

# Supporting Information - First-principles study of phase equilibrium in Cu-Pd alloy

Feiyang Geng<sup>a</sup>, Jacob R. Boes<sup>a</sup>, John R. Kitchin<sup>a,\*</sup>

<sup>a</sup>*Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213*

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Introduction to the SI . . . . .	4
1.2	Guidance for using raw data . . . . .	4
<b>2</b>	<b>Convergence studies</b>	<b>8</b>
2.1	Convergence of DFT calculation . . . . .	8
2.1.1	DFT calculations in cluster expansion . . . . .	8
2.1.2	Summary of converged energy cutoff and k-points in different functionals (PBE, PBEsol, LDA, AMO5) . . .	9
2.2	Convergence of stiffness of bonds on energy cutoff . . . . .	12
2.3	Convergence of bond-length-dependent force constant on density of k-point grid . . . . .	15
2.4	Choice of ISMEAR in calculation of phonon free energy . . . .	18

---

\*Corresponding author

Email address: [jkitchin@andrew.cmu.edu](mailto:jkitchin@andrew.cmu.edu) (John R. Kitchin)

2.5	Convergence of bond-length-dependent force constant on displacement of atom . . . . .	21
2.5.1	bcc structure . . . . .	21
2.5.2	fcc structure . . . . .	25
2.6	Convergence of bond-length-dependent force constant on super-cell radius . . . . .	28
2.7	Convergence of phase boundary on temperature interval . . .	32
2.8	Convergence of phase boundary on various super-cell size . . .	34
<b>3</b>	<b>How to choose phases considered in phase diagram</b>	<b>35</b>
<b>4</b>	<b>Stretching and bending mode of force constant</b>	<b>37</b>
<b>5</b>	<b>Effect of thermal expansion on phase diagram</b>	<b>38</b>
<b>6</b>	<b>Effect of different functionals on ground state energy</b>	<b>39</b>
<b>7</b>	<b>Effect of error in DFT calculation on phase diagram</b>	<b>41</b>
<b>8</b>	<b>References</b>	<b>42</b>
<b>9</b>	<b>Appendix</b>	<b>43</b>
9.1	Example of k-points convergence in various functionals . . . . .	43
9.1.1	Cu fcc A1 PBE . . . . .	43
9.1.2	Cu fcc A1 PBEsol . . . . .	44
9.1.3	Cu fcc A1 LDA . . . . .	45
9.1.4	Cu fcc A1 AMO5 . . . . .	46
9.2	Examples of encut convergence in various functionals . . . . .	48

9.2.1	Cufcc A1 PBESOL . . . . .	48
9.2.2	Cufcc A1 LDA . . . . .	49
9.2.3	Cufcc A1 AMO5 . . . . .	50
9.2.4	Cufcc A1 PBE . . . . .	51
9.3	Figure 1 result of cluster expansion . . . . .	53
9.4	Convergence of encut for force constant . . . . .	57
9.4.1	Cu-Cu bond . . . . .	57
9.4.2	Cu-Pd bond . . . . .	59
9.4.3	Pd-Pd bond . . . . .	61
9.5	Convergence of er for force constant . . . . .	63
9.5.1	Cu-Cu bond . . . . .	63
9.5.2	Cu-Pd bond . . . . .	65
9.5.3	Pd-Pd bond . . . . .	67
9.6	Convergence of k-points for force constant . . . . .	68
9.6.1	Cu-Cu bond . . . . .	68
9.6.2	Cu-Pd bond . . . . .	71
9.6.3	Pd-Pd bond . . . . .	74
9.7	Convergence of displacement of atom for force constant . . . . .	77
9.7.1	bcc . . . . .	77
9.7.2	fcc . . . . .	85
9.8	SI plot stiffness vs bond length together . . . . .	93
9.9	Figure. 3 plot stiffness of stretching mode vs bond length . . . . .	99
9.10	Figure. 5 plot effect of vibrational free energy . . . . .	103
9.11	SI plot different functionals . . . . .	110
9.12	SI use more accurate force calculation based on Ismear=1 . . . . .	117

9.12.1	Cu-Cu bond . . . . .	117
9.12.2	Cu-Pd bond . . . . .	119
9.12.3	Pd-Pd bond . . . . .	120
9.13	SI Compare convex hull of fcc to bcc . . . . .	122
9.14	Construction of phase diagram . . . . .	123
9.14.1	without vibrational free energy . . . . .	123
9.14.2	with vibrational free energy . . . . .	125
9.14.3	effect of thermal expansion . . . . .	127
9.14.4	effect of temperature interval . . . . .	130
9.14.5	effect of super-cell radius . . . . .	135
9.14.6	effect of error in DFT calculation . . . . .	137

## 1. Introduction

### 1.1. Introduction to the SI

This document contains the supporting data for this work. The figures and corresponding codes to generate figures are provided for reproducibility and transparency. All the calculation results of cluster expansion is based on ATAT package from Alex Van de Walle. Because most of the data is generated through ATAT and stored in a tar.gz file, one can reproduce our results by installing ATAT to use our shared raw data. ATAT and relevant documents can be found at <https://www.brown.edu/Departments/Engineering/Labs/avdw/atat/>.

### 1.2. Guidance for using raw data

There are several parts of the data that is important to reproduce this work. a) data of basic cluster expansion parameters we developed (ECIs,

clusters, structure information). b) Calculation results for vibrational free energy and cluster expansion (ECIs, vibrational free energies). c) The code used in construction of the phase diagram.

In the first step, basic cluster expansion is stored in a zipped file which can be found in <https://github.com/KitchinHUB/kitchingroup-XX>. Please download the "CE-novib.tar.gz" in the github. After downloading the file to a directory, you can extract the file in the following code:

---

```
1 tar -zxvf CE-novib.tar.gz
```

---

There are two folders in the file which represents the cluster expansion of CuPd bcc and fcc lattice.

Note that all the POTCAR and OUTCAR files have been deleted due to license restrictions and to reduce file size. To get further information of the structures, the DFT calculations can be reproduced through ATAT and the vasp.wrap file which contains all the parameter used. Fitting of the cluster expansion does not need any other calculations. The eci.out and clusters.out file contains all the information about the ECIs we use and nbclusters.in is used to reduce several 4-atoms ECIs obtained from MAPS algorithm. The following command will perform the fitting of cluster expansion.

---

```
1 maps -d
```

---

After the CE is ready, a phase diagram without vibrational free energy can be reproduced by the following command.

---

```
1 phb -d1=CuPd_bcc/ -d2=CuPd_fcc/ -gs1=1 -gs2=3 -dT=10 -dx=1e-3 -tstat=3 -er=35 -ltep=5e-3 -keV -o=phb01.out
```

---

Note that the -d1 = ... / is the direction of each cluster expansion. -gs1=1 indicates the first phase is the second ground state (L12 structure) in the cluster expansion of CuPd fcc. -gs2=3 indicates second phase is the fourth ground state (CuPd B2 structure) in the cluster expansion of CuPd bcc. The temperature interval is 10 K, the accuracy of the phase boundary is 0.001. The radius of super-cell is 35 Å. -ltep=5e-3 indicates the Low Temperature Expansion (LTE) should be used instead of MC simulation when its accuracy is better than  $5 \times 10^{-3}$  eV [1, 2].

The output of phase boundaries is stored in phb01.out file. This file contains data like this

```
330 -0.0357044 -0.492946 -0.0208088 -0.119509 -0.118761
340 -0.0357539 -0.491726 -0.0234675 -0.11944 -0.118627
350 -0.0358065 -0.49033 -0.0260033 -0.119358 -0.118502
360 -0.03586 -0.488726 -0.0288523 -0.119259 -0.11836
370 -0.0359151 -0.487051 -0.0316699 -0.119156 -0.118223
380 -0.0359709 -0.437098 -0.0346087 -0.119559 -0.118081
390 -0.0360882 -0.394105 -0.0373727 -0.119112 -0.117949
400 -0.0361653 -0.39122 -0.040336 -0.118867 -0.117809
```

Note that the meaning of six columns above is : Temperature (K), chemical potential, composition of L12 phase (range from -1 to 1), composition of B2 phase (range from -1 to 1), energy of L12 phase and the energy of B2 phase.

