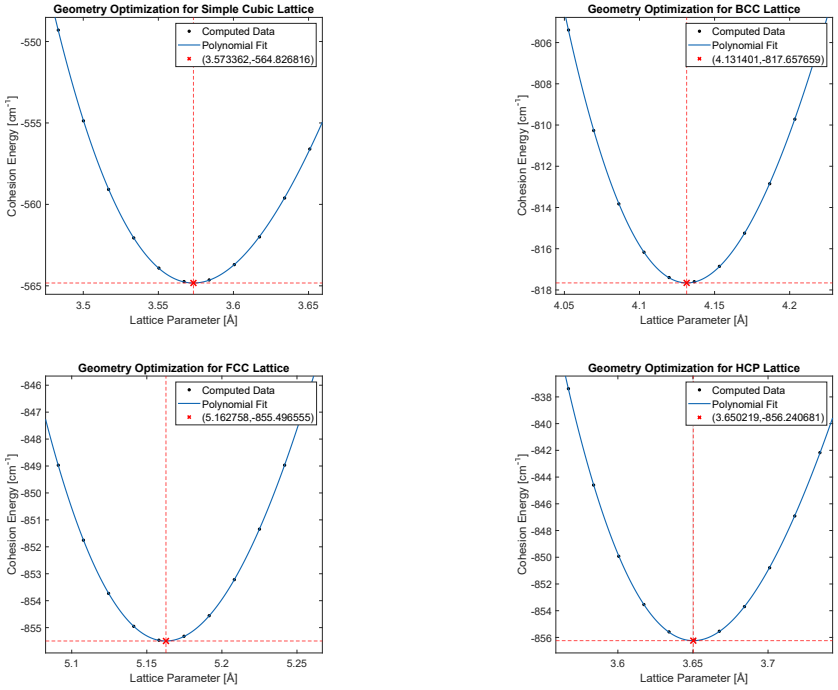


Figure 7: Determination of optimal geometries at $\sim 2 \times 10^3$ atoms.

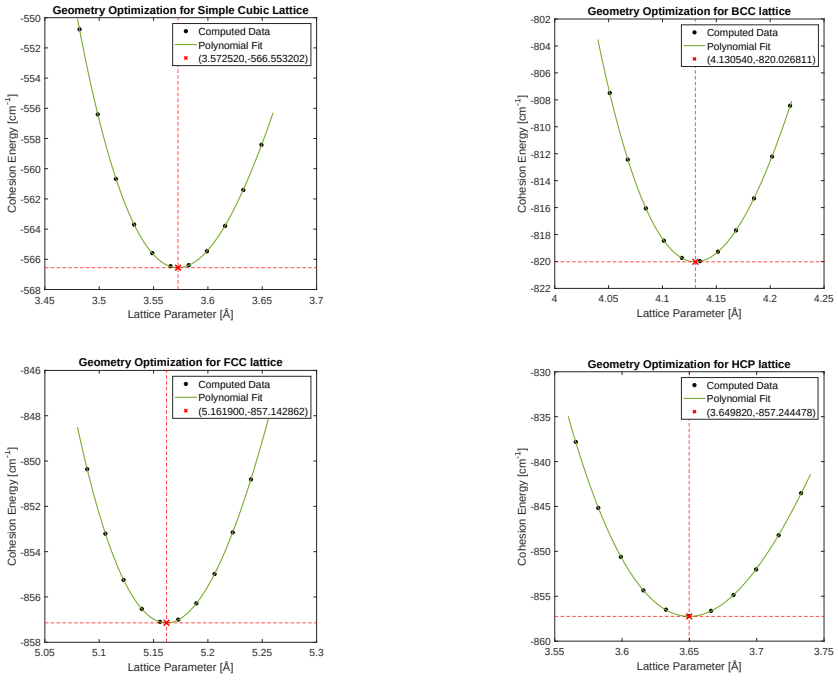


Figure 8: Determination of optimal geometries at $\sim 5 \times 10^5$ atoms.

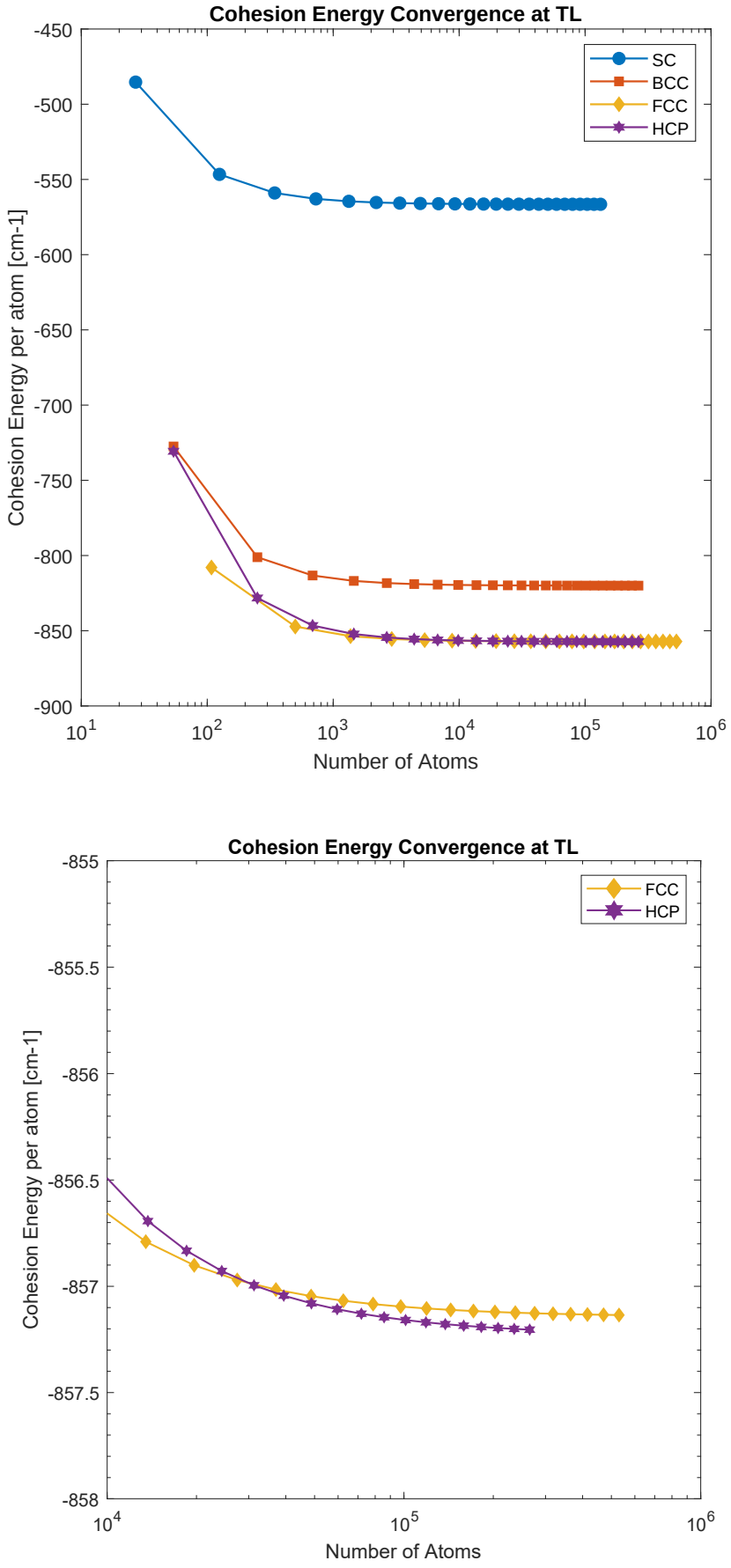


Figure 9: Thermodynamic limit convergence curves for the cohesion energies of the four crystal structures considered for geometry optimization (Upper Panel). Detail of the crossing between the FCC and the HCP convergence curves (Lower Panel). All the calculation are performed for “isotropic” supercells: the number of repetition is equal for all axes.

Phonon Dispersions

In the following we will present phonon dispersion relations for Hexagonal Close Packed (HCP) and Face Centered Cubic (FCC) structures, as computed within nearest neighbors harmonic approximation. Hence we will revise here the general dynamical matrix formalism and the implementation details for our specific lattices. We will follow the conventions and notations of Ref. [1].

Harmonic approximation

Harmonic approximation is the simplest upgrade one can implement to the static study of crystals cohesion energy, in the sense that it allows for the motion of ions in the lattice within the first nonzero term of the Taylor expansion of the total lattice energy, around the equilibrium position.

Hence the two underlying assumptions are:

1. The equilibrium positions of ions are fixed to the Bravais lattice sites (+ a fixed basis displacement)[5] and the instantaneous position of the ion can be written as
$$\vec{r} = \vec{R} + (\,\vec{s} +\,)\,\vec{u}(\vec{R})$$
where $\vec{r}$ is the instantaneous position, $\vec{R}(+\vec{s})$ is the equilibrium position, and $\vec{u}(\vec{R})$ is the instantaneous position of ion with respect to $\vec{R}(+\vec{s})$.
2. The the instantaneous position of ion with respect to the equilibrium geometry is always small compared to the inter-ionic spacing.

This two assumption allows us to write the total potential energy of the system as:[6]

$$U = \sum_{\text{all bonds}} \phi(\vec{r}) \simeq U^{\text{eq}} + U^{\text{harm}}$$

where

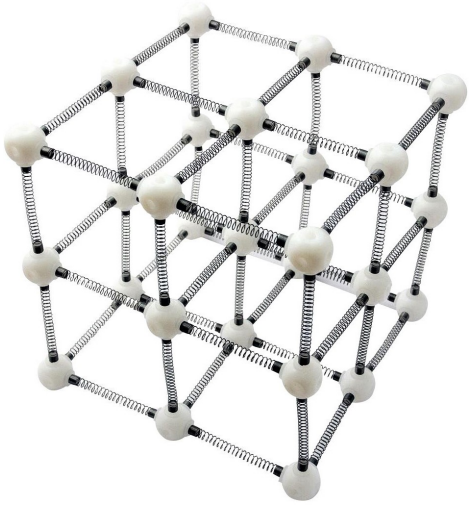
$$U^{\text{harm}} = \frac{1}{2} \sum_{\vec{R}\vec{r'},\mu\nu} u_\mu(\vec{R}) D_{\mu\nu}(\vec{R} - \vec{r'}) u_\nu(\vec{r'}),$$

$$D_{\mu\nu}(\vec{R} - \vec{r'}) = \delta_{\vec{R},\vec{r'}} \sum_{\vec{r''}} \phi_{\mu\nu}(\vec{R} - \vec{r''}) - \phi_{\mu\nu}(\vec{R} - \vec{r'}),$$

and

$$\phi_{\mu\nu}(\vec{R}) = \frac{\partial^2 \phi(\vec{R})}{\partial R_\mu \partial R_\nu}$$

with μ and ν being Cartesian indices for the axes.



[5] We will postpone for a moment the full discussion of lattices with a basis.

[6] In general the 2-body potential $\phi(\vec{r})$ describes a pairwise interaction of the ions, so in our model for solid Argon it coincides with the experimentally-fitted $V_{\text{LJ}}(r)$.

Equations of motion and dynamical matrix

For a classical system of N particles in three dimensions there are $3N$ equation of motions, given by

$$M\frac{\partial^2 u_\mu(\vec{R})}{\partial t^2} = -\frac{\mathrm{d}U^{\mathrm{harm}}}{\mathrm{d}u_\mu(\vec{R})} = -\sum_{\vec{r'},\nu} D_{\mu\nu}(\vec{R}-\vec{r'})u_\nu(\vec{r'}).$$

We seek solutions of the form

$$u_\nu(\vec{R},t) = \epsilon_\nu e^{i(\vec{k}\cdot\vec{R}-\omega t)},$$

where ω is frequency. Hence we get

$$M\omega^2\epsilon_\mu = D_{\mu\nu}(\vec{k})\epsilon_\nu,$$

which in matrix form reads:

$$M\omega^2\epsilon = D(\vec{k})\epsilon,$$

where

$$D(\vec{k}) = \sum_{\vec{R}} D(\vec{R})e^{i\vec{k}\cdot\vec{R}}$$

is known as the *Dynamical Matrix* of the harmonic system.

Clearly the eigenvalues of the Dynamical matrix $D(\vec{k})$ give the vibrational dispersion relation as

$$\omega_\nu(\vec{k}) = \sqrt{\frac{\lambda_\nu(\vec{k})}{M}},$$

where the index ν labels the different eigenvalues at a fixed wavevector $\vec{k}$ (normal modes of the solid).

So the problem of computing the dispersion relation of a crystal involves two steps: a) determining the form of the dynamical matrix and b) diagonalizing it for the appropriate $\vec{k}$ values.

Lattice with a basis

Before we start the actual calculation for the crystals of interest, let us generalize the expression of dynamical matrix for a Bravais lattice with basis. In this case the real space dynamical matrix is written with extra indices, representing the basis labels:

$$D_{\mu\nu}(\vec{R}+\vec{s}-(\vec{r'}+\vec{s'})) = \delta_{\vec{s},\vec{s'}}\delta_{\vec{R},\vec{r'}}\sum_{\vec{r''},\vec{s''}}\phi_{\mu\nu}(\vec{R}+\vec{s}-(\vec{r''}+\vec{s''})) - \phi_{\mu\nu}(\vec{R}+\vec{s}-(\vec{r'}+\vec{s'})),$$

where $\vec{s}$, $\vec{s'}$ and $\vec{s''}$ are the indices for basis ions.

We can then get the reciprocal space dynamical matrix by Fourier transformation

$$\begin{aligned}
D_{\mu\nu;\vec{s},\vec{s'}}(\vec{k}-\vec{k'}) &= \sum_{\vec{R}}\sum_{\vec{r'}} e^{i\vec{k}\cdot\vec{R}}e^{i\vec{k'}\cdot\vec{r'}}D_{\mu\nu}(\vec{R}+\vec{s}-(\vec{r'}+\vec{s'})) \\
&= \sum_{\vec{R}}\sum_{\vec{r'}} e^{i\vec{k}\cdot\vec{R}}e^{i\vec{k'}\cdot\vec{r'}}\left[\delta_{\vec{s},\vec{s'}}\delta_{\vec{R},\vec{r'}}\sum_{\vec{r''},\vec{s''}}\phi_{\mu\nu}(\vec{R}+\vec{s}-(\vec{r''}+\vec{s''})) - \phi_{\mu\nu}(\vec{R}+\vec{s}-(\vec{r'}+\vec{s'}))\right] \\
&= \sum_{\vec{R}} e^{i(\vec{k}+\vec{k'})\cdot\vec{R}}\delta_{\vec{s}\vec{s'}}\sum_{\vec{r''},\vec{s''}}\phi_{\mu\nu}(\vec{R}+\vec{s}-(\vec{r''}+\vec{s''})) - \sum_{\vec{R}\vec{r'}} e^{i\vec{k}\cdot\vec{R}}e^{i\vec{k'}\cdot\vec{r'}}\phi_{\mu\nu}(\vec{R}+\vec{s}-(\vec{r'}+\vec{s'})) \\
&= \delta_{\vec{k}+\vec{k'}}\delta_{\vec{s}\vec{s'}}\sum_{\vec{r},\vec{s''}}\phi_{\mu\nu}(\vec{r}+\vec{s}-\vec{s''}) - \sum_{\vec{R}\vec{r'}} e^{i\vec{k}\cdot\vec{R}}e^{i\vec{k'}\cdot(\vec{R}-\vec{r'})}\phi_{\mu\nu}(\vec{r'}+\vec{s}-\vec{s'}) \\
&= \delta_{\vec{k}+\vec{k'}}\delta_{\vec{s}\vec{s'}}\sum_{\vec{r},\vec{s''}}\phi_{\mu\nu}(\vec{r}+\vec{s}-\vec{s''}) - \delta_{\vec{k}+\vec{k'}}\sum_{\vec{r'}} e^{-i\vec{k'}\cdot\vec{r'}}\phi_{\mu\nu}(\vec{r'}+\vec{s}-\vec{s'})
\end{aligned}$$

so the dynamical matrix in reciprocal space is

$$D_{\mu\nu;\vec{s},\vec{s'}}(\vec{k}) = \sum_{\vec{r}}\delta_{\vec{s}\vec{s'}}\sum_{\vec{s''}}\phi_{\mu\nu}(\vec{r}+\vec{s}-\vec{s''}) - \sum_{\vec{r'}} e^{i\vec{k}\cdot\vec{r}}\phi_{\mu\nu}(\vec{r}+\vec{s}-\vec{s'})$$

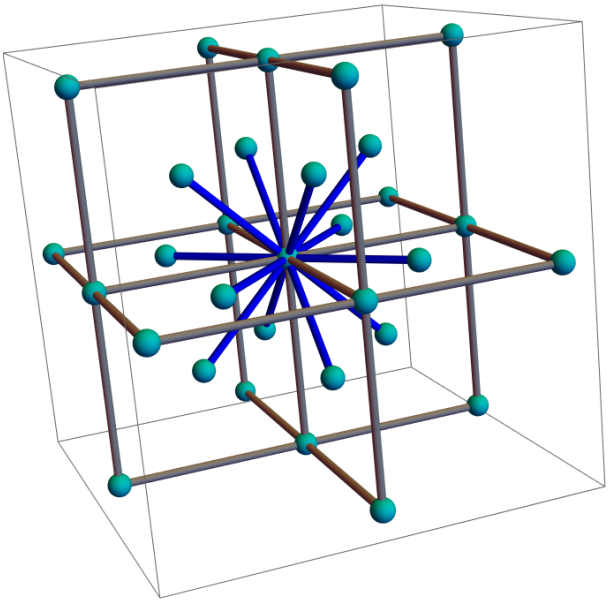


Figure 10: The 12 nearest neighbors of a representative FCC lattice site are connected by blue-colored bonds.

Dynamical matrix for FCC

In FCC there is only one basis ion per lattice point so the dynamical matrix reduces to

$$\begin{aligned}
D_{\mu\nu}(\vec{k}) &= \sum_{\vec{r}} \phi_{\mu\nu}(\vec{r}) \left(1 - e^{i\vec{k}\cdot\vec{r}}\right) \\
&= \frac{1}{2} \sum_{\vec{r}} \phi_{\mu\nu}(\vec{r}) \left(2 - 2e^{i\vec{k}\cdot\vec{r}}\right) \\
&= \frac{1}{2} \sum_{\vec{r}} \phi_{\mu\nu}(\vec{r}) \left(2 - e^{i\vec{k}\cdot\vec{r}} - e^{-i\vec{k}\cdot\vec{r}}\right) \\
&= \sum_{\vec{r}} \phi_{\mu\nu}(\vec{r}) \left(1 - \cos\left(\vec{k}\cdot\vec{r}\right)\right) \\
&= \sum_{\vec{r}} 2\phi_{\mu\nu}(\vec{r}) \sin^2\left(\frac{\vec{k}\cdot\vec{R}}{2}\right),
\end{aligned}$$

where for FCC the sum is extended to the 12 nearest neighbor of a representative lattice point, as shown in Figure 10. The relevant vectors are then:

$$\tfrac{a}{2}(\pm\hat{x}, \pm\hat{y}), \quad \tfrac{a}{2}(\pm\hat{y}, \pm\hat{z}), \quad \tfrac{a}{2}(\pm\hat{z}, \pm\hat{x}),$$

where the $\pm$ signs are meant to prescribe to take all the possible combinations.

The tensor $\phi_{\mu\nu}(\vec{r})$ can be easily evaluated for 2-body potentials, where the interaction between two ions is dependent only in the distance between them, as:

$$\begin{aligned}
\phi_{\mu\nu}(\vec{r}) &= \frac{\partial^2 \phi(\vec{r})}{\partial r_\mu \partial r_\nu} \\
&= \frac{\partial \left(\frac{\partial \phi(r)}{\partial r} \frac{\partial r}{\partial r_\nu}\right)}{\partial r_\mu} \\
&= \frac{\partial \left(\phi'(r) \frac{r_\nu}{r}\right)}{\partial r_\mu} \\
&= \frac{\phi'(r)}{r}\delta_{\mu\nu} + \frac{\phi''(r)}{r^2} r_\mu r_\nu - \frac{\phi'(r)}{r^3} r_\mu r_\nu
\end{aligned}$$

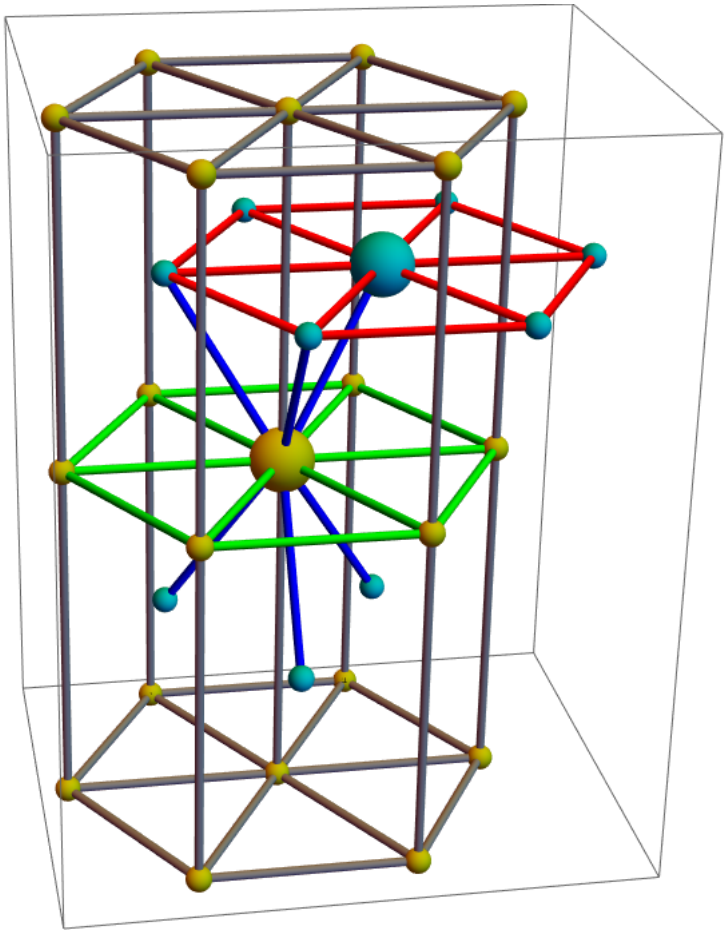


Figure 11: The nearest neighbor scheme for HCP lattice. The A and B sublattices are colored in green-yellow and blue-green respectively. The nearest neighbors AA, BB, and AB/BA interactions are represented respectively by light-green, red, and blue bonds.

Dynamical matrix for HCP

The hexagonal close packed structure is a Bravais lattice with basis with primitive lattice vectors

$$\vec{a}_1 = a\hat{x}, \quad \vec{a}_2 = \frac{a}{2}\hat{x} + \frac{\sqrt{3}}{2}\hat{y}, \quad \vec{a}_3 = c\hat{z},$$

with the two basis ions located at

$$\vec{s}_A = (0,0,0), \quad \vec{s}_B = (\tfrac{a}{2}, \tfrac{a}{2\sqrt{3}}, \tfrac{c}{2}).$$

The dynamical matrix in reciprocal space is then

$$D_{\mu\nu;\vec{s},\vec{s'}}(\vec{k}) = \sum_{\vec{r}} \delta_{\vec{s}\vec{s'}} \sum_{\vec{s''}} \phi_{\mu\nu}(\vec{r} + \vec{s} - \vec{s''}) - \sum_{\vec{r'}} e^{i\vec{k}\cdot\vec{r'}} \phi_{\mu\nu}(\vec{r'} + \vec{s} - \vec{s'}),$$

where both the summations are understood to be over all the nearest neighbors. But since in this case we have two basis ions, we have to clearly define the nearest neighbors for all the four different kinds of interaction. These are highlighted explicitly in Figure 11.

Let us thus calculate the dynamical matrix term by term:

$$\begin{aligned}
D_{\mu\nu;\vec{s}_A,\vec{s'}_A}(\vec{k}) &= \sum_{\vec{r}}\sum_{\vec{s''}}\phi_{\mu\nu}(\vec{r}+\vec{s}_A-\vec{s''}) - \sum_{\vec{r'}} e^{i\vec{k}\cdot\vec{r'}}\phi_{\mu\nu}(\vec{r'}) \\
&= \sum_{\vec{r}}\left[\phi_{\mu\nu}(\vec{r})+\phi_{\mu\nu}(\vec{r}-\vec{s}_B)\right] - \sum_{\vec{r'}} e^{i\vec{k}\cdot\vec{r'}}\phi_{\mu\nu}(\vec{r'}) \\
&= \sum_{\vec{r}}\phi_{\mu\nu}(\vec{r})(1-e^{i\vec{k}\cdot\vec{r}}) + \sum_{\vec{r'}}\phi_{\mu\nu}(\vec{r'}-\vec{s}_B) \\
&= \sum_{\vec{r}} 2\phi_{\mu\nu}\sin^2\left(\frac{\vec{k}\cdot\vec{R}}{2}\right) + \sum_{\vec{r'}}\phi_{\mu\nu}(\vec{r'}-\vec{s}_B),
\end{aligned}$$

$$D_{\mu\nu;\vec{s}_A,\vec{s'}_B}(\vec{k}) = \sum_{\vec{r'}} -e^{i\vec{k}\cdot\vec{r'}}\phi_{\mu\nu}(\vec{r'}-\vec{s}_B),$$

$$\begin{aligned}
D_{\mu\nu;\vec{s}_B,\vec{s'}_B}(\vec{k}) &= \sum_{\vec{r}}\sum_{\vec{s''}}\phi_{\mu\nu}(\vec{r}+\vec{s}_B-\vec{s''}) - \sum_{\vec{r'}} e^{i\vec{k}\cdot\vec{r'}}\phi_{\mu\nu}(\vec{r'}) \\
&= \sum_{\vec{r}}\left[\phi_{\mu\nu}(\vec{r})+\phi_{\mu\nu}(\vec{r}+\vec{s}_B)\right] - \sum_{\vec{r'}} e^{i\vec{k}\cdot\vec{r'}}\phi_{\mu\nu}(\vec{r'}) \\
&= \sum_{\vec{r}}\phi_{\mu\nu}(\vec{r})(1-e^{i\vec{k}\cdot\vec{r}}) + \sum_{\vec{r'}}\phi_{\mu\nu}(\vec{r'}+\vec{s}_B) \\
&= \sum_{\vec{r}}\phi_{\mu\nu}(\vec{r})(1-e^{i\vec{k}\cdot\vec{r}}) + \sum_{\vec{r'}}\phi_{\mu\nu}(\vec{r'}-\vec{s}_B) \\
&= \sum_{\vec{r}} 2\phi_{\mu\nu}\sin^2\left(\frac{\vec{k}\cdot\vec{R}}{2}\right) + \sum_{\vec{r'}}\phi_{\mu\nu}(\vec{r'}-\vec{s}_B),
\end{aligned}$$

$$D_{\mu\nu;\vec{s}_B,\vec{s'}_A}(\vec{k}) = \sum_{\vec{r''}} -e^{i\vec{k}\cdot\vec{r''}}\phi_{\mu\nu}(\vec{r''}+\vec{s}_B),$$

where the sums run through the following vectors:[7]

$$\vec{r} \in \left\{ \pm(a,0,0), \pm\left(\frac{a}{2},\frac{\sqrt{3}}{2},0\right), \pm\left(-\frac{a}{2},\frac{\sqrt{3}}{2},0\right) \right\}$$

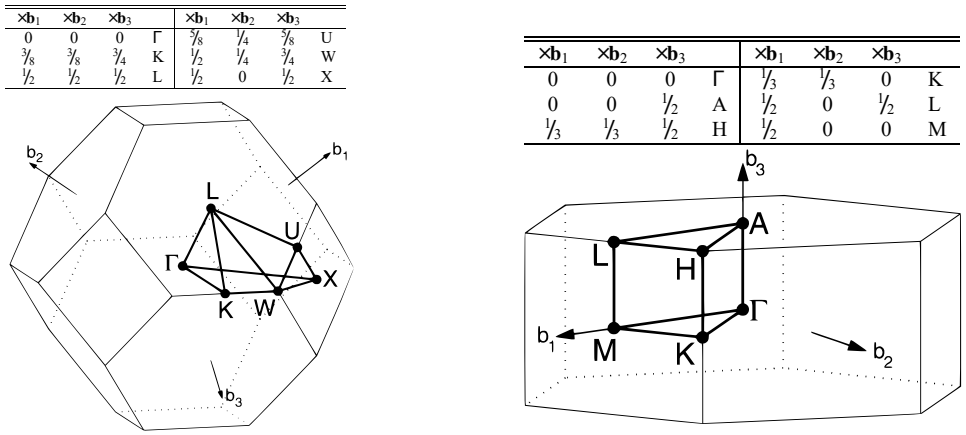
$$\vec{r'} \in \left\{ (0,0,0), (a,0,0), \left(\frac{a}{2},\frac{\sqrt{3}}{2},0\right), \left(0,0,\sqrt{\frac{8}{3}}a\right), \left(a,0,\sqrt{\frac{8}{3}}a\right), \left(\frac{a}{2},\frac{\sqrt{3}}{2},\sqrt{\frac{8}{3}}a\right), \right\}$$

$$\vec{r''} \in \left\{ (0,0,0), (-a,0,0), \left(-\frac{a}{2},-\frac{\sqrt{3}}{2},0\right), \left(0,0,-\sqrt{\frac{8}{3}}a\right), \left(-a,0,-\sqrt{\frac{8}{3}}a\right), \left(-\frac{a}{2},-\frac{\sqrt{3}}{2},-\sqrt{\frac{8}{3}}a\right) \right\}$$

[7]The $\pm$ signs are again meant to prescribe to take all the possible combinations.