It is clear that at 380 K, the composition of L12 phase has a obvious change (from -0.487 at 370 K to -0.437 at 380 K) which indicates a phase

transition from ordered phase to disordered phase. Therefore, L12 structure is replaced with disordered fcc based on following command.

---

```
1 phb -T=380 -mu=-0.0359709 -d1=CuPd_bcc/ -d2=CuPd_fcc/ -gs1=-1 -gs2=3 -dT=10 -dx=1e-3 -tstat=3 -er=35 -lstep=5e-3 -keV -
```

---

The summarized data of phase diagram is stored in "novib-0-0.5.txt" and "novib-0.5-0.75.txt". First and third columns in those txt files are compositions of Pd on the phase boundaries. Second and fourth columns are lists of temperature corresponding to the composition in the same row. More details about the PHb program can be found in Refs. [1](#), [3](#).

In the next step, the vibrational free energy is considered in the cluster expansion. The calculation of the vibrational free energy is based on displacement of the atoms by 0.035 Å for fcc and 0.02 Å for bcc. Details of these calculations can be found in stiffness-fcc.tar.gz and stiffness-bcc-tar.gz files. The following commands can be used to extract the files.

---

```
1 tar -zxf stiffness-fcc.tar.gz
2 tar -zxf stiffness-bcc.tar.gz
```

---

All the fitted result of stiffness versus bond length is stored in the slspring.out file in each of the directories and then used to calculate the vibrational free energy. The calculation result and corresponding cluster expansion is stored in CE-vib-fcc.tar.gz and CE-vib-bcc.tar.gz files, which can be extracted as:

---

```
1 tar -zxf CE-vib-fcc.tar.gz
2 tar -zxf CE-vib-bcc.tar.gz
```

---

Vibrational free energy results are kept in all the fvib file in each structure. The following code is used to calculate vibrational free energy.

---

```
1 cd ~/CuPd_fcc_vib/
2 echo 1000 1001 > Trange.in
3 foreachfile str_relax.out pwd /; svsl -pa -ns=5 -ms=0.01 -sig=6
4 clusterexpand -pa -cv fvib
5 mkteci fvib.eci
```

---

When the CE is ready, users can repeat the previous steps to run the PHb program to create a phase diagram. The calculation result is summaried in a txt file called "dt10-vib.txt" in the same directory as cluster expansion files.

## 2. Convergence studies

### 2.1. Convergence of DFT calculation

#### 2.1.1. DFT calculations in cluster expansion

In the cluster expansion, the PBE functional was used to calculate energies of all the structures. Convergence on energy cutoff for various structures in CuPd bcc and fcc lattice was performed. The energy cutoff changes from 350 eV to 600 eV and number of k-points per reciprocal atom is set to be 2000. An energy cutoff of 500 eV is found to be sufficient to convergence of 1 meV/atom. Convergence on density of  $k$ -points grid was also tested. The number of k-points per reciprocal atom was changed from 1000 to 10000. For structures with 1-8 atoms, 4000 k-points per reciprocal atom is found to be sufficient to converge within 1-2 meV/atom. In cluster expansion, the number of k-points per reciprocal atom is set to be 4000.

### 2.1.2. Summary of converged energy cutoff and $k$ -points in different functionals (PBE, PBESol, LDA, AMO5)

In order to reach convergence of 1 meV/atom, we increase energy cutoff and density of  $k$ -points respectively and the resulting parameters in various functionals are summarized in following tables 1, 2, 3, 4. Several examples that illustrates how we performed those calculations can be found in Appendix part.

Table 1: Parameter sets for converged results on ground state structures with PBE functional (convergence level is 1 meV/atom).

structure	converged encut	converged $k$ -points	density	functional
Cu bcc	500	16×16×16		PBE
Pd bcc	500	16×16×16		PBE
Cu5Pd3 bcc	500	10×10×10		PBE
CuPd bcc	500	10×10×10		PBE
CuPd bcc	500	13×13×13		PBE
Cu6Pd2 bcc	500	10×10×10		PBE
CuPd bcc	500	10×10×10		PBE
CuPd bcc	500	16×16×16		PBE
Pd fcc	500	16×16×16		PBE
Cu3Pd L12	500	13×13×13		PBE
CuPd3 L12	500	13×13×13		PBE
Cu5Pd3 fcc	500	10×10×10		PBE
Cu4Pd4 fcc	500	10×10×10		PBE
Cu3Pd5 fcc	500	10×10×10		PBE

Table 2: Parameter sets for converged results on ground state structures with PBEsol functional (convergence level is 1 meV/atom).

structure	converged encut	converged $k$ -points density	functional
Cu bcc	500	$22 \times 22 \times 22$	PBEsol
Pd bcc	500	$23 \times 23 \times 23$	PBEsol
Cu5Pd3 bcc	500	$14 \times 14 \times 14$	PBEsol
Cu6Pd2 bcc	500	$17 \times 17 \times 17$	PBEsol
CuPd bcc	500	$13 \times 13 \times 13$	PBEsol
Cu fcc	500	$23 \times 23 \times 23$	PBEsol
Pd fcc	500	$24 \times 24 \times 24$	PBEsol
Cu3Pd L12	500	$17 \times 17 \times 17$	PBEsol
CuPd3 L12	500	$19 \times 19 \times 19$	PBEsol
Cu5Pd3 fcc	500	$17 \times 17 \times 17$	PBEsol
Cu4Pd4 fcc	500	$16 \times 16 \times 16$	PBEsol
Cu3Pd5 fcc	500	$14 \times 14 \times 14$	PBEsol

Table 3: Converged study result on ground state structures with AMO5 functional (Convergence level is 1 meV/atom).

structure	converged encut	converged $k$ -points density	functional
Cu bcc	500	$19 \times 19 \times 19$	AMO5
Pd bcc	500	$23 \times 23 \times 23$	AMO5
Cu <sub>5</sub> Pd <sub>3</sub> bcc	500	$14 \times 14 \times 14$	AMO5
Cu <sub>6</sub> Pd <sub>2</sub> bcc	500	$16 \times 16 \times 16$	AMO5
CuPd bcc	500	$13 \times 13 \times 13$	AMO5
Cu fcc	500	$22 \times 22 \times 22$	AMO5
Pd fcc	500	$24 \times 24 \times 24$	AMO5
Cu <sub>3</sub> Pd L1 <sub>2</sub>	500	$20 \times 20 \times 20$	AMO5
CuPd <sub>3</sub> L1 <sub>2</sub>	500	$18 \times 18 \times 18$	AMO5
Cu <sub>5</sub> Pd <sub>3</sub> fcc	500	$15 \times 15 \times 15$	AMO5
Cu <sub>4</sub> Pd <sub>4</sub> fcc	500	$14 \times 14 \times 14$	AMO5
Cu <sub>3</sub> Pd <sub>5</sub> fcc	500	$14 \times 14 \times 14$	AMO5

Table 4: Converged study result on ground state structures with LDA functional (Convergence level is 1 meV/atom).

structure	converged encut	converged $k$ -points density	functional
Cu bcc	500	$24 \times 24 \times 24$	LDA
Pd bcc	500	$23 \times 23 \times 23$	LDA
Cu <sub>5</sub> Pd <sub>3</sub> bcc	500	$19 \times 19 \times 19$	LDA
Cu <sub>6</sub> Pd <sub>2</sub> bcc	500	$16 \times 16 \times 16$	LDA
CuPd bcc	500	$13 \times 13 \times 13$	LDA
Cu fcc	500	$17 \times 17 \times 17$	LDA
Pd fcc	500	$24 \times 24 \times 24$	LDA
Cu <sub>3</sub> Pd L12	500	$17 \times 17 \times 17$	LDA
CuPd <sub>3</sub> L12	500	$19 \times 19 \times 19$	LDA
Cu <sub>5</sub> Pd <sub>3</sub> fcc	500	$13 \times 13 \times 13$	LDA
Cu <sub>4</sub> Pd <sub>4</sub> fcc	500	$15 \times 15 \times 15$	LDA
Cu <sub>3</sub> Pd <sub>5</sub> fcc	500	$15 \times 15 \times 15$	LDA

## 2.2. Convergence of stiffness of bonds on energy cutoff

Before the fitting of stiffness of stretching and bending force constants as function of bond length, a convergence test is needed to determine preferable parameters for static calculation.