Dispersions along High-Symmetry Lines

Having the FCC and HCP nearest neighbors dynamical matrices we proceed to define the set of k-vectors for which we want diagonalize the vibrational eigenproblem. In principle one should sample the whole Brillouin Zone (or more precisely its symmetry-induced *irreducible wedge*) but usually all the valuable information on frequency dispersion can be gathered by further restricting to *high-symmetry lines*, which we looked for in literature (Ref. [4]).



(a) FCC

(b) HCP

Figure 12: Brilluoin Zones and High-Symmetry Lines for FCC and HCP structures. Adapted from Ref. [4].

In the upper panels of Figures 12a & 12b the high-symmetry point coordinates are listed, expressed in the reciprocal unit-vector basis, i.e.

- For FCC

$$\vec{b}_1 = \frac{2\pi}{a}(\hat{k}_x - \hat{k}_y + \hat{k}_z), \quad \vec{b}_2 = \frac{2\pi}{a}(\hat{k}_x + \hat{k}_y - \hat{k}_z), \quad \vec{b}_3 = \frac{2\pi}{a}(-\hat{k}_x + \hat{k}_y + \hat{k}_z).$$

- For HCP

$$\vec{b}_1 = \frac{2\pi}{\sqrt{3}a}(\sqrt{3}\,\hat{k}_x - \hat{k}_y), \quad \vec{b}_2 = \frac{4\pi}{\sqrt{3}a}\,\hat{k}_y, \quad \vec{b}_3 = \frac{2\pi}{c}\,\hat{k}_z, \quad c = \sqrt{\frac{8}{3}}\,.$$

The phonon dispersions computed along some of the lines connecting these points are reported in Figure 13. The dispersion shapes match the nearest neighbors results reported in Ref. [5] and the total frequency range is the same for both structures, as we expect being the nearest neighbors distance (and so the harmonic springs) almost equal.

Comparison with Experiments

In Figures 14, 15 and 16 we compare our results with some literature data. The FCC is compared with an inelastic neutron scattering study of solid Argon (Ref. [6]), and the HCP with some measurements (Ref. [7]) on Berillium[8]. The latter comparison was indeed expected to be barely qualitative, since the inter-ion interaction in Be is not even comparable to a LJ system, due to the metallic nature of the chemical bonds. Yet the main features are clearly reproduced, suggesting that crystal symmetry dominates on the particular choice of the 2-body potential, in determining the qualitative shape of phonon dispersion. Instead for the FCC structure a quantitative comparison is readily done: in Figure 14 we have expressed the frequencies in meV – thus matching the reference data units – and defined a *non*high-symmetry path, roughly following the choice in reference. The maximum relative deviation in frequency values appears to be under 10%, which we found to be a striking match for a simple nearest neighbors calculation.

[8]We could not find any reference for HCP solid Argon, maybe due to the great experimental difficulties involved in its growth. (Ref. [3])

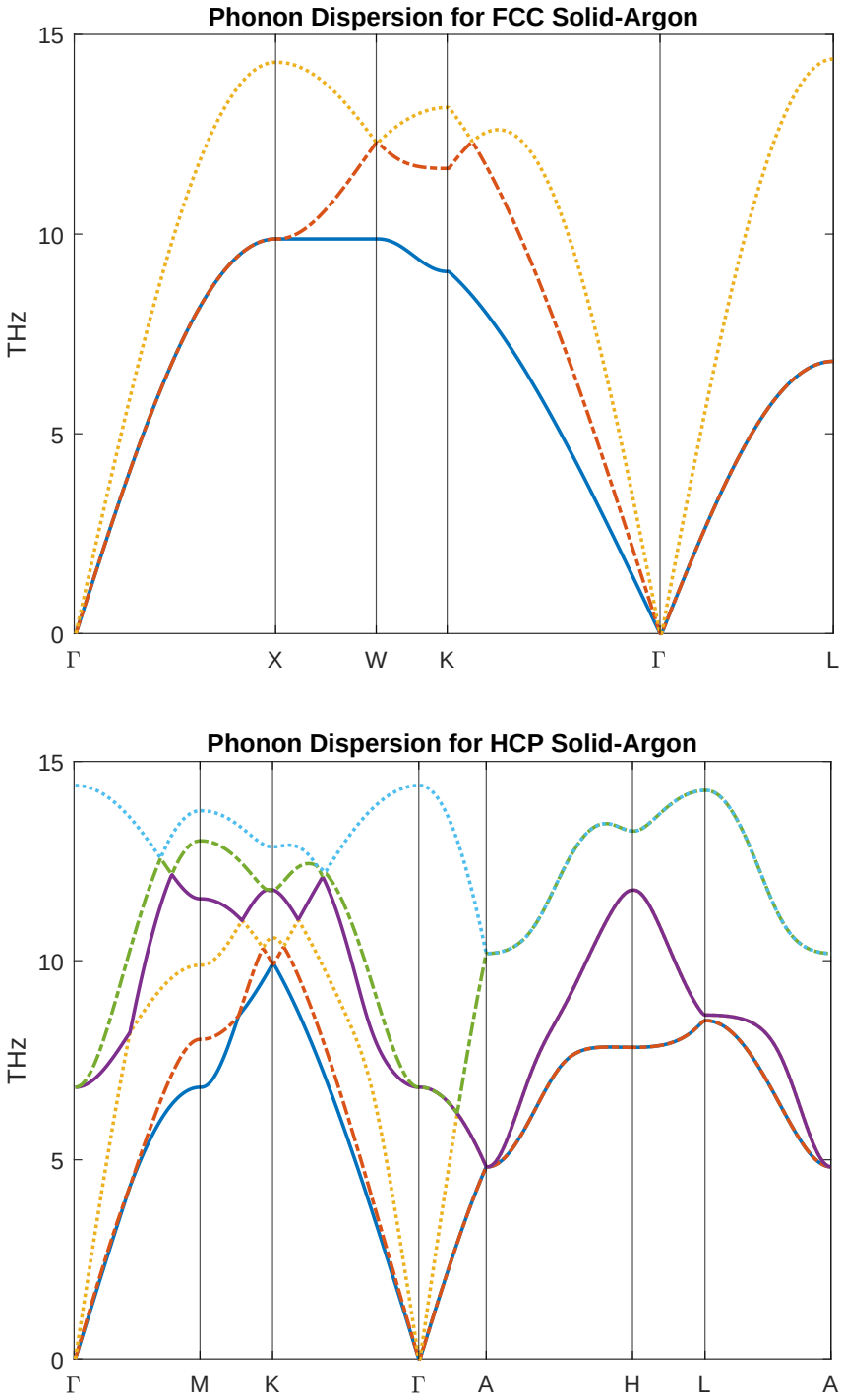


Figure 13: Computed phonon dispersions for the FCC (upper) and HCP (lower) Solid-Argon structures. The dynamical matrices include n.n. interactions only and the paths are those suggested in Ref. [4]

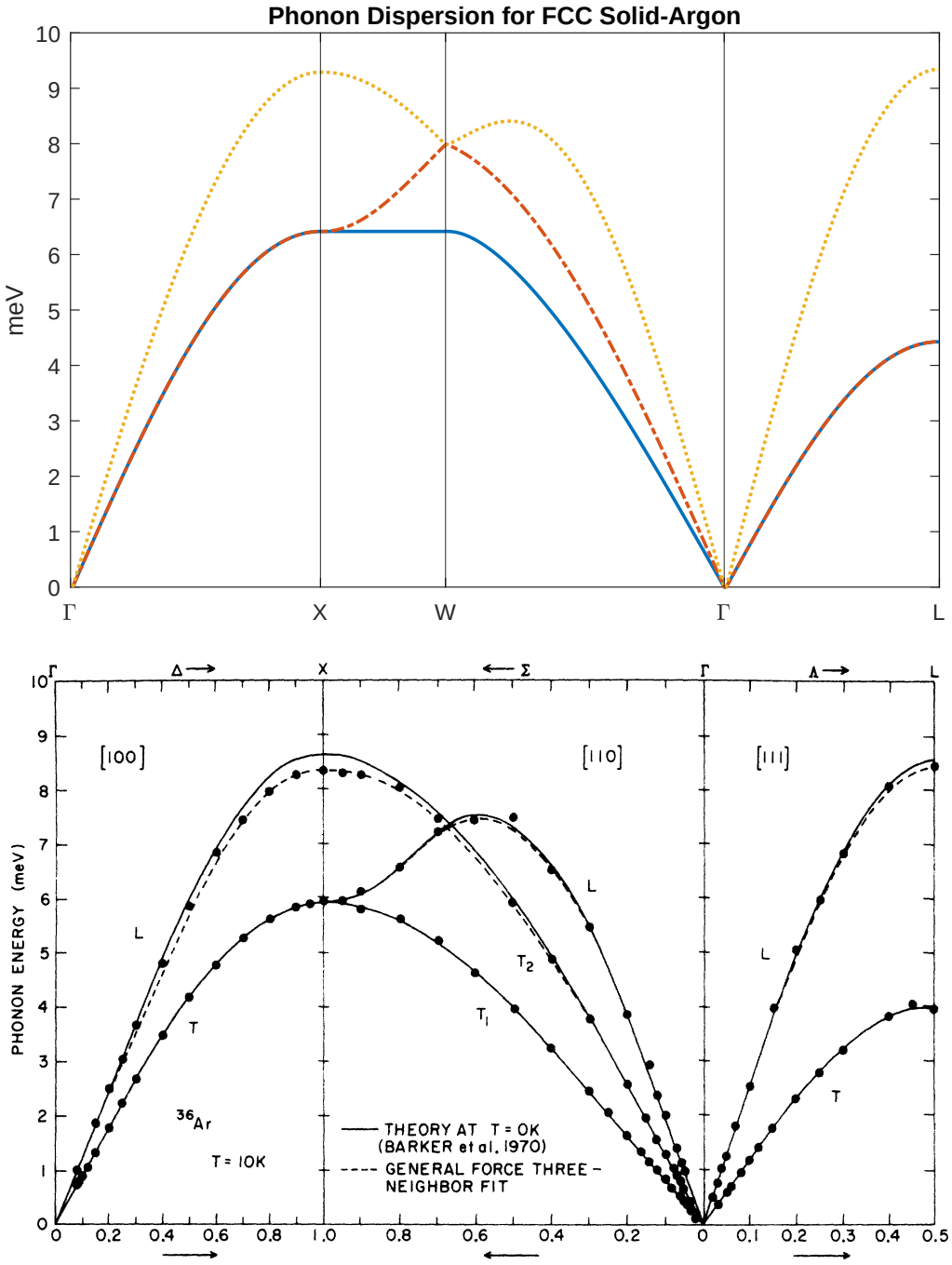


Figure 14: Comparison between our nearest-neighbors phonon calculation for FCC solid Argon (upper panel) and experimental data [dots] and theory [lines] from Ref. [6] (lower panel). Main features are reproduced, except for the unmatched 1st band plateau in our calculation, which we regard as an artifact coming from the nearest neighbors truncation. Frequency values, here expressed in meV, match up to 10% accuracy.

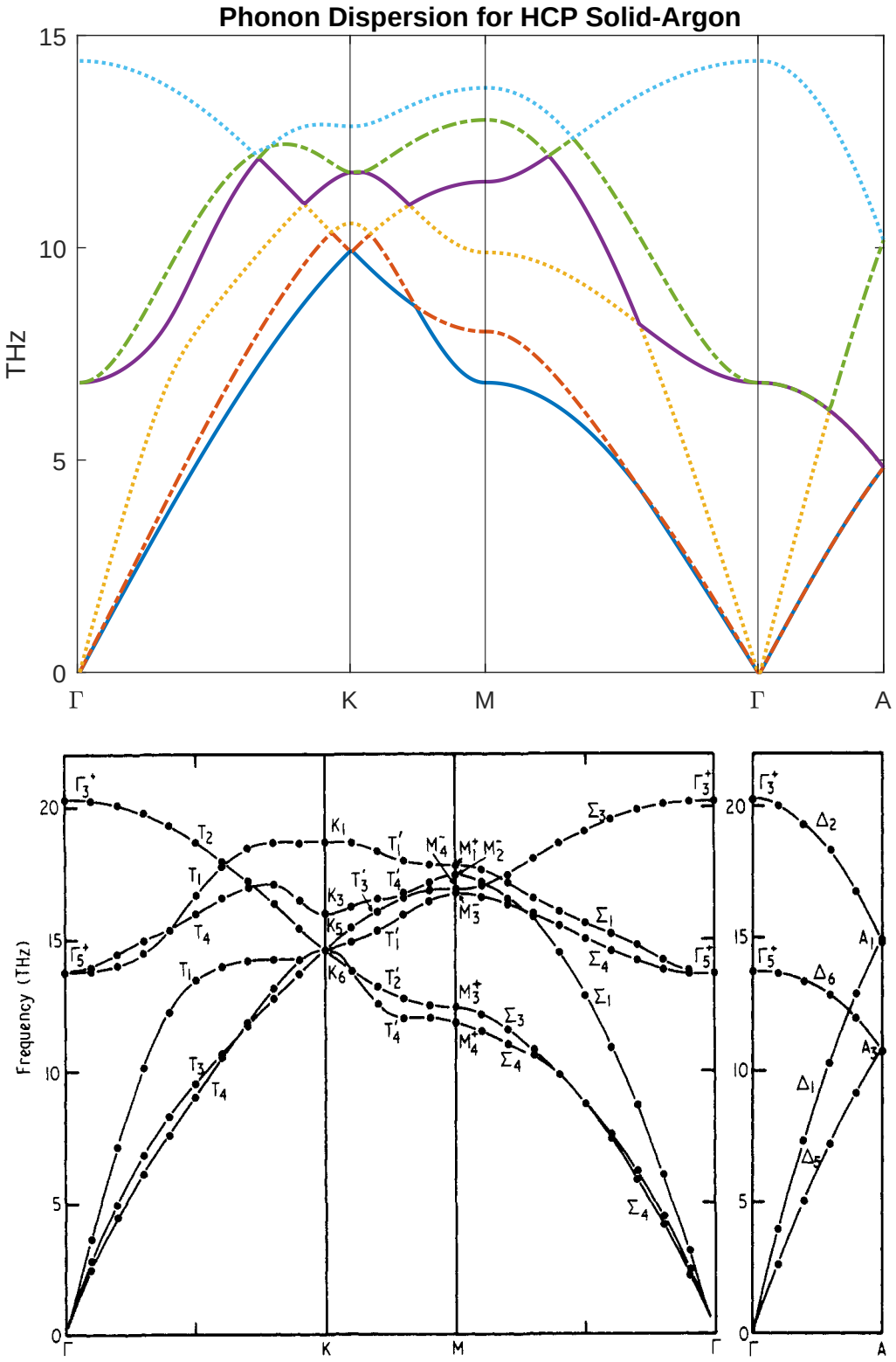


Figure 15: Comparison between our nearest-neighbors phonon calculation for HCP solid Argon (upper panel) and experimental data [dots] for Berillium from Ref. [7] (lower panel). [lines] are interpolations of experimental data. Along the $\Gamma - K - M - \Gamma - A$ path most features are reproduced quite well, despite the very different nature of inter-ion interaction.

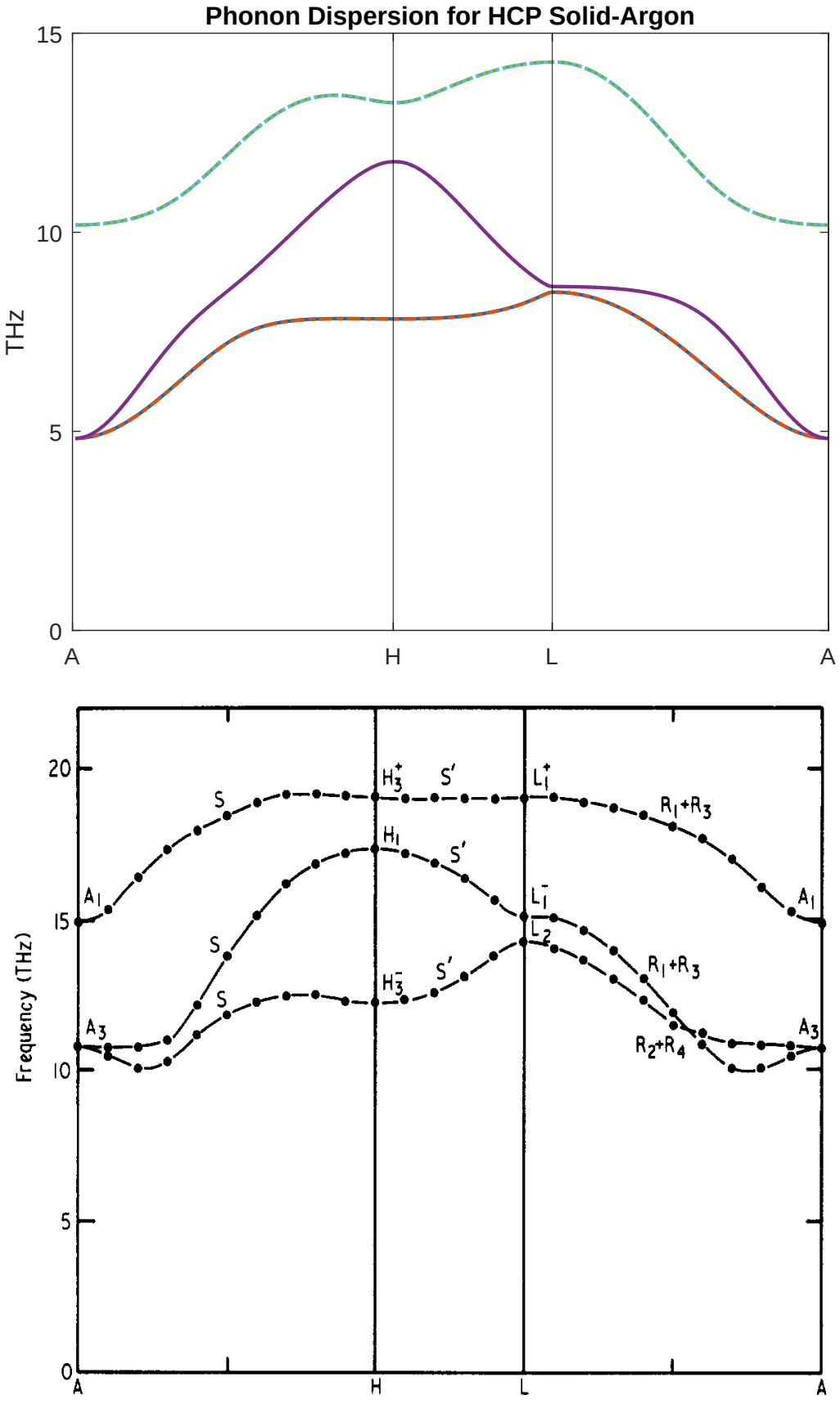


Figure 16: Comparison between our nearest-neighbors phonon calculation for HCP solid Argon (upper panel) and experimental data [dots] for Berillium from Ref. [7] (lower panel). [lines] are interpolations of experimental data. Along the $A-H-L-A$ path most we have some important deviations, especially on the higher (2-fold degenerate) band and on the avoided-crossing between the two lower ones (again, each one 2-fold degenerate).

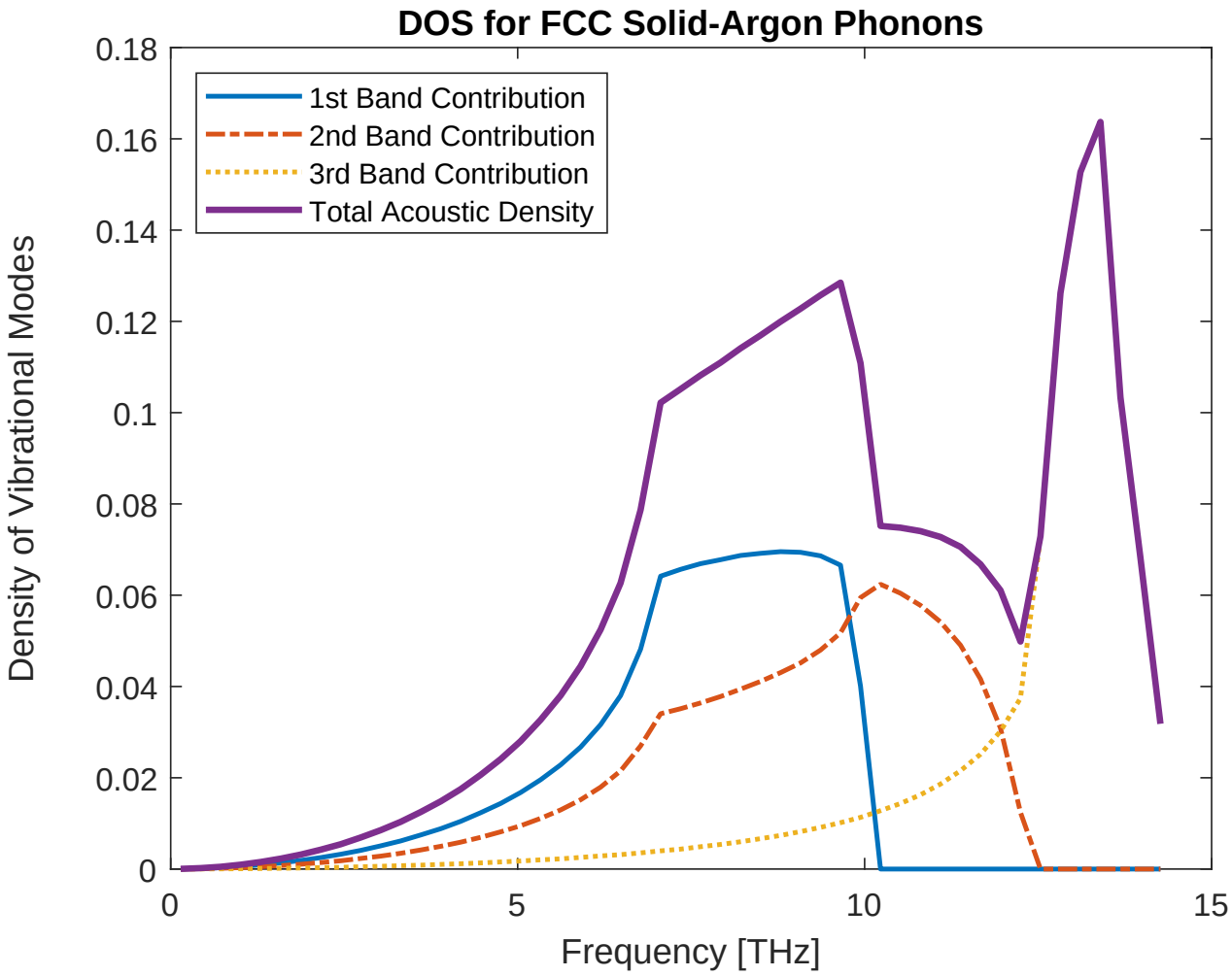


Figure 17: Vibrational Density of States (DOS) for the FCC Solid-Argon structure. The contributions from the single phonon bands are shown in different colors and linetypes.

Phonon Density of States (FCC)

In Figure 17 is reported the vibrational density of states (DOS) for the FCC structure, as computed by histograming the phonon frequencies computed in a reciprocal unit-cell[9], togheter with the single acoustic band contributions. The normalization is such that the sum on the *discrete* sampling grid gives 3 (i.e. the number of modes per lattice unit-cell).

We report also the computed density of states for both a larger and a smaller lattice parameter then the one determined by cohesion energy minimization (respectively right and left panels of Figure 18). We see that the *shape* of the DOS is basically unchanged except for a rescaling of the frequency range (naturally arising from the change of spring costants in the dynamical matrix).

This outstanding result[10] allows us to perform any Brillouin Zone integral by just sampling the appropriate frequency range (which can be inexpensively computed by diagonalizing dynamical matrix on high-symmetry lines only) and then weight the sum by means of the equilibrium volume DOS shape, resulting in a great performance improvement.

In the next section we shall apply this *constant-DOS-shape approximation* to a thermodynamic study of thermal expansion of the FCC phase of Argon.

[9]For the sake of simplicity we sampled $\{\vec{k} = \sum_i c_i \vec{b}_i$, for $c_i \in [0, 1] \ \forall i\}$ instead of the first Brilluoin Zone.
[10]Actually the statement is not really surprising, as we have seen that the phonon dispersion shapes are quite robust for a fixed lattice symmetry, even when the 2-body potential has very different behaviour. (Cf. Figures 15 & 16)

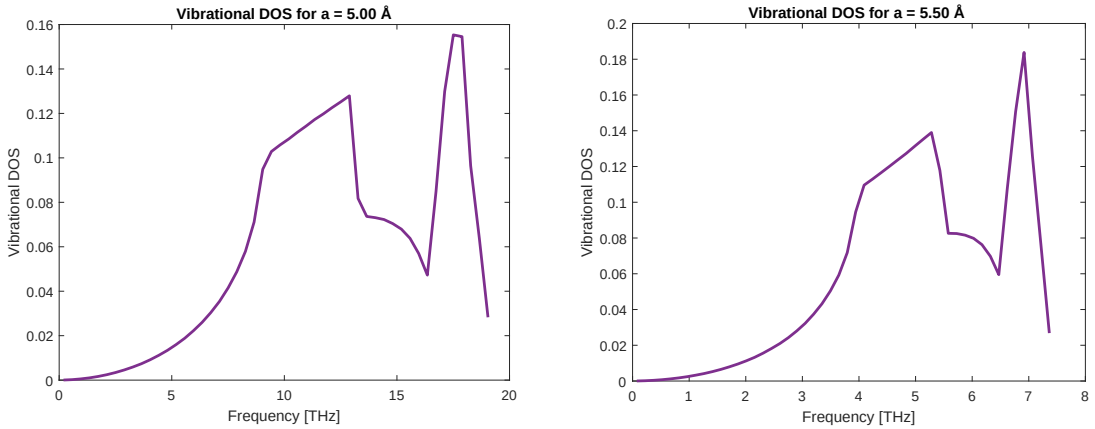


Figure 18: FCC vibrational DOS for slightly smaller (left panel) and larger (right panel) lattice parameter, with respect to the minimal energy one ($a \simeq 5.16$ Å).

Thermal Expansion of FCC Structure

Within harmonic approximation the only role of a finite temperature is to populate the phonon states, which all have the same expectation value for the position operator (corresponding to the equilibrium lattice geometry, around which the harmonic expansion is carried out). Hence it's not a surprise that this picture cannot explain the phenomenon of thermal expansion in solids. Then in principle, to explain thermal expansion one has to include higher order (*anharmonic*) terms in the expansion of the cohesive energy, with a severe drawback in terms of complexity of the task. A much simpler alternative is achieved via the so called *quasi-harmonic approximation*, whose essential details are described in the next section.

Quasi-Harmonic approximation

The goal of quasi-harmonic lattice dynamics (QHLD) is to account for the anharmonicity effects that are responsible for thermal expansion. Therefore the phonon frequencies are assumed to depend *parametrically* on total crystal volume (i.e. on lattice parameter): the lattice dynamics are treated within the harmonic approximation for each fixed value of a that is *not too far* from the equilibrium geometry.[11]

The total energy of the system is then given, within a quantum treatment of the normal modes, as:

$$U(a,T) = U^{\mathrm{eq}}(a) + \frac{1}{2}\sum_{\vec{k},s}\hbar\omega_s(\vec{k},a) + \sum_{\vec{k},s}\frac{\hbar\omega_s(\vec{k},a)}{\exp\left(\frac{-\hbar\omega_s(\vec{k},a)}{k_BT}\right)-1} \quad ,$$

where $\frac{1}{2}\sum_{\vec{k},s}\hbar\omega_s(\vec{k},a)$ is the zero-point vibrational energy, and $U^{\mathrm{eq}}(a)$ the static cohesion energy, as computed previously for zero temperature geometry optimization.