We generated super-cell structure of B2 with displacement of perturbed atom for four compositions (Cu, Pd, CuPd, Cu<sub>5</sub>Pd<sub>3</sub>). The displacement is chosen to be 0.02 Å and radius of super-cell is set to be 9 Å which is more than three times the distance between nearest neighbors. The sufficiently large radius is of necessity so as to make sure the element of force constant matrix

falls to negligible value at the boundary of super-cell. The isotropic strain of 1% is applied in order to determine the volume dependent of spring constant. Besides, two volumes are sampled which is the minimum requirement to determine the relationship between bond stiffness and bond length [2].

The energy cutoff increases from 350 to 500 eV and  $k$ -points grid remains at  $7 \times 7 \times 7$ . After all the calculations, the relationship between stiffness and bond length is determined by least-square fitting. Figure 1 describes the relationship between stiffness and bond length for Cu-Cu bond. As energy cutoff increases, the change of stiffness at various bond length is negligible which indicates the convergence of stiffness at energy cutoff of 350 eV.

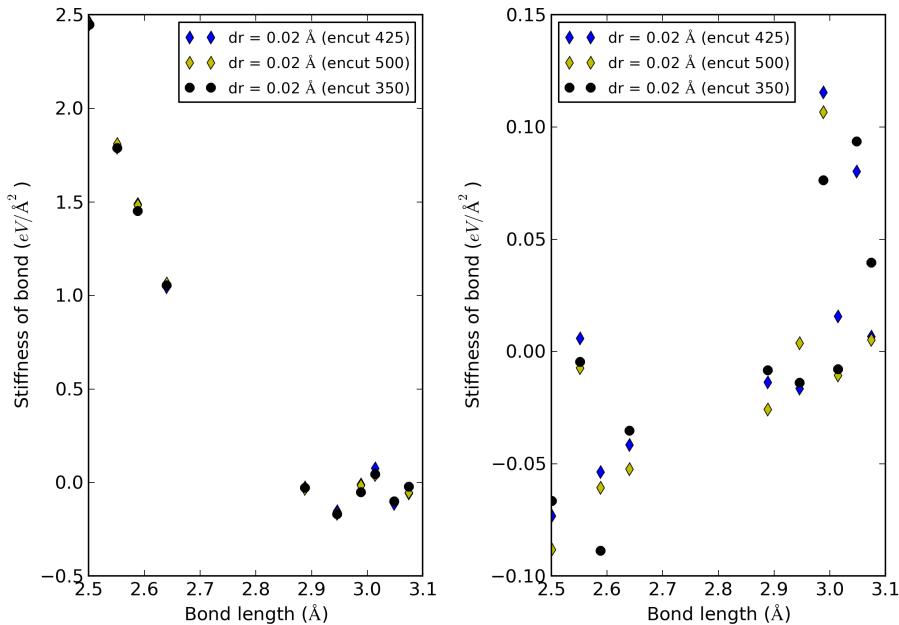


Figure 1: Bond-length-dependent stiffness of Cu-Cu bond with various energy cutoff. The left plot represents Cu-Cu stretching mode and right one represent Cu-Cu bending mode (dr means displacement of perturbed atom).

Figure 2 is the relationship between stiffness and bond length for Cu-Pd bond. It is clear that larger energy cutoff does not make obvious change to the trend of stiffness versus bond length. In order to reduce the computational cost, energy cutoff is chosen to be 350 eV.

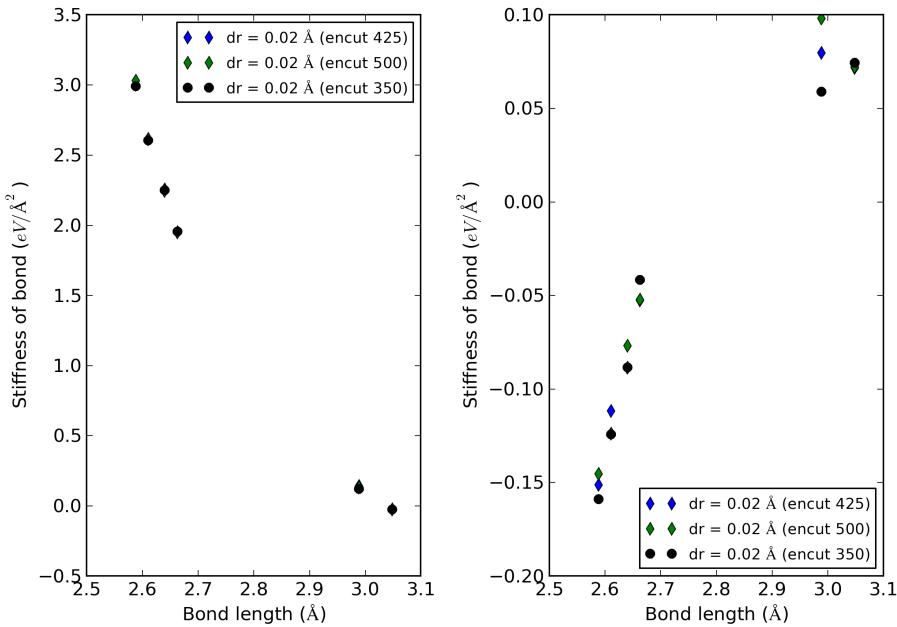


Figure 2: Bond-length-dependent stiffness of Cu-Pd bond with various energy cutoff. The left plot represents Cu-Pd stretching mode and right one represent Cu-Pd bending mode (dr means displacement of perturbed atom).

Figure 3 represent the trend of bond-length-dependent stiffness of Pd-Pd bond as a function of energy cutoff. Although energy cutoff changes from 350 to 500 eV, the change of stiffness for both stretching and bending mode is within  $0.02 \text{ eV}/\text{\AA}^2$ . Therefore, energy cutoff of 350 eV is sufficient for convergence of stiffness within  $0.02 \text{ eV}/\text{\AA}^2$ .

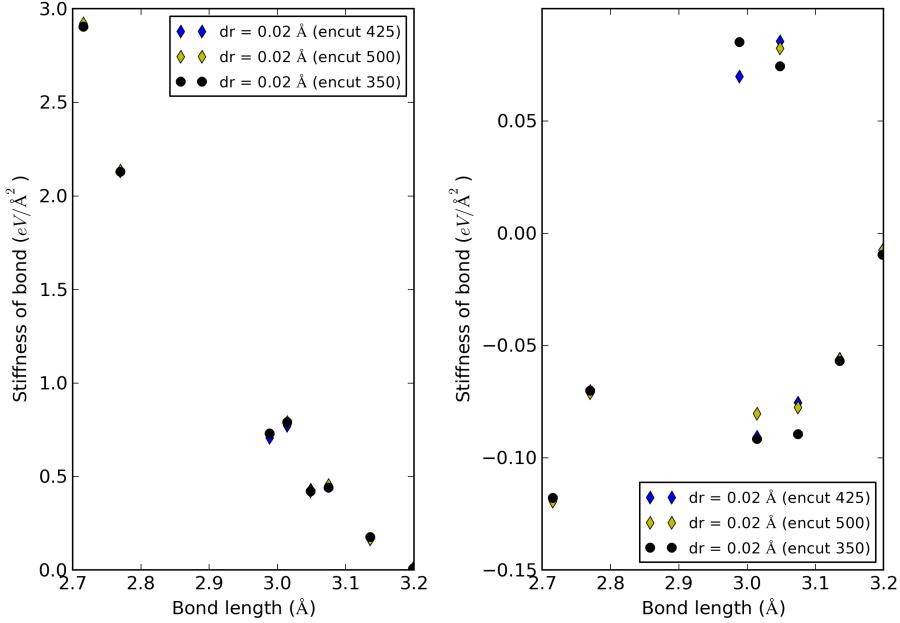


Figure 3: Bond-length-dependent stiffness of Pd-Pd bond with various energy cutoff. The left plot represents Pd-Pd stretching mode and right one represent Pd-Pd bending mode (dr means displacement of perturbed atom).