The corresponding Helmholtz free-energy is thus (Ref. [8])

$$F(a,T) = U^{\mathrm{eq}}(a) + \frac{1}{2}\sum_{\vec{k},s}\hbar\omega_s(\vec{k},a) + k_BT\sum_{\vec{k},s}\log\left[1-\exp\left(\frac{-\hbar\omega_s(\vec{k},a)}{k_BT}\right)\right],$$

which can be easily minimized numerically[12], with respect to the lattice parameter a. The resulting optimal lattice parameter will trivially give the equilibrium volume, at any given temperature, thus defining the thermal expansion of the solid.

To be consistent with our previous treatment for the zero temperature geometry optimization we further remove the quantum zero-point energy term, thus defining a *classical* quasi-harmonic free energy to minimize

$$F(a,T) \simeq F_{\mathrm{classical}}(a,T) = U^{\mathrm{eq}}(a) + k_BT\sum_{\vec{k},s}\log\left[1-\exp\left(\frac{-\hbar\omega_s(\vec{k},a)}{k_BT}\right)\right],$$

[11] So, comparing to full anharmonic treatment, we see that phonons are still regarded as independent (noninteracting) particles with infinite lifetime. Regarding the domain of application a rule of thumb is to trust the resulting frequencies only for volumes that keep them real (stable vibrational modes): beyond this point one would expect the system to approach the melting point, a situation that cannot be expected to be well described within such a simple approximation scheme.

[12] Within our constant-DOS-shape approximation.

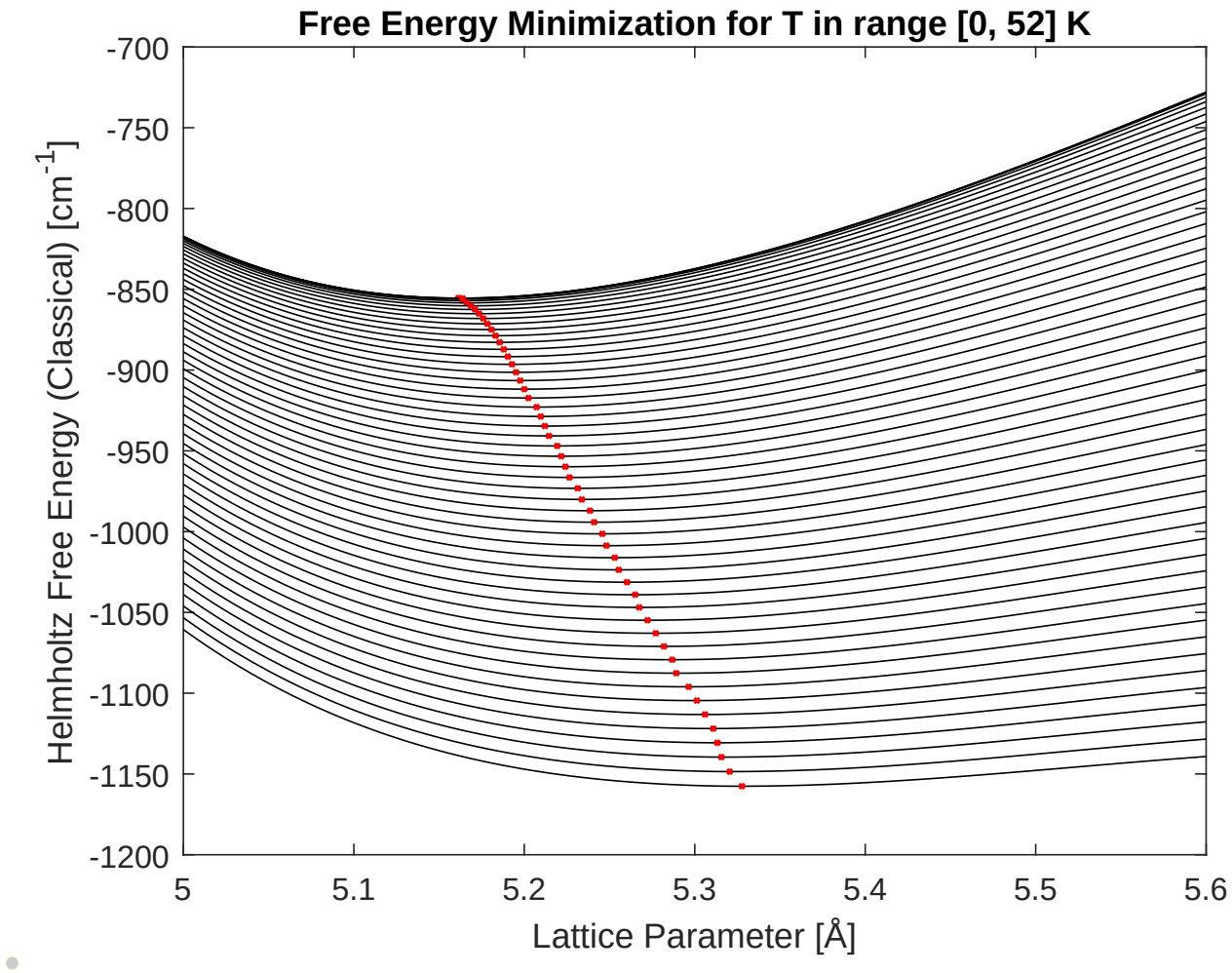


Figure 19: Classical free-energy dependence on lattice parameter for several temperature values. The numerical minima are indicated by red crosses. For further details see the discussion in text.

from which the zero temperature limit gives just the already discussed cohesion energy minimization procedure.

The free energy *vs* volume curves for several given temperatures are shown in Figure 19, where is evident a positive shift (expansion) of the equilibrium lattice parameter for increasing temperatures. We point out that the calculation failed for any greater temperature: the resulting free energy has ill-defined concavity. The reason is not clear to us.[13]

A corresponding *thermal expansion curve* (a_{eq} *vs* T) has been extrapolated with a quadratic fit (Figure 20). While obtaining an excellent *goodness of fit* we might not trust the resulting high-temperature behaviour. Nevertheless the low-temperature linear vanishing of the expansion seems to be reproduced quite satisfactorily with respect to comparable literature data (Cf. Figure 21, Ref. [9]) given that:

- the linear slope apper to be almost identical to reference Classical-QHLD and MD computations;
- the zero temperature intercept has a shift that matches with the difference between our hard-core radius ($\sigma = 3.348$ Å) and the value used by the authors ($\sigma = 3.405$ Å), in turn taken from Ref. [10];

So in conclusion we can recognize quasi-harmonic lattice dynamics as a powerful tool to study thermal expansion for temperatures well below melting point (linear expansion regime).

[13] In principle, doing a proper QHLD calculation, we should obtain meaningful results for any lattice parameter below 5.925 Å, i.e. the largest volume for which we obtain real phonon frequencies. Also the constant-DOS-shape approximation should hold safely until $a = 5.5$ Å (Cf. Figure 18, right panel). Nevertheless the highest acceptable minimum we get is far below both these critical values.

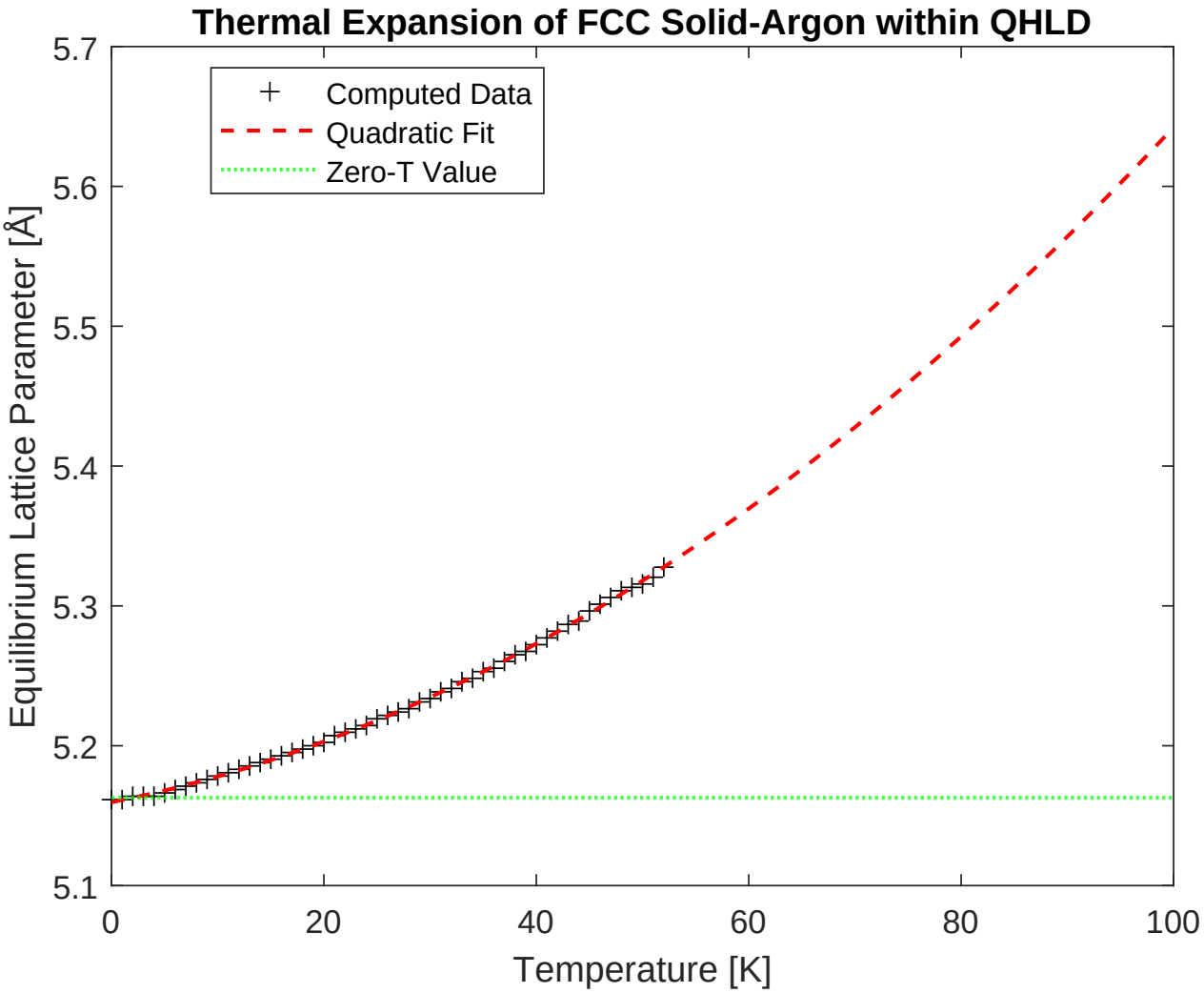


Figure 20: Thermal expansion curve from a quadratic fit of the computed classical free-energy minima. The zero temperature value is trivially recovered; the high-temperature behaviour might not be reliable.

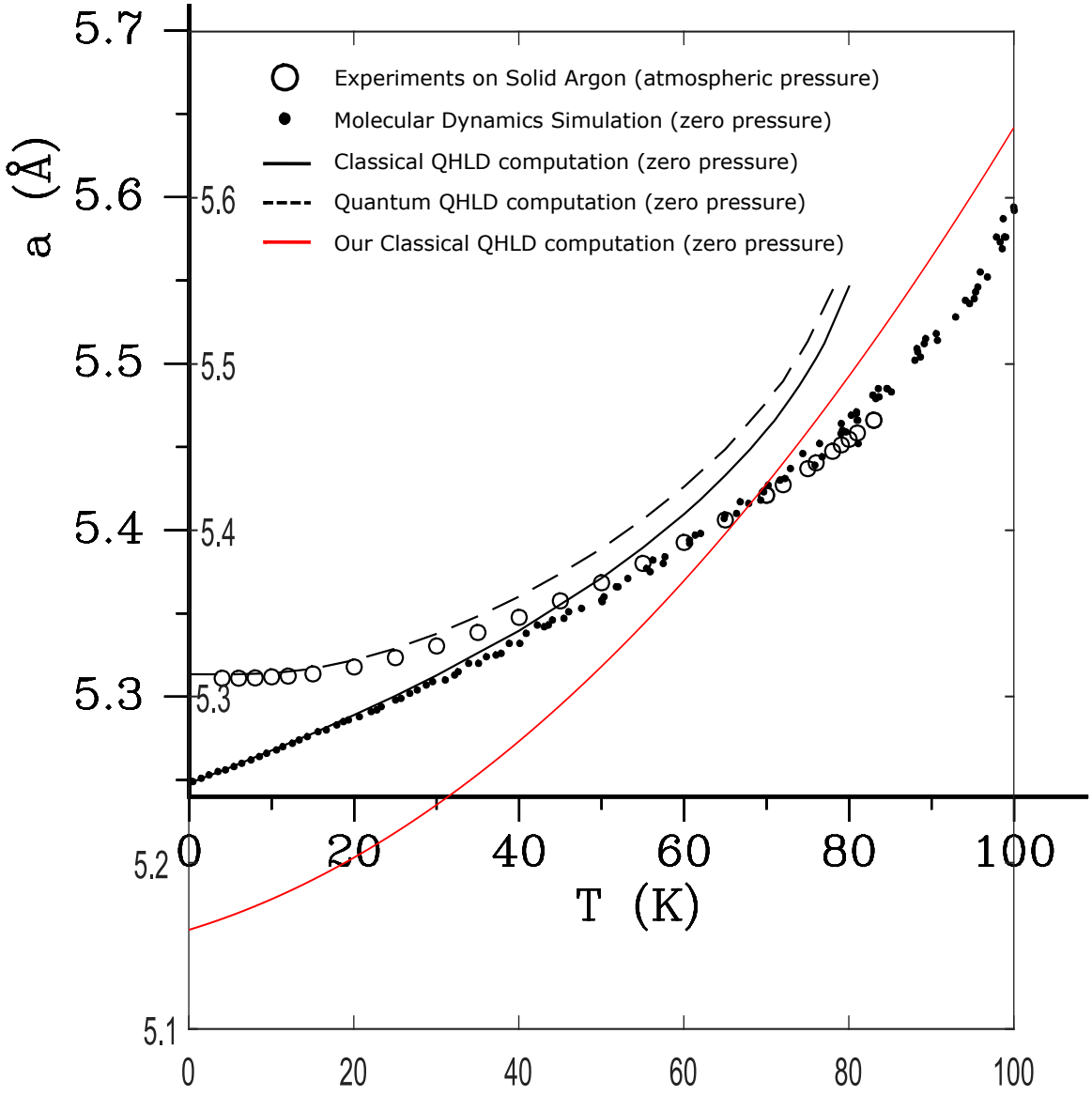


Figure 21: Graphical comparison between our result (red line) and the experimental and computational data reported in Ref. [9]. For further details see the discussion in text.

References

[1] N. W. Ashcroft & N. D. Mermin – Solid State Physics, Saunders College (1976)

[2] P. Schwerdtfeger et al. – Extension of the Lennard-Jones potential: Theoretical investigations into rare-gas clusters and crystal lattices of He, Ne, Ar, and Kr using many-body interaction expansions, *Phys. Rev. B* **73**, 064112 (2006)

[3] N. V. Krainyukova et al. – Observation of the FCC-to-HCP Transition in Ensembles of Argon Nanoclusters, *Phys. Rev. Lett.* **109**, 245505 (2012)

[4] W. Setyawan & S. Curtarolo – High-Throughput Electronic Band Structure Calculations: Challenges and Tools, *Comp. Mat. Sci.* **49**, 2 299-312 (2010)

[5] P. Hadley – Lectures on Molecular and Solid State Physics: http://lampx.tugraz.at/~hadley/ss1

[6] Y. Fujii et al. – Inelastic Neutron Scattering from Solid Ar, *Phys. Rev. B* **10**, 3647 (1974)

[7] R. Stedman et al. – Phonon Spectrum of Beryllium at 80 K, *J. Phys. F: Metal Phys.* **6**, 157-166 (1976)

[8] M. T. Dove – Introduction to Lattice Dynamics, Cambridge Univeristy Press (1993)

[9] R. Della Valle et al. – Quasi Harmonic Lattice Dynamics vs Molecular Dynamics: Thermal Expansion of Solid Argon, *Gazz. Chim. Ital.* **126**, 615-617 (1996)

[10] L. Verlet – Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules, *Phys. Rev.* **159**, 98 (1967)

MATLAB Scripts

Listing 1: Matlab script for supercell generation

```
function [r, Nb, Lx, Ly, Lz]  = generateLattice(a,Nx,Ny,Nz,sym)

N = Nx*Ny*Nz;

% Simple Cubic Lattice
if sym == 'scl'
    Rx = a;
    Ry = a;
    Rz = a;
    L = 2*a;    % "unit" cell length
    Nb = 1;     % Number of basis atoms
    x = -1*ones(1,Nb*N);
    y = -1*ones(1,Nb*N);
    z = -1*ones(1,Nb*N);
    atomcounter = 0;
    for i = -Nx/2:Nx/2
        for j = -Ny/2:Ny/2
            for k = -Nz/2:Nz/2
                atomcounter = atomcounter+1;
                x(atomcounter) = i*Rx;
                y(atomcounter) = j*Ry;
                z(atomcounter) = k*Rz;
            end
        end
    end
    Lx = L;
    Ly = L;
    Lz = L;

% Body Centered Cubic
elseif sym == 'bcc'
    Rx = a;
    Ry = a;
    Rz = a;
    L = 2*a;    % "unit" cell length
    Nb = 2;     % Number of basis atoms
    Delta = [a/2, a/2, a/2]; % Sublattice hopper
    x = -1*ones(1,Nb*N);
    y = -1*ones(1,Nb*N);
    z = -1*ones(1,Nb*N);
    atomcounter = 0;
    for i = -Nx/2:Nx/2
        for j = -Ny/2:Ny/2
            for k = -Nz/2:Nz/2
                atomcounter = atomcounter+1;
                x(atomcounter) = i*Rx;
                y(atomcounter) = j*Ry;
                z(atomcounter) = k*Rz;
                atomcounter = atomcounter+1;
                x(atomcounter) = i*Rx + Delta(1);
                y(atomcounter) = j*Ry + Delta(2);
                z(atomcounter) = k*Rz + Delta(3);
            end
        end
    end
    Lx = L;
    Ly = L;
    Lz = L;

% Face Centered Cubic
elseif sym == 'fcc'
    Rx = a;
    Ry = a;
    Rz = a;
    L = 2*a;    % "unit" cell length
    Nb = 4;     % Number of basis atoms
    Delta1 = [a/2, a/2, 0]; % Sublattice hopper
    Delta2 = [0, a/2, a/2]; % Sublattice hopper
    Delta3 = [a/2, 0, a/2]; % Sublattice hopper
    x = -1*ones(1,Nb*N);
    y = -1*ones(1,Nb*N);
    z = -1*ones(1,Nb*N);
    atomcounter = 0;
```

```
    for i = -Nx/2:Nx/2
        for j = -Ny/2:Ny/2
            for k = -Nz/2:Nz/2
                atomcounter = atomcounter+1;
                x(atomcounter) = i*Rx;
                y(atomcounter) = j*Ry;
                z(atomcounter) = k*Rz;
                atomcounter = atomcounter+1;
                x(atomcounter) = i*Rx + Delta1(1);
                y(atomcounter) = j*Ry + Delta1(2);
                z(atomcounter) = k*Rz + Delta1(3);
                atomcounter = atomcounter+1;
                x(atomcounter) = i*Rx + Delta2(1);
                y(atomcounter) = j*Ry + Delta2(2);
                z(atomcounter) = k*Rz + Delta2(3);
                atomcounter = atomcounter+1;
                x(atomcounter) = i*Rx + Delta3(1);
                y(atomcounter) = j*Ry + Delta3(2);
                z(atomcounter) = k*Rz + Delta3(3);
            end
        end
    end
    Lx = L;
    Ly = L;
    Lz = L;

% Hexagonal Close Packed
elseif sym == 'hcp'
    c = sqrt(8/3)*a;
    a1 = [a, 0, 0];
    a2 = [a/2, sqrt(3)/2*a, 0];
    a3 = [0, 0, c];
    Lx = 2*sqrt(3)*a;           % "unit" cell lengths
    Ly = 2*2*a;                 % "unit" cell lengths
    Lz = 2*c;                   % "unit" cell lengths
    Nb = 2;                     % Number of basis atoms
    Delta = 1/3*(a1+a2)+1/2*(a3); % Sublattice hopper
    x = -1*ones(1,Nb*N);
    y = -1*ones(1,Nb*N);
    z = -1*ones(1,Nb*N);
    atomcounter = 0;
    for i = -Nx/2:Nx/2
        for j = -Ny/2:Ny/2
            for k = -Nz/2:Nz/2
                atomcounter = atomcounter+1;
                x(atomcounter) = i*a1(1) + j*a2(1) + k*a3(1);
                y(atomcounter) = i*a1(2) + j*a2(2) + k*a3(2);
                z(atomcounter) = i*a1(3) + j*a2(3) + k*a3(3);
                atomcounter = atomcounter+1;
                x(atomcounter) = i*a1(1) + j*a2(1) + k*a3(1) + Delta(1);
                y(atomcounter) = i*a1(2) + j*a2(2) + k*a3(2) + Delta(2);
                z(atomcounter) = i*a1(3) + j*a2(3) + k*a3(3) + Delta(3);
            end
        end
    end

% Error Message
else
    fprintf('ERROR: Bravais Structure not recognized!\n');
end

% Embeed all coordinates
r = [x; y; z]; % i.e. r(1,:) == x and same for y <-> 2 and z <-> 3

end
```

Listing 2: `Matlab` script for cohesion energy evaluation

```
function E = cohesionEnergy(x,y,z)
% Total Energy (per atom)
    N = length(x);
    j = find(x==0&y==0&z==0);
    E = 0;
    for i = 1:N
        dx = x(j) - x(i); dy = y(j) - y(i); dz = z(j) - z(i);
        rij = norm([dx,dy,dz]);
```

```
        if rij > 0
            E = E + 1/2*LJpot(rij);
        end
    end
end

function V = LJpot(r)
% Lennard-Jones Potential:

    % van der Waals radius (experimental)
    r0 = 3.758/2^(1/6); %\AA
    % Binding Energy
    e = 99.55; %cm^{-1}
    Repulsive = 4*e.*(r0./r).^12;
    Attractive = -4*e.*(r0./r).^6;
    V = Repulsive + Attractive;

end
```

Listing 3: `Matlab` script for supercell graphical displaying

```
% Lattice Parameter
a =  3.5734; %\AA [3.5734 SC | 4.1314 BCC | 5.1628 FCC | 3.6502 HCP]
% vdW Radius
r0 =  3.758/2; %\AA

% Symmetry Selection
sym = 'scl'; %['scl','bcc','fcc','hcp']

% Number of Cells
Nx = 13
Ny = 13
Nz = 13

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[r, Nb, Lx, Ly, Lz] = generateLattice(a,Nx,Ny,Nz,sym);
x = r(1,:); y = r(2,:); z = r(3,:);
rOLD = (r' - [min(x), min(y), min(z)])';
xOLD = rOLD(1,:); yOLD = rOLD(2,:); zOLD = rOLD(3,:);

drawcrystal(xOLD,yOLD,zOLD,a,r0,Lx,Ly,Lz,Nb);
print(gcf,'SC_SuperCell','-fillpage','-dpdf','-r720')

function drawcrystal(x,y,z,a,r0,Lx,Ly,Lz,Nb)
N = length(x)/Nb;
% Displaying Crystal
figure("Name", "Crystal")
[Sx,Sy,Sz] = sphere;
for i = 1:Nb*N
    if mod(i,Nb) == 0
    if ~((x(i) >= Lx-a/2 || y(i) >= Ly-a/2 || z(i) >= Lz-a/2) && N <= 8)
    surf(x(i)+Sx*r0,y(i)+Sy*r0,z(i)+Sz*r0,'FaceColor',[0 .75 .75],'EdgeColor','none','
    FaceLighting','gouraud'); hold on
    end
    elseif mod(i,Nb) == 1
    if ~((x(i) >= Lx-a/2 || y(i) >= Ly-a/2 || z(i) >= Lz-a/2) && N <= 8)
        surf(x(i)+Sx*r0,y(i)+Sy*r0,z(i)+Sz*r0,'FaceColor',[.75 .75 0],'EdgeColor','none','
    FaceLighting','gouraud'); hold on
    end
    elseif mod(i,Nb) == 2
    if ~((x(i) >= Lx-a/2 || y(i) >= Ly-a/2 || z(i) >= Lz-a/2) && N <= 8)
        surf(x(i)+Sx*r0,y(i)+Sy*r0,z(i)+Sz*r0,'FaceColor',[.75 0 .75],'EdgeColor','none','
    FaceLighting','gouraud'); hold on
    end
    elseif mod(i,Nb) == 3
    if ~((x(i) >= Lx-a/2 || y(i) >= Ly-a/2 || z(i) >= Lz-a/2) && N <= 8)
        surf(x(i)+Sx*r0,y(i)+Sy*r0,z(i)+Sz*r0,'FaceColor',[.75 .75 .75],'EdgeColor','none'
    ,'FaceLighting','gouraud'); hold on
    end
    end
end
plotcube([Lx Ly Lz],[-a/2 -a/2 -a/2],.1,[0.3 0.3 0.3]); % Conventional Cell Displaying
axis equal
view(24.2576,19.5898);
lightangle(24.2576,19.5898); % add some light
```

```
end

function plotcube(varargin)
% PLOTCUBE - Display a 3D-cube in the current axes
%
%   PLOTCUBE(EDGES,ORIGIN,ALPHA,COLOR) displays a 3D-cube in the current axes
%   with the following properties:
%   * EDGES : 3-elements vector that defines the length of cube edges
%   * ORIGIN: 3-elements vector that defines the start point of the cube
%   * ALPHA : scalar that defines the transparency of the cube faces (from 0
%             to 1)
%   * COLOR : 3-elements vector that defines the faces color of the cube

% Default input arguments
inArgs = { ...
  [10 56 100] , ... % Default edge sizes (x,y and z)
  [10 10  10] , ... % Default coordinates of the origin point of the cube
  .7          , ... % Default alpha value for the cube's faces
  [1 0 0]       ... % Default Color for the cube
  };

% Replace default input arguments by input values
inArgs(1:nargin) = varargin;

% Create all variables
[edges,origin,alpha,clr] = deal(inArgs{:});

XYZ = { ...
  [0 0 0 0]  [0 0 1 1]  [0 1 1 0] ; ...
  [1 1 1 1]  [0 0 1 1]  [0 1 1 0] ; ...
  [0 1 1 0]  [0 0 0 0]  [0 0 1 1] ; ...
  [0 1 1 0]  [1 1 1 1]  [0 0 1 1] ; ...
  [0 1 1 0]  [0 0 1 1]  [0 0 0 0] ; ...
  [0 1 1 0]  [0 0 1 1]  [1 1 1 1]   ...
  };

XYZ = mat2cell(...
  cellfun( @(x,y,z) x*y+z , ...
    XYZ , ...
    repmat(mat2cell(edges,1,[1 1 1]),6,1) , ...
    repmat(mat2cell(origin,1,[1 1 1]),6,1) , ...
    'UniformOutput',false), ...
  6,[1 1 1]);


cellfun(@patch,XYZ{1},XYZ{2},XYZ{3},...
  repmat({clr},6,1),...
  repmat({'FaceAlpha'},6,1),...
  repmat({alpha},6,1)...
  );

view(3);
end
```