### 2.3. Convergence of bond-length-dependent force constant on density of $k$ -point grid

In this section, we generated super-cell structure of B2 with displacement of perturbed atom for four compositions (Cu, Pd, CuPd, Cu<sub>5</sub>Pd<sub>3</sub>) of bcc lattice. The displacement is chosen to be 0.02 Å and radius of super-cell-is set to be 9 Å which is more than three times the distance between nearest neighbors. The sufficiently large radius is of necessity so as to make sure the element of force constant matrix falls to negligible value at the boundary of super-cell. The isotropic strain of 1% is applied in order to determine the volume dependent of spring constant. This parameter is important for use

of quasi-harmonic approximation to calculate vibrational free energy that accounts for thermal expansion.

Density of k-points mesh increases from  $5 \times 5 \times 5$  to  $10 \times 10 \times 10$  and energy cutoff is fixed to 350 eV. After all the calculation is finished, the stiffness versus bond length is fitted automatically. From the figure 4, it is obvious that density of k-points mesh significantly influences the stiffness of Cu-Cu bond at various bond length and the stiffness will converge to less than  $0.1 \text{ eV}/\text{\AA}^2$  when the density of  $k$ -points mesh is greater than  $7 \times 7 \times 7$ .

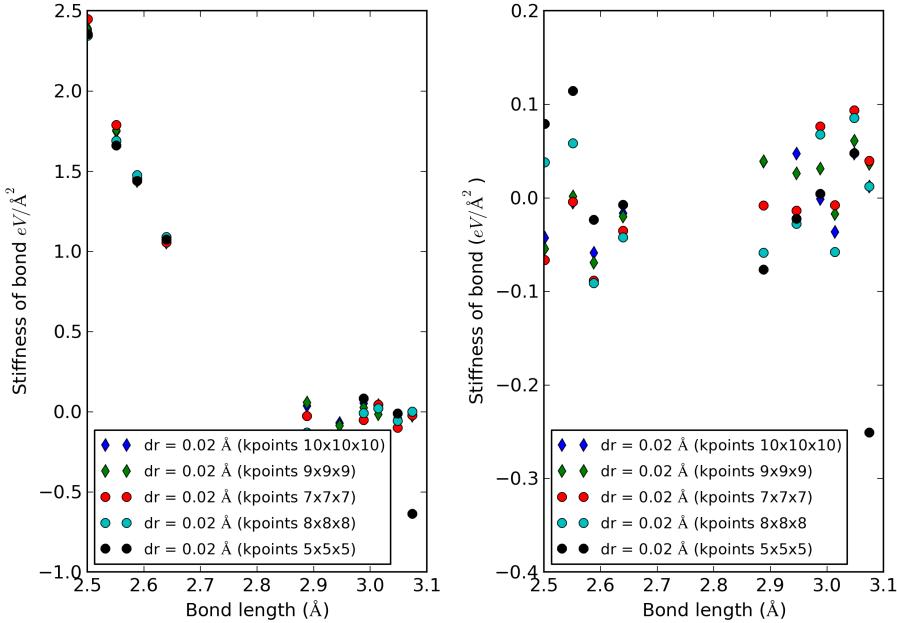


Figure 4: Bond-length-dependent stiffness of Cu-Cu bond with various density of K-points grid. The left plot represents Cu-Cu stretching mode and right one represent Cu-Cu bending mode (dr means displacement of perturbed atom).

Figure 5 represent the trend of bond-length dependent stiffness of Cu-Pd bond. As seen in left plot of figure 5, stiffness of stretching mode shows

very small change as density of k-points grids increases. As for the bending mode, the stiffness of short bond length (shorter than  $2.8 \text{ \AA}$ ) does not show significant increase in higher k-points. At longer bond length (longer than  $2.8 \text{ \AA}$ ), the change in stiffness is noticeable, but converges at k-points mesh density of  $7 \times 7 \times 7$ .

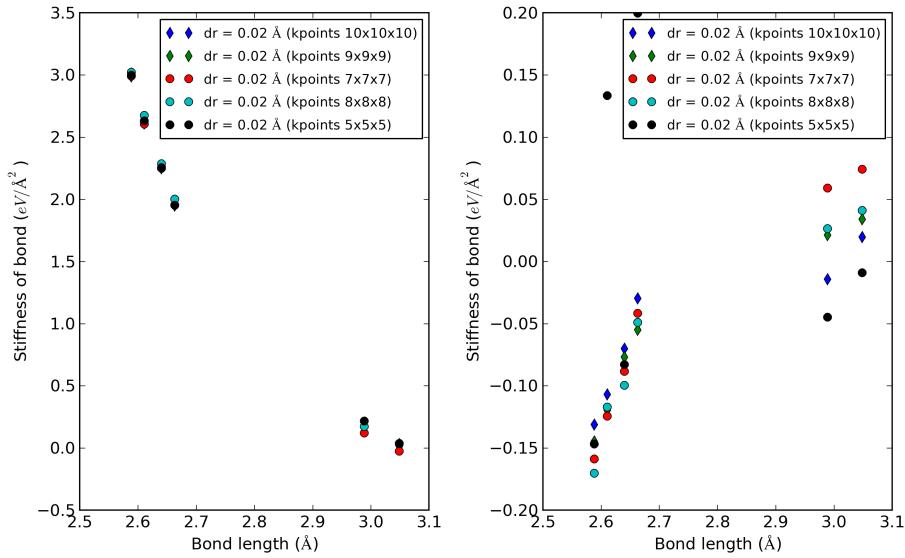


Figure 5: Bond-length-dependent stiffness of Cu-Pd bond with various density of k-points grid. The left plot represents Cu-Pd stretching mode and right one represent Cu-Pd bending mode (dr means displacement of perturbed atom).

Figure 6 shows the bond-length dependent stiffness of Pd-Pd bond. As shown in left plot of figure 6, the stiffness of stretching mode of Pd-Pd bond reaches convergence of  $0.05 \text{ eV}/\text{\AA}^2$  at k-points mesh density of  $7 \times 7 \times 7$ . On the right plot of figure 6, stiffness of bending mode shows small changes when density of k-points grids is larger than  $7 \times 7 \times 7$ .

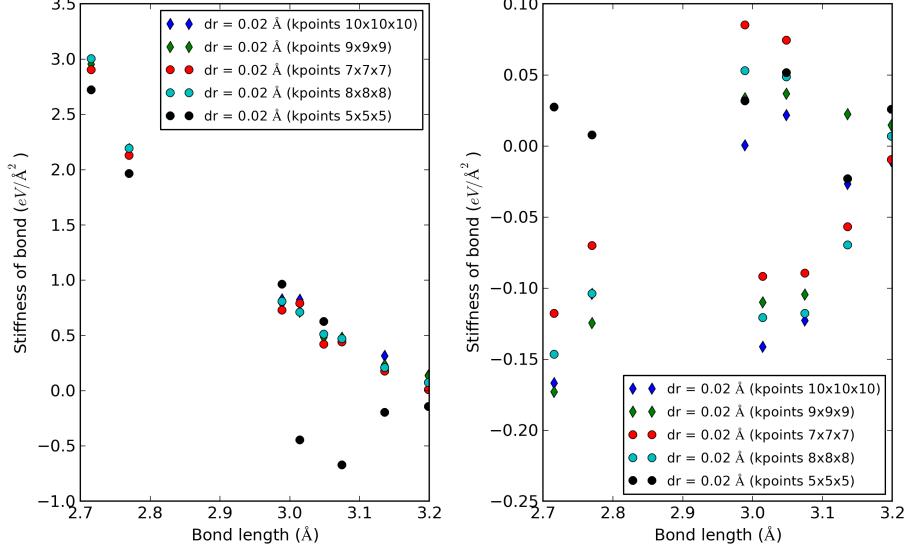


Figure 6: Bond-length-dependent stiffness of Pd-Pd bond with various k-points. The left plot represents Pd-Pd stretching mode and right one represent Pd-Pd bending mode (dr means displacement of perturbed atom).

In our final result, all the calculations are performed on k-point mesh that is denser than  $9 \times 9 \times 9$  in order to reach sufficient accuracy.

#### 2.4. Choice of ISMEAR in calculation of phonon free energy

ISMEAR determines how the partial occupancies are set for each wave function. We used to utilize tetrahedron method with Blöchl corrections (ISMEAR=-5). However, according to VASP manual, the only drawback is that the methods is not variational with respect to the partial occupancies, therefore the calculated forces might be wrong by a few percent for metals. In this section, method of Methfessel-Paxton (ISMEAR=1) is used and the calculation based on energy cutoff of 350 eV and density of k-points mesh of  $9 \times 9 \times 9$  is reproduced. The result is shown in figure 7, 8 and 9.

Figure 7 shows the obtained trend of stiffness versus bond length. As is shown in the figure 7, the stiffness of Cu-Cu bond turn to positive value at bond length of around 2.9 Å when method of Methfessel-Paxton, rather than tetrahedron method with Blöchl corrections, is utilized. This indicates the error of force calculation is the source of wrong trend of stiffness versus bond length. For the right plot of figure 7, the change in stiffness of bending mode is irregular and negligible.

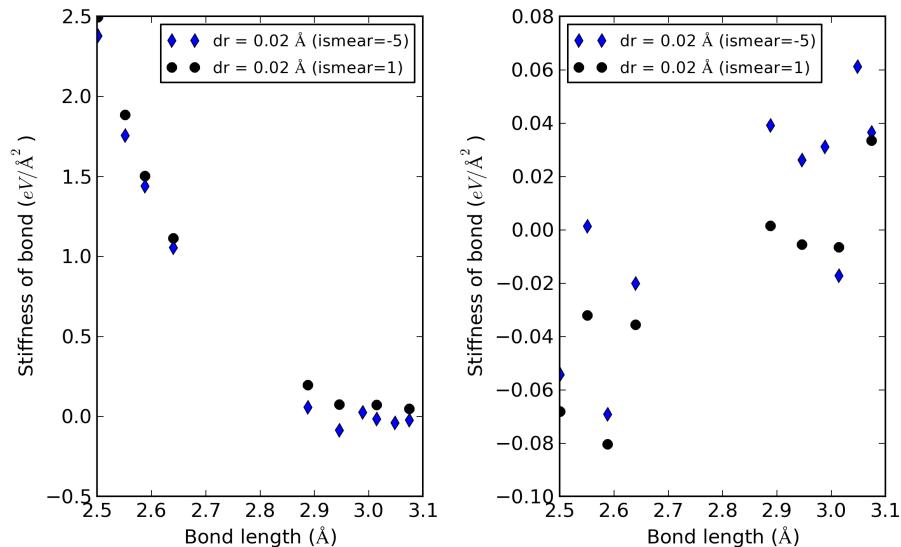


Figure 7: Bond-length-dependent stiffness of Cu-Cu bond with various ismear method. The left plot represents Cu-Cu stretching mode and right one represent Cu-Cu bending mode (dr means displacement of perturbed atom).

Figure 8 shows the stiffness of Cu-Pd bond as a function of different ISMEAR. When method of Methfessel-Paxton (ISMEAR=1), rather than tetrahedron method with Blöchl corrections (ISMEAR=-5), is utilized, the resulting stiffness is really close to that of tetrahedron method. The result

of Cu-Pd bond vibration shows ISMEAR does not make obvious changes to stiffness of Cu-Pd bond, in both bending (the right plot) and stretching mode (the left plot).

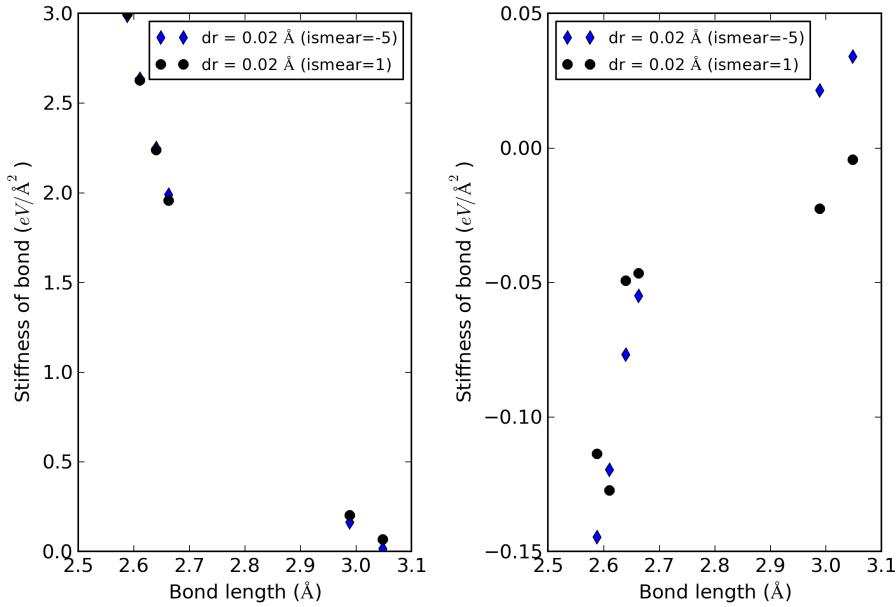


Figure 8: Bond-length-dependent stiffness of Cu-Pd bond with various ismear method. The left plot represents Cu-Pd stretching mode and right one represent Cu-Pd bending mode (dr means displacement of perturbed atom).

Figure 9 is the result of stiffness of Pd-Pd bond versus bond length. As method of Methfessel-Paxton (ISMEAR = 1), rather than tetrahedron method with Blöchl corrections (ISMEAR=-5), is utilized, the stiffness of stretching mode of Pd-Pd bond increases a little, as shown in left plot of figure 9. In comparison, the change of stiffness in bending mode seems random and largest difference is about  $0.05 \text{ eV/}\text{\AA}^2$ , as shown in the right plot of figure ??.

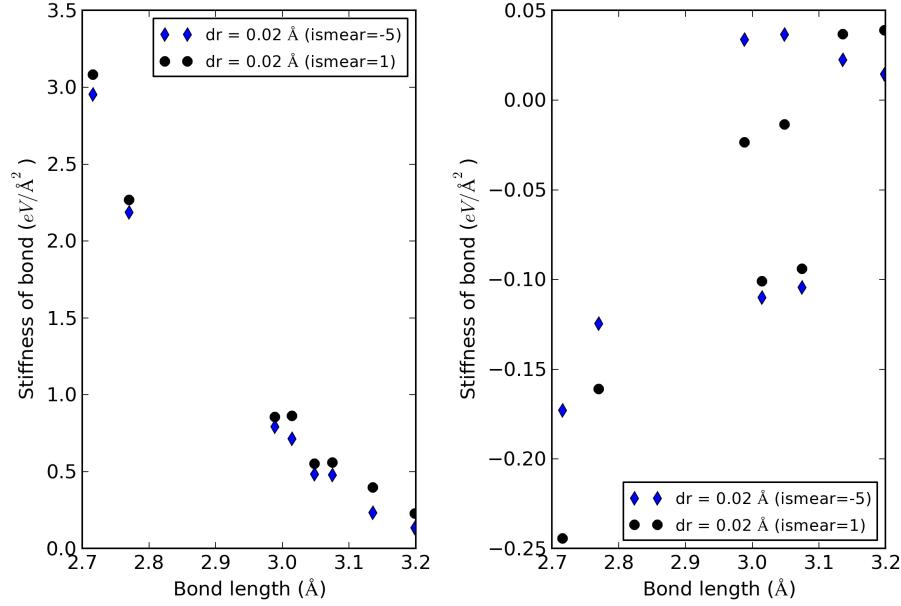


Figure 9: Bond-length-dependent stiffness of Pd-Pd bond with various ismear method. The left plot represents Pd-Pd stretching mode and right one represent Pd-Pd bending mode (dr means displacement of perturbed atom).

In conclusion, method of Methfessel-Pexton is found to be important for a correct value of force. The trend of stiffness will be well above zero as this method is utilized. In our final result, we use ISMEAR = 1 for all the accurate calculations.

## 2.5. Convergence of bond-length-dependent force constant on displacement of atom

### 2.5.1. bcc structure

We generated super-cell structures of B2 with displacement of perturbed atom for five compositions (Cu, Pd, CuPd, Cu<sub>5</sub>Pd<sub>3</sub>, Cu<sub>3</sub>Pd). The displacement is chosen from 0.005 to 0.04 Å and radius of super-cell-is set to be 9 Å

which is more than three times the distance between nearest neighbors. The sufficiently large radius is of necessity so as to make sure the element of force constant matrix falls to negligible value at the boundary of super-cell. The isotropic strain of 1% is applied in order to determine the volume dependent of spring constant. Besides, energy cutoff is 350 eV and k points density is fixed to  $10 \times 10 \times 10$ . After all the calculations are finished, the stiffness versus bond length is fitted. The figure 10, 11, 12 are the relationship between stiffness and bond length for Cu-Cu bond, Cu-Pd bond and Pd-Pd bond respectively.