Listing 4: Matlab script for geometry optimization

```
% van der Waals radius (experimental)
r0 = 3.758/2^(1/6); %\AA
% Binding Energy
e = 99.55; % cm^{-1}

% Symmetry Selection
sym = 'fcc'; %['scl','bcc','fcc','hcp']

% Number of Cells
Nx = 50;
Ny = 50;
Nz = 50;

Nsteps = 11;
LatticeParameters = (1.52+ 0.005*(0:Nsteps-1))*r0; % [INIT: 1.04 scl | 1.21 bcc | 1.52 fcc
     | 1.065 hcp]
E = zeros(1,Nsteps);

for i = 1:Nsteps
    a = LatticeParameters(i);
```

```

[r, Nb, Lx, Ly, Lz] = generateLattice(a,Nx,Ny,Nz,sym);
x = r(1,:); y = r(2,:); z = r(3,:);

% Compute cohesion energy
E(i) = cohesionEnergy(x,y,z);

end

scatter(LatticeParameters,E,'.k');
fitted = fit(LatticeParameters','E','poly3');
plot(fitted);

h = findobj(gca,'Type','line')
x = h.XData;
y = h.YData;
Emin = min(y)
OptimalLatticeParameter = x(y==min(y))
hold on
scatter(OptimalLatticeParameter,Emin,125,'x','r','LineWidth',2);
xline(OptimalLatticeParameter,'--r');
yline(Emin,'--r');
axis square
xlabel('Lattice Parameter [\AA]','Interpreter','tex');
ylabel('Cohesion Energy [cm^{-1}]','Interpreter','tex');
legend({'Computed Data', 'Polynomial Fit', sprintf('(%f,%f)',OptimalLatticeParameter,Emin)
    },'FontSize', 14)
title('Geometry Optimization for FCC lattice')
set(gca,'FontSize',14)
```

Listing 5: Matlab script for theoretical equilibrium parameters determination

```
% van der Waals radius (experimental)
r0 = 3.758; /AA
% Binding Energy
e = 99.55; % cm^{-1}

% Symmetry Selection
sym = {'scl','bcc','fcc','hcp'};
for i = 1:4

if sym{i} == 'scl'
    L6 = 8.40192397482537;
    L12 = 6.20214904504752;
    gFactor = 1; % Geometry of the lattice
elseif sym{i} == 'bcc'
    L6 = 12.2536678672899;
    L12 = 9.11418326807536;
    gFactor = 3^(-1/2)*2; % Geometry of the lattice
elseif sym{i} == 'fcc'
    L6 = 14.4539210437416;
    L12 = 12.1318801965446;
    gFactor = 2^(1/2); % Geometry of the lattice
elseif sym{i} == 'hcp'
    L6 = 14.4548972778391;
    L12 = 12.1322937690989;
    gFactor = 1; % Geometry of the lattice
end

req = (L12/L6)^(1/6)*r0; % N.N. Distance
Ucoh = 1/2*e*L6^2/L12;
aeq = req * gFactor; % Geometry of the lattice

fprintf('\n**************************************\n')
fprintf('For %s lattice:\n',sym{i})
fprintf('Cohesion Energy is: %f\n',Ucoh)
fprintf('Equilibrium Lattice Parameter is: %f\n',aeq)
fprintf('**************************************\n')

end
```

Listing 6: Matlab script for TL convergence of cohesion energy

```
% van der Waals radius (experimental)
r0 = 3.758/2^(1/6); %\AA
% Binding Energy
e = 99.55; %cm^{-1}
% Lattice Parameter
a =  3.6502; %\AA [3.5734 SC | 4.1314 BCC | 5.1628 FCC | 3.6502 HCP]

% Symmetry Selection
sym = 'hcp'; %['scl','bcc','fcc','hcp']

% Number of Cells
E2 = zeros(1,4)
N = E2;
for Nx = 2:2:50
Ny = Nx;
Nz = Nx;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[r, Nb, Lx, Ly, Lz] = generateLattice(a,Nx,Ny,Nz,sym);
x = r(1,:); y = r(2,:); z = r(3,:);
% Compute cohesion energy
N(Nx/2) = length(x);
E(Nx/2) = cohesionEnergy(x,y,z);
end
plot(N,E2,'.-'); hold on
```

Listing 7: Matlab script for FCC phonons

```
% LJ Parameters
r0 = 3.758/2^(1/6); %\AA
e = 99.55; %cm^{-1}

% Equilibrium Lattice Parameter
a = 5.1628; %\AA

% LJ Derivatives [NUMERICAL]
l = a/sqrt(2);
r = linspace(l,l+1/10^4,10^3);
ndV = (LJpot(r(2))-LJpot(r(1)))/(r(2)-r(1));
r = [l-(r(2)-r(1)),r];
nddV = (LJpot(r(3))+LJpot(r(1))-2*LJpot(r(2)))/(r(2)-r(1))^2;

% LJ Derivatives [ANALYTICAL]
l = a/sqrt(2);
dV = 4*e*(6*r0^6/l^7 - 12*r0^12/l^13);
ddV = 4*e*(156*r0^12/l^14 - 42*r0^6/l^8);

% Nearest Neighbors List
R = {[a/2, a/2, 0], [0, a/2, a/2], [a/2, 0, a/2], [-a/2, -a/2, 0], [0, -a/2, -a/2], [-a/2,
     0, -a/2], [a/2, -a/2, 0], [0, a/2, -a/2], [a/2, 0, -a/2], [-a/2, a/2, 0], [0, -a/2, a
    /2], [-a/2, 0, a/2]};

% High Symmetry k-points
    G = [0,0,0];
    X = [2*pi/a,0,0];
    L = [pi/a,pi/a,pi/a];
    W = [2*pi/a,pi/a,0];
    K = [3/2*pi/a,3/2*pi/a,0];

% High Symmetry Paths
Nfactor = 100;
Npoints = cell(1,5);
    %G-X
    Npoints{1} = round(Nfactor*norm(G-X));
    t = linspace(0,1,Npoints{1});
    Kx{1} = G(1) + (X(1)-G(1))*t;
    Ky{1} = G(2) + (X(2)-G(2))*t;
    Kz{1} = G(3) + (X(3)-G(3))*t;
    %X-W
    Npoints{2} = round(Nfactor*norm(X-W));
    t = linspace(0,1,Npoints{2});
    Kx{2} = X(1) + (W(1)-X(1))*t;
```

```
    Ky{2} = X(2) + (W(2)-X(2))*t;
    Kz{2} = X(3) + (W(3)-X(3))*t;
    %W-K
    Npoints{3} = round(Nfactor*norm(W-K));
    t = linspace(0,1,Npoints{3});
    Kx{3} = W(1) + (K(1)-W(1))*t;
    Ky{3} = W(2) + (K(2)-W(2))*t;
    Kz{3} = W(3) + (K(3)-W(3))*t;
    %K-G
    Npoints{4} = round(Nfactor*norm(K-G));
    t = linspace(0,1,Npoints{4});
    Kx{4} = K(1) + (G(1)-K(1))*t;
    Ky{4} = K(2) + (G(2)-K(2))*t;
    Kz{4} = K(3) + (G(3)-K(3))*t;
    %G-L
    Npoints{5} = round(Nfactor*norm(G-L));
    t = linspace(0,1,Npoints{5});
    Kx{5} = G(1) + (L(1)-G(1))*t;
    Ky{5} = G(2) + (L(2)-G(2))*t;
    Kz{5} = G(3) + (L(3)-G(3))*t;


freqs = cell(1,3);
freqs{1} = zeros(1,sum([Npoints{:}]));
freqs{2} = zeros(1,sum([Npoints{:}]));
freqs{3} = zeros(1,sum([Npoints{:}]));
counter = 0;
for i = 1:5 % Number of HighSym Lines
    kx = Kx{i};
    ky = Ky{i};
    kz = Kz{i};
    for j = 1:Npoints{i}
        k = [kx(j),ky(j),kz(j)];
        % Dynamical Matrix
        A = 2*dV/l;
        B = 2*ddV-A;
        D = zeros(3,3);
        for r = 1:length(R)
            D = D + ((\sin(1/2*k*R{r}'))^2*(A*eye(3) + B*(R{r}'*R{r})./(norm(R{r}))^2));
        end
        % EigenThings
        counter = counter + 1;
        eigs = eig(D)*(12*1.6*10^(-4))/(39.948*1.66*10^(-27));
        freqs{1}(counter) = sqrt(eigs(1))/10^(12); %
        freqs{2}(counter) = sqrt(eigs(2))/10^(12); % x 0.4135665538536/2*pi for meV
        freqs{3}(counter) = sqrt(eigs(3))/10^(12); %
    end
end
k = 1:sum([Npoints{:}]);
figure("Name",'FCC Dispersion')
plot(k,freqs{1},'-','LineWidth',1.5); hold on
plot(k,freqs{2},'-.','LineWidth',1.5);
plot(k,freqs{3},':','LineWidth',1.5);
Value = 0;
Tick = cell(1,6);
Tick{1} = 0;
for i = 1:5
    Value = Value + Npoints{i};
    Tick{i+1} = Value;
    xline(Value,'k');
end
xlim([0,sum([Npoints{:}])]);
xticks([Tick{:}])
xticklabels({'\Gamma','X','W','K','\Gamma','L'})
title('Phonon Dispersion for FCC Solid-Argon');
ylabel('THz'); % meV
```

Listing 8: Matlab script for HCP phonons

```
% LJ Parameters
r0 = 3.758/2^(1/6); %\AA
e = 99.55; %cm^{-1}

% Equilibrium Lattice Parameters
a = 3.6502; %\AA
c = sqrt(8/3)*a;

% LJ Derivatives [NUMERICAL]
l = a;
r = linspace(l,l+1/10^4,10^3);
ndV = (LJpot(r(2))-LJpot(r(1)))/(r(2)-r(1));
r = [l-(r(2)-r(1)),r];
nddV = (LJpot(r(3))+LJpot(r(1))-2*LJpot(r(2)))/(r(2)-r(1))^2;

% LJ Derivatives [ANALYTICAL]
l = a;
dV = 4*e*(6*r0^6/l^7 - 12*r0^12/l^13);
ddV = 4*e*(156*r0^12/l^14 - 42*r0^6/l^8);

% Nearest Neighbors Lists
RA = {[a, 0, 0], -[a,0,0],[a/2, sqrt(3)*a/2,0], -[a/2, sqrt(3)*a/2,0], [-a/2,sqrt(3)*a
    /2,0], -[-a/2,sqrt(3)*a/2,0]};
RAB = {[0,0,0], [a,0,0], [a/2,sqrt(3)/2*a,0], [0,0,sqrt(8/3)*a], [a,0,sqrt(8/3)*a], [a/2,
    sqrt(3)/2*a, sqrt(8/3)*a]};
RBA = {-[0,0,0], -[a,0,0], -[a/2,sqrt(3)/2*a,0], -[0,0,sqrt(8/3)*a], -[a,0,sqrt(8/3)*a],
    -[a/2, sqrt(3)/2*a, sqrt(8/3)*a]};
SB = [a/2,a/(2*sqrt(3)),sqrt(2/3)*a];

% High Symmetry k-points
    G = [0,0,0];
    A = [0,0,pi/c];
    H = [4/3*pi/a,0,pi/c];
    L = [pi/a,-pi/(sqrt(3)*a),pi/c];
    M = [pi/a,-pi/(sqrt(3)*a),0];
    K = [4/3*pi/a,0,0];

% High Symmetry Paths
Nfactor = 100;
Npoints = cell(1,7);
    %G-M
    Npoints{1} = round(Nfactor*norm(G-M));
    t = linspace(0,1,Npoints{1});
    Kx{1} = G(1) + (M(1)-G(1))*t;
    Ky{1} = G(2) + (M(2)-G(2))*t;
    Kz{1} = G(3) + (M(3)-G(3))*t;
    %M-K
    Npoints{2} = round(Nfactor*norm(M-K));
    t = linspace(0,1,Npoints{2});
    Kx{2} = M(1) + (K(1)-M(1))*t;
    Ky{2} = M(2) + (K(2)-M(2))*t;
    Kz{2} = M(3) + (K(3)-M(3))*t;
    %K-G
    Npoints{3} = round(Nfactor*norm(K-G));
    t = linspace(0,1,Npoints{3});
    Kx{3} = K(1) + (G(1)-K(1))*t;
    Ky{3} = K(2) + (G(2)-K(2))*t;
    Kz{3} = K(3) + (G(3)-K(3))*t;
    %G-A
    Npoints{4} = round(Nfactor*norm(G-A));
    t = linspace(0,1,Npoints{4});
    Kx{4} = G(1) + (A(1)-G(1))*t;
    Ky{4} = G(2) + (A(2)-G(2))*t;
    Kz{4} = G(3) + (A(3)-G(3))*t;
    %A-L
    Npoints{5} = round(Nfactor*norm(A-L));
    t = linspace(0,1,Npoints{5});
    Kx{5} = A(1) + (L(1)-A(1))*t;
    Ky{5} = A(2) + (L(2)-A(2))*t;
    Kz{5} = A(3) + (L(3)-A(3))*t;
    %L-H
    Npoints{6} = round(Nfactor*norm(L-H));
    t = linspace(0,1,Npoints{6});
    Kx{6} = L(1) + (H(1)-L(1))*t;
```

```
    Ky{6} = L(2) + (H(2)-L(2))*t;
    Kz{6} = L(3) + (H(3)-L(3))*t;
    %H-A
    Npoints{7} = round(Nfactor*norm(H-A));
    t = linspace(0,1,Npoints{7});
    Kx{7} = H(1) + (A(1)-H(1))*t;
    Ky{7} = H(2) + (A(2)-H(2))*t;
    Kz{7} = H(3) + (A(3)-H(3))*t;

freqs = cell(1,6);
freqs{1} = zeros(1,sum([Npoints{:}]));
freqs{2} = zeros(1,sum([Npoints{:}]));
freqs{3} = zeros(1,sum([Npoints{:}]));
freqs{4} = zeros(1,sum([Npoints{:}]));
freqs{5} = zeros(1,sum([Npoints{:}]));
freqs{6} = zeros(1,sum([Npoints{:}]));
counter = 0;
for i = 1:7 % Number of HighSym Lines
    kx = Kx{i};
    ky = Ky{i};
    kz = Kz{i};
    for j = 1:Npoints{i}
        k = [kx(j),ky(j),kz(j)];
        % Dynamical Matrix
        A0 = 2*dV/l;
        B0 = 2*ddV-A0;
        D = zeros(6,6);
        % In-plane Blocks
        for r = 1:length(RA)
            D(1:3,1:3) = D(1:3,1:3) + (\sin(1/2*k*RA{r}'))^2*(A0*eye(3) + B0*(RA{r}'*RA{r
    }).*(norm(RA{r}))^2);
            D(1:3,1:3) = D(1:3,1:3) + A0/2*eye(3) + B0/2*((RAB{r}-SB)'*(RAB{r}-SB)).*(norm
    (RAB{r}-SB))^2;
            D(4:6,4:6) = D(4:6,4:6) + (\sin(1/2*k*RA{r}'))^2*(A0*eye(3) + B0*(RA{r}'*RA{r
    }).*(norm(RA{r}))^2);
            D(4:6,4:6) = D(4:6,4:6) + A0/2*eye(3) + B0/2*((RAB{r}-SB)'*(RAB{r}-SB)).*(norm
    (RAB{r}-SB))^2;
        end
        % Inter-plane Blocks
        for r = 1:length(RAB)
            D(1:3,4:6) = D(1:3,4:6) - exp(1i*k*RAB{r}')*(A0/2*eye(3) + B0/2*((RAB{r}-SB)
    '*(RAB{r}-SB)).*(norm(RAB{r}-SB))^2);
            D(4:6,1:3) = D(4:6,1:3) - exp(1i*k*RBA{r}')*(A0/2*eye(3) + B0/2*((RBA{r}+SB)
    '*(RBA{r}+SB)).*(norm(RBA{r}+SB))^2);
        end
        % EigenThings
        counter = counter + 1;
        eigs = eig(D)*(12*1.6*10^(-4))/(39.948*1.66*10^(-27));
        freqs{1}(counter) = sqrt(eigs(1))/10^(12);
        freqs{2}(counter) = sqrt(eigs(2))/10^(12);
        freqs{3}(counter) = sqrt(eigs(3))/10^(12);
        freqs{4}(counter) = sqrt(eigs(4))/10^(12);
        freqs{5}(counter) = sqrt(eigs(5))/10^(12);
        freqs{6}(counter) = sqrt(eigs(6))/10^(12);
    end
end
k = 1:sum([Npoints{:}]);
figure("Name",'HCP Dispersion')
plot(k,freqs{1},'-','LineWidth',1.5); hold on
plot(k,freqs{2},'-.','LineWidth',1.5);
plot(k,freqs{3},':','LineWidth',1.5);
plot(k,freqs{4},'-','LineWidth',1.5);
plot(k,freqs{5},'-.','LineWidth',1.5);
plot(k,freqs{6},':','LineWidth',1.5);
Value = 0;
Tick = cell(1,6);
Tick{1} = 0;
for i = 1:7
    Value = Value + Npoints{i};
    Tick{i+1} = Value;
    xline(Value,'k');
end
xlim([0,sum([Npoints{:}])]);
xticks([Tick{:}])
xticklabels({'\Gamma','M','K','\Gamma','A','L','H','A'})
title('Phonon Dispersion for HCP Solid-Argon');
ylabel('THz');
```