As for stretching mode of Cu-Cu bond, increasing displacement of atom from 0.005 to 0.04 Å does not change the stiffness, as shown in left panel of figure 10. Comparably, right panel of figure 10 indicates the stiffness of bending mode is influenced by various displacements of atom. However, the dependence on displacement of atom is irregular and changes of stiffness is within  $0.04 \text{ eV}/\text{\AA}^2$ . Also most of points are below zero which is not quiet reasonable. The reason behind this observation is not yet known, but based on convergence of stiffness in stretching mode, we can conclude that displacement from 0.005 to 0.04 Å does not apparently change the stiffness of bond and therefore, the vibration mode of system is harmonic with appropriate displacement of atom.

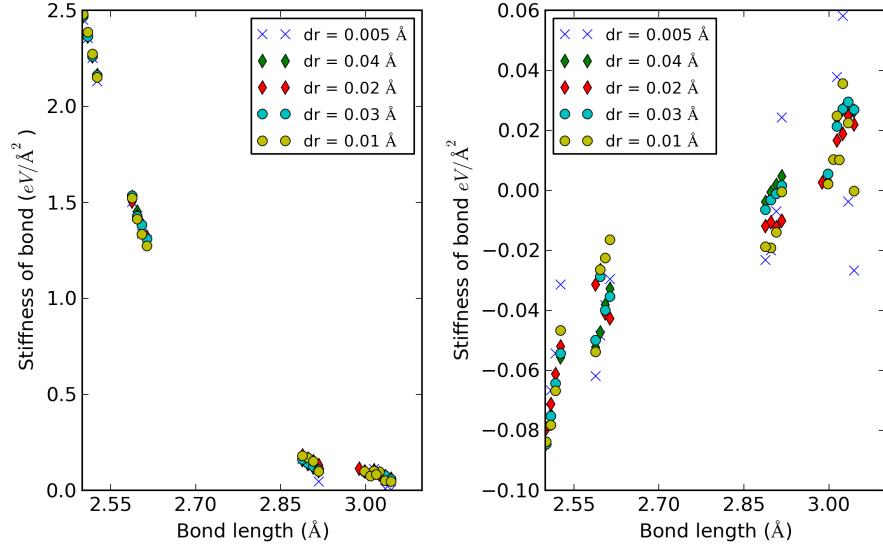


Figure 10: Bond-length-dependent stiffness of Cu-Cu bond with various displacements of atoms. The left plot represents Cu-Cu stretching mode and right one represent Cu-Cu bending mode (dr means displacement of perturbed atom).

As for stretching mode of Cu-Pd bond, increasing displacement of atom from 0.005 to 0.04 Å does not change the stiffness, as shown in left panel of figure 11. Comparably, right panel of figure 11 indicates the stiffness of bending mode does not depend on displacement of atom. Also most of points are below zero which is not quiet reasonable. Based on convergence of stiffness in stretching mode, we can conclude that displacement from 0.005 to 0.04 Å does not apparently change the stiffness of Cu-Pd bond and therefore, the displacement of atom within this range will lead to a harmonic vibration.

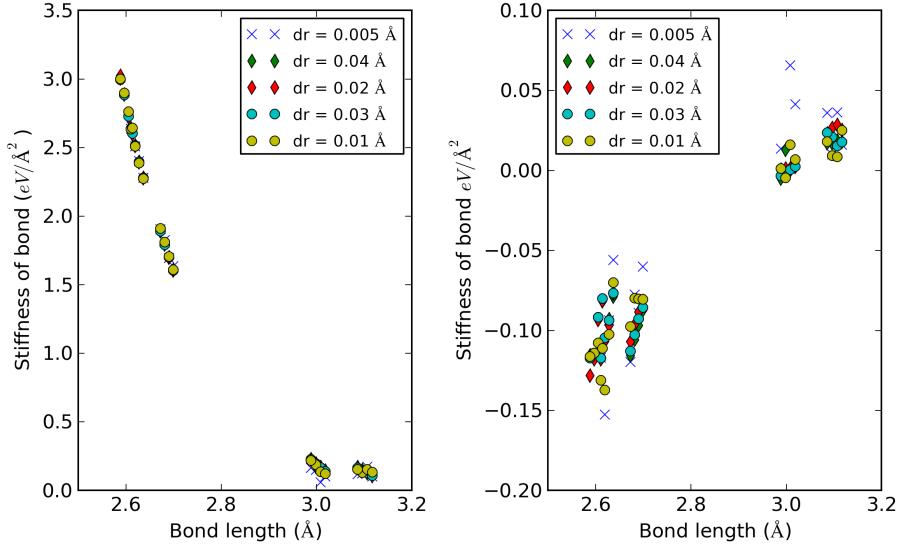


Figure 11: Bond-length-dependent stiffness of Cu-Pd bond with various displacements of atoms. The left plot represents Cu-Pd stretching mode and right one represent Cu-Pd bending mode (dr means displacement of perturbed atom).

As for stretching mode of Pd-Pd bond, increasing displacement of atom from 0.005 to 0.04  $\text{\AA}$  does not change the stiffness, as shown in left panel of figure 12. Comparably, right panel of figure 12 indicates the stiffness of bending mode shows no dependence on displacement of atom at short bond length ( $\approx 2.9 \text{\AA}$ ). As bond length increases, the dependence of displacement of atom is not significant and changes of stiffness is within  $0.05 \text{ eV}/\text{\AA}^2$ . Another important feature of right part of figure 12 is that most of points are below zero which is not quiet reasonable. The reason behind this observation is not yet known, but based on convergence of stiffness in stretching mode, we can conclude that displacement from 0.005 to 0.04  $\text{\AA}$  does not apparently change the stiffness of Pd-Pd bond and therefore, the displacement of atom within

this range will result in harmonic vibration.

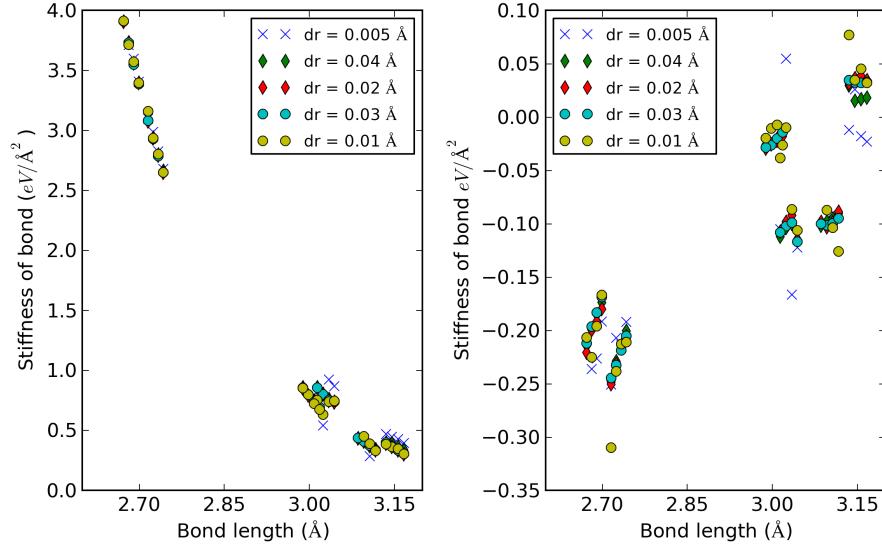


Figure 12: Bond-length-dependent stiffness of Pd-Pd bond with various displacements of atoms. The left plot represents Pd-Pd stretching mode and right one represent Pd-Pd bending mode (dr means displacement of perturbed atom).