Listing 9: Matlab script for FCC DOS

```
% LJ Parameters
r0 = 3.758/2^(1/6); %\AA
e = 99.55; %cm^{-1}

% Equilibrium Lattice Parameter
a = 5.75 %5.1628; %\AA

% LJ Derivatives [NUMERICAL]
l = a/sqrt(2);
r = linspace(l,l+1/10^4,10^3);
ndV = (LJpot(r(2))-LJpot(r(1)))/(r(2)-r(1));
r = [l-(r(2)-r(1)),r];
nddV = (LJpot(r(3))+LJpot(r(1))-2*LJpot(r(2)))/(r(2)-r(1))^2;

% LJ Derivatives [ANALYTICAL]
l = a/sqrt(2);
dV = 4*e*(6*r0^6/l^7 - 12*r0^12/l^13);
ddV = 4*e*(156*r0^12/l^14 - 42*r0^6/l^8);

% Nearest Neighbors List
R = {[a/2, a/2, 0], [0, a/2, a/2], [a/2, 0, a/2], [-a/2, -a/2, 0], [0, -a/2, -a/2], [-a/2,
     0, -a/2], [a/2, -a/2, 0], [0, a/2, -a/2], [a/2, 0, -a/2], [-a/2, a/2, 0], [0, -a/2, a
    /2], [-a/2, 0, a/2]};

% Cubic Unit-Cell in Reciprocal Space
b1 = 2*pi/a*[1, -1, 1]; %
b2 = 2*pi/a*[1, 1, -1]; % \vec{k} = k1\vec{b1} + k2\vec{b2} + k3\vec{b3}
b3 = 2*pi/a*[-1, 1, 1]; %
Npoints = 250;
k1 = linspace(0,1,Npoints);
k2 = linspace(0,1,Npoints);
k3 = linspace(0,1,Npoints);
% Translating Back to XYZ frame
kpoint = cell(1,Npoints^3);
kCounter = 0;
for i = 1:Npoints
    for j=1:Npoints
        for l=1:Npoints
            kCounter = kCounter + 1;
            kpoint{kCounter} = k1(i).*b1 + k2(j).*b2 + k3(l).*b3;
            %scatter3(kpoint{kCounter}(1),kpoint{kCounter}(2),kpoint{kCounter}(3)); hold
    on
        end
    end
end

freqs = cell(1,3);
freqs{1} = zeros(1,Npoints^3);
freqs{2} = zeros(1,Npoints^3);
freqs{3} = zeros(1,Npoints^3);
counter = 0;
for j = 1:Npoints^3
    k = kpoint{j};
    % Dynamical Matrix
    A = 2*dV/a;
    B = 2*ddV-A;
    D = zeros(3,3);
    for r = 1:length(R)
        D = D + ((\sin(1/2*k*R{r}'))^2*(A*eye(3) + B*(R{r}'*R{r})./(norm(R{r}))^2));
    end
    % EigenThings
    counter = counter + 1;
    eigs = eig(D)*(12*1.6*10^(-4))/(39.948*1.66*10^(-27));
    freqs{1}(counter) = sqrt(eigs(1))/10^(12);
    freqs{2}(counter) = sqrt(eigs(2))/10^(12);
    freqs{3}(counter) = sqrt(eigs(3))/10^(12);
end

% Density of Modes
figure("Name", 'DOS')
number_of_bins = 50;
bin = linspace(min([freqs{:}]),max([freqs{:}]),number_of_bins+1);
DOS = cell(1,3);
edges = cell(1,3);
```

```
for i = 1:3
[DOS{i},edges{i}] = histcounts(freqs{i},bin);
    C = zeros(1,number_of_bins);
    for j = 1:number_of_bins
        C(j) = (edges{i}(j)+edges{i}(j+1))/2; % Bin centers
    end
    DOS{i} = DOS{i}./(length(freqs{i}))
    plot(C,DOS{i},'LineWidth',1.5); hold on
end
TOTALDOS = DOS{1}+DOS{2}+DOS{3};
plot(C,TOTALDOS,'LineWidth',2);
```

Listing 10: `Matlab` script for FCC thermal expansion

```
%% List of lattice parameters of interest %%
a = linspace(5,5.6,250);

%% Cohesion Energy at different lattice parameters %%
CohU = zeros(1,length(a));
for l = 1:length(a)
    % van der Waals radius (experimental)
    r0 = 3.758/2^(1/6); %\AA
    % Binding Energy
    e = 99.55; % cm^{-1}
    % Symmetry Selection
    sym = 'fcc';
    % Number of Cells
    Nx = 8;
    Ny = 8;
    Nz = 8;
    Nsteps = 11;
    E = zeros(1,Nsteps);
    for i = 1:Nsteps
        [r, Nb, Lx, Ly, Lz] = generateLattice(a(l),Nx,Ny,Nz,sym);
        x = r(1,:); y = r(2,:); z = r(3,:);
        N = length(x)/Nb;
        CohU(l) = cohesionEnergy(x,y,z);
    end
end

%% Density of States, *intensive*, computed at equilibrium %%
DOS = [4.39040000000000e-05,0.000204928000000000,0.000496640000000000,
       0.000952448000000000,0.00154572800000000,0.00228160000000000,
       0.00316736000000000,0.00423910400000000,0.00543219200000000,
       0.00690790400000000,0.00851648000000000,0.0103412480000000,
       0.0124872960000000,0.0148213760000000,0.0174940160000000,
       0.0206319360000000,0.0240764160000000,0.0280175360000000,
       0.0326915840000000,0.0380090880000000,0.0444478720000000,
       0.0524551680000000,0.0626758400000000,0.0786915840000000,
       0.102180224000000,0.105160576000000,0.108192640000000,
       0.110972032000000,0.114103168000000,0.116946304000000,
       0.119946880000000,0.122776576000000,0.125741824000000,
       0.128491648000000,0.110858368000000,0.0751828480000000,
       0.0747947520000000,0.0740446720000000,0.0727651840000000,
       0.0705964800000000,0.0667864320000000,0.0610794240000000,
       0.0498443520000000,0.0729265920000000,0.126208512000000,
       0.152722688000000,0.163704704000000,0.103349376000000,
       0.0681634560000000,0.0318310400000000];

%% Frequency Range for Phonons at different lattice parameters %%
maxFreq = zeros(1,length(a));
minFreq = zeros(1,length(a));
for p = 1:length(a)
    % LJ Deriva(p)tives [ANALYTICAL]
    l = a(p)/sqrt(2);
    dV = 4*e*(6*r0^6/l^7 - 12*r0^12/l^13);
    ddV = 4*e*(156*r0^12/l^14 - 42*r0^6/l^8);
    % Nearest Neighbors List
    R = {[a(p)/2, a(p)/2, 0], [0, a(p)/2, a(p)/2], [a(p)/2, 0, a(p)/2], [-a(p)/2, -a(p)/2,
     0], [0, -a(p)/2, -a(p)/2], [-a(p)/2, 0, -a(p)/2], [a(p)/2, -a(p)/2, 0], [0, a(p)/2, -
    a(p)/2], [a(p)/2, 0, -a(p)/2], [-a(p)/2, a(p)/2, 0], [0, -a(p)/2, a(p)/2], [-a(p)/2,
    0, a(p)/2]};
    % High Symmetry k-points
        G = [0,0,0];
        X = [2*pi/a(p),0,0];
        L = [pi/a(p),pi/a(p),pi/a(p)];
```

```
        W = [2*pi/a(p),pi/a(p),0];
        K = [3/2*pi/a(p),3/2*pi/a(p),0];
    % High Symmetry Paths
    Nfactor = 100;
    Npoints = cell(1,5);
        %G-X
        Npoints{1} = round(Nfactor*norm(G-X));
        t = linspace(0,1,Npoints{1});
        Kx{1} = G(1) + (X(1)-G(1))*t;
        Ky{1} = G(2) + (X(2)-G(2))*t;
        Kz{1} = G(3) + (X(3)-G(3))*t;
        %X-W
        Npoints{2} = round(Nfactor*norm(X-W));
        t = linspace(0,1,Npoints{2});
        Kx{2} = X(1) + (W(1)-X(1))*t;
        Ky{2} = X(2) + (W(2)-X(2))*t;
        Kz{2} = X(3) + (W(3)-X(3))*t;
        %W-K
        Npoints{3} = round(Nfactor*norm(W-K));
        t = linspace(0,1,Npoints{3});
        Kx{3} = W(1) + (K(1)-W(1))*t;
        Ky{3} = W(2) + (K(2)-W(2))*t;
        Kz{3} = W(3) + (K(3)-W(3))*t;
        %K-G
        Npoints{4} = round(Nfactor*norm(K-G));
        t = linspace(0,1,Npoints{4});
        Kx{4} = K(1) + (G(1)-K(1))*t;
        Ky{4} = K(2) + (G(2)-K(2))*t;
        Kz{4} = K(3) + (G(3)-K(3))*t;
        %G-L
        Npoints{5} = round(Nfactor*norm(G-L));
        t = linspace(0,1,Npoints{5});
        Kx{5} = G(1) + (L(1)-G(1))*t;
        Ky{5} = G(2) + (L(2)-G(2))*t;
        Kz{5} = G(3) + (L(3)-G(3))*t;
    freqs = cell(1,3);
    freqs{1} = zeros(1,sum([Npoints{:}]));
    freqs{2} = zeros(1,sum([Npoints{:}]));
    freqs{3} = zeros(1,sum([Npoints{:}]));
    counter = 0;
    for i = 1:5 % Number of HighSym Lines
        kx = Kx{i};
        ky = Ky{i};
        kz = Kz{i};
        for j = 1:Npoints{i}
            k = [kx(j),ky(j),kz(j)];
            % Dynamical Matrix
            A = 2*dV/l;
            B = 2*ddV-A;
            D = zeros(3,3);
            for r = 1:length(R)
                D = D + ((\sin(1/2*k*R{r}'))^2*(A*eye(3) + B*(R{r}'*R{r})./(norm(R{r}))^2)
);
            end
            % EigenThings
            counter = counter + 1;
            eigs = eig(D)*(12*1.6*10^(-4))/(39.948*1.66*10^(-27));
            freqs{1}(counter) = sqrt(eigs(1))/10^(12);
            freqs{2}(counter) = sqrt(eigs(2))/10^(12);
            freqs{3}(counter) = sqrt(eigs(3))/10^(12);
        end
    end
    maxFreq(p) = max(freqs{3});
    minFreq(p) = min(freqs{1});
end

%% Helmholtz Free Energy Computation and Minimization for several a and T values
F = zeros(30,11);
aEQ = zeros(1,30);
for j = 1:53
    T = (j-1)+0.0001; %Kelvin
    for i = 1:length(a)
        w = linspace(minFreq(i)+0.0001,maxFreq(i),length(DOS));
        % hbar set to one
        h = 1;
        % boltzmann set to one
        k = 1;
```

```
        % Helmholtz Free Energy
        F(j,i) = CohU(i) + sum(k*T*log(1-exp(-h*w/(k*T))).*DOS);
    end
    plot(a,F(j,:),'-k'); hold on
    legend off
    aEQ(j) = a(F(j,:) == min(F(j,:)));
    scatter(aEQ(j),min(F(j,:)),5,'x','r','LineWidth',1);
end
xlabel('Lattice Parameter [\AA]', 'Interpreter','tex');
ylabel('Helmholtz Free Energy (Classical) [cm^{-1}]', 'Interpreter','tex');
title('Free Energy Minimization for T in range [0, 52] K', 'Interpreter','tex');
figure("Name",'Thermal Expansion')
T = 0:52+0.0001;
fitted = fit(T',aEQ','poly2');
plot(T,aEQ,'+k'); hold on
T = 0:100;
fitY = fitted.p1*T.^2 + fitted.p2*T + fitted.p3;
plot(T,fitY,'--r','LineWidth',1.2);
ylim([5.1, 5.7]);
yline(5.1628,':g','LineWidth', 1.2);
xlabel('Temperature [K]');
ylabel('Equilibrium Lattice Parameter [\AA]')
title('Thermal Expansion of FCC Solid-Argon within QHLD');
legend('Computed Data', 'Quadratic Fit', 'Zero-T Value');
```