### 2.5.2. fcc structure

We generate super-cell structure of fcc with displacement of perturbed atom for four compositions (Cu, Pd, Cu<sub>3</sub>Pd, CuPd<sub>3</sub>). The displacement is chosen from 0.02 to 0.2 Å and radius of super-cell-is set to be 8.4 Å which is more than three times the distance between nearest neighbors. The sufficiently large radius is of necessity so as to make sure the element of force constant matrix falls to negligible value at the boundary of super-cell. The isotropic strain of 1% is applied in order to determine the volume dependent of spring constant. This parameter is important for use of quasi-harmonic approximation to calculate vibrational free energy that accounts for thermal

expansion. Besides, two volumes are sampled which is the minimum requirement to determine the relationship between bond stiffness and bond length. Besides, energy cutoff is 350 eV and  $k$ -points density is fixed to  $9 \times 9 \times 9$ . After all the calculations are finished, the stiffness versus bond length is fitted. The figure 13, 14, 15 are the relationship between stiffness and bond length for Cu-Cu bond, Cu-Pd bond and Pd-Pd bond respectively.

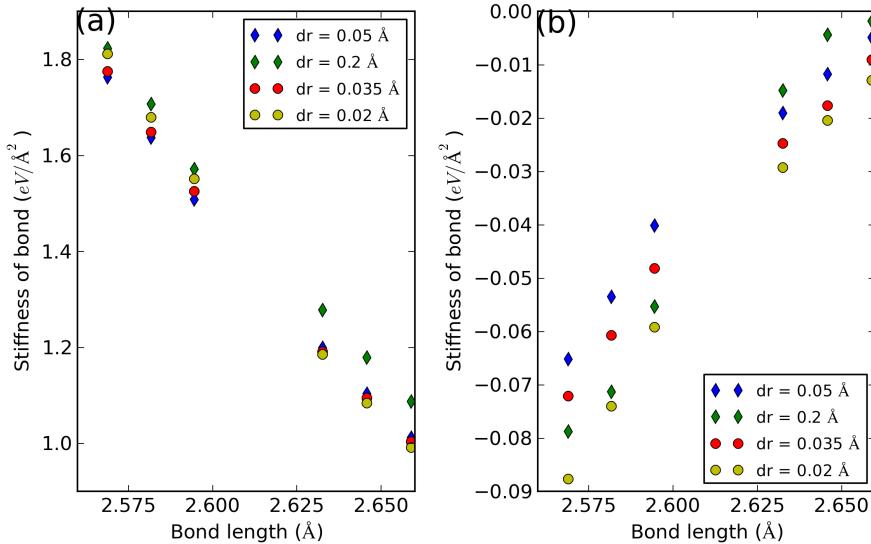


Figure 13: Bond-length-dependent stiffness of Cu-Cu bond with various displacements of atoms. The left plot represents Cu-Cu stretching mode and right one represent Cu-Cu bending mode (dr means displacement of perturbed atom).

As for stretching mode of Cu-Cu bond, increasing displacement of atom from  $0.02$  to  $0.2$   $\text{\AA}$  cause a small disparity of stiffness (less than  $0.05$   $eV/\text{\AA}^2$ ), as shown in left panel of figure 13. The stiffness does not seem to converge as displacement decreases from  $0.035$  to  $0.02$   $\text{\AA}$ . The resulting stiffnesses of displacement of  $0.035$  and  $0.05$   $\text{\AA}$  are quite consistent with each other.

Therefore, the only conclusion we can reach is that stiffness converges to  $0.04 \text{ eV}/\text{\AA}^2$  at displacement of  $0.035 \text{ \AA}$ . Comparably, right panel of figure 13 indicates the stiffness of bending mode is influenced by various displacements of atom. However, most of points are so approaching to zero that their value can be negligible compared with stretching mode. Therefore, we ignore the bending mode and care about stretching term of stiffness.

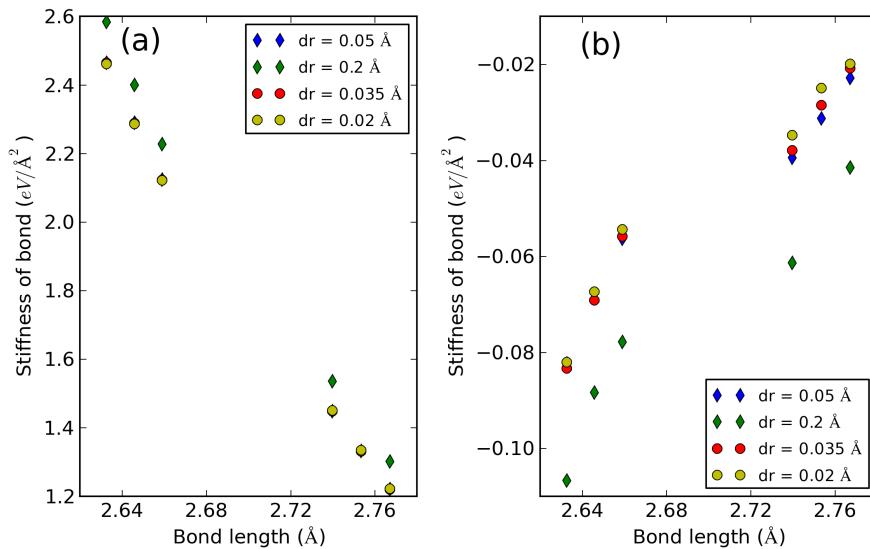


Figure 14: Bond-length-dependent stiffness of Cu-Pd bond with various energy cutoff. The left plot represents Cu-Pd stretching mode and right one represent Cu-Pd bending mode ( $dr$  means displacement of perturbed atom).

As for stretching mode of Cu-Pd bond, increasing displacement of atom from  $0.02$  to  $0.05 \text{ \AA}$  cause negligible disparity of stiffness, as shown in left panel of figure 14. Therefore, displacement of  $0.035 \text{ \AA}$  is appropriate for further calculation. For the bending mode in right part of figure 14, all the values are negative and we ignore their contribution to vibrational free

energy.

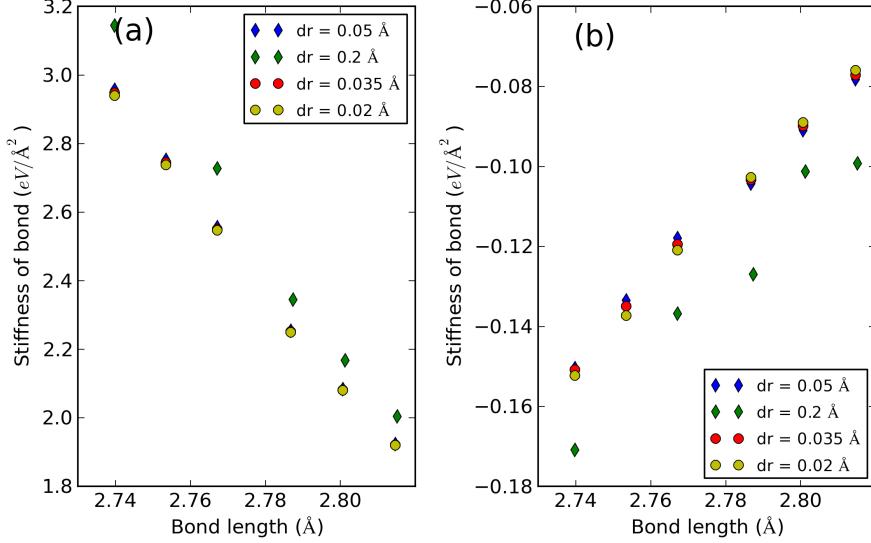


Figure 15: Bond-length-dependent stiffness of Pd-Pd bond with various energy cutoff. The left plot represents Pd-Pd stretching mode and right one represent Pd-Pd bending mode (dr means displacement of perturbed atom).

As for stretching mode of Pd-Pd bond, increasing displacement of atom from 0.02 to 0.05 Å does not change the stiffness, as shown in left panel of figure 15. The result indicates a convergence of stiffness as displacement is lower than 0.05 Å. Comparably, right panel of figure 15 also indicates the displacement lower than 0.05 Å can result in convergence of stiffness of bending mode.

## 2.6. Convergence of bond-length-dependent force constant on super-cell radius

Super-cell size is very important for correct bond-length-dependent force constant. Because sufficiently large radius is able to make sure the element

of force constant matrix falls to negligible value at the boundary of super-cell [4]. The recommended super-cell radius is 3-4 times nearest neighbor distance [2]. There we test force constant with super-cell radius of 7 and 8.4 Å. The displacement of atom is 0.035 Å and structures considered are Cu fcc, Pd fcc, Cu<sub>3</sub>Pd L1<sub>2</sub> and CuPd<sub>3</sub> L1<sub>2</sub> structure. Converged energy cutoff is 350 eV and converged density of  $k$ -point mesh is 9×9×9-10×10×10. Method of Methfessel-Paxton (ISMEAR=1) is used for all the calculations. From the left panel of figure 16, 17 and 18 below, it is very clear that the resulting stretching term of force constant at a super-cell radius of 8.4 Å is very close to that at super-cell radius of 7 Å. The bending terms of vibration in right panel of figure 16, 17 and 18 also show no obvious change as super-cell radius enlarges. Therefore, the super-cell radius of 7-8.4 Å is sufficient to obtain accurate force constant.

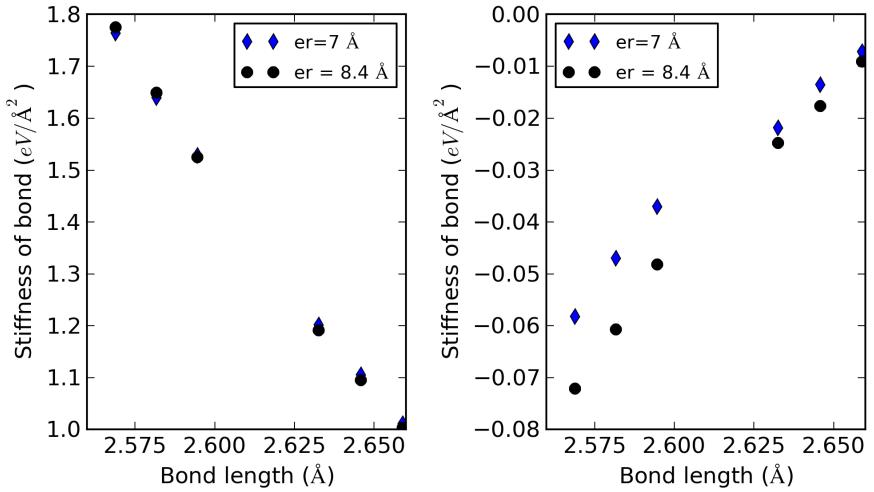


Figure 16: Left panel: streching term of bond-length-dependent force constant of Cu-Cu bond as a function of super-cell radius. Right panel: Bending term of bond-length-dependent force constant of Cu-Cu bond as a function of super-cell radius.

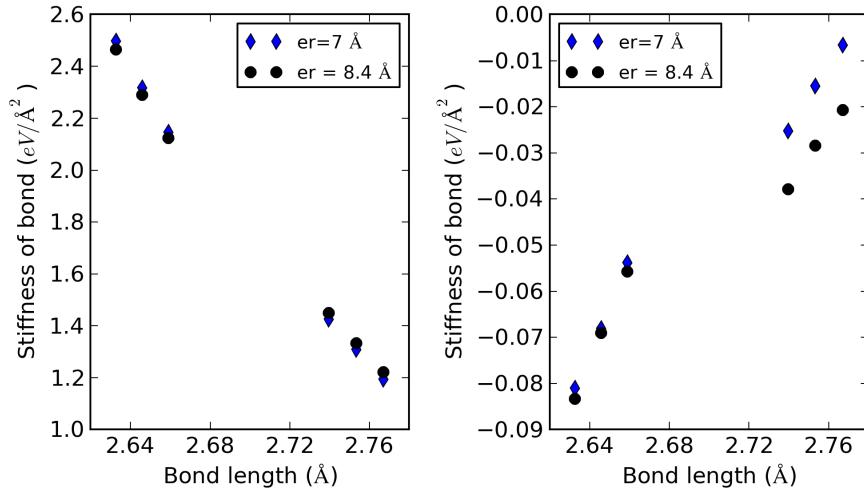


Figure 17: Left panel: Stretching term of bond-length-dependent force constant of Cu-Pd bond as a function of super-cell radius. Right panel: Bending term of bond-length-dependent force constant of Cu-Pd bond as a function of super-cell radius.

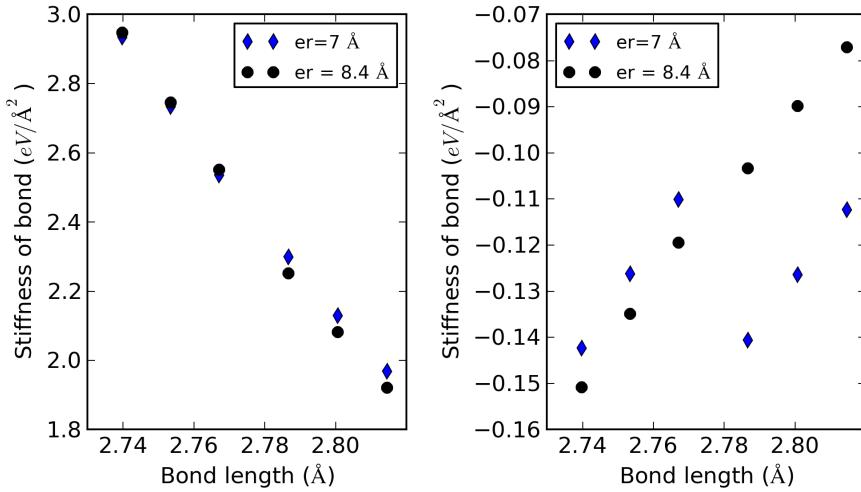


Figure 18: Left panel: Stretching term of bond-length-dependent force constant of Pd-Pd bond as a function of super-cell radius. Right panel: Bending term of bond-length dependent force constant of Pd-Pd bond as a function of super-cell radius.

### 2.7. Convergence of phase boundary on temperature interval

Figure 20 and figure 19 are the result of temperature interval convergence of phase diagram. The temperature interval changes from 25 K to 5 K per step. The thermal expansion is 1% and super-cell radius is 35  $\text{\AA}$ . It is clear that in both diagrams, the phase boundaries converge as temperature interval is smaller than 25 K.

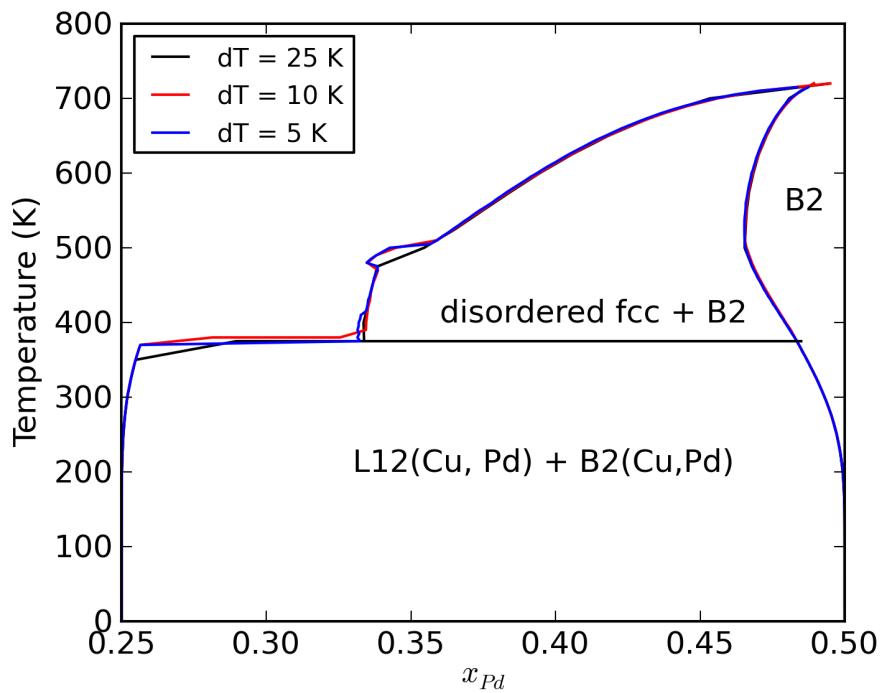


Figure 19: Convergence of temperature interval in phase boundary without considering vibrational free energy.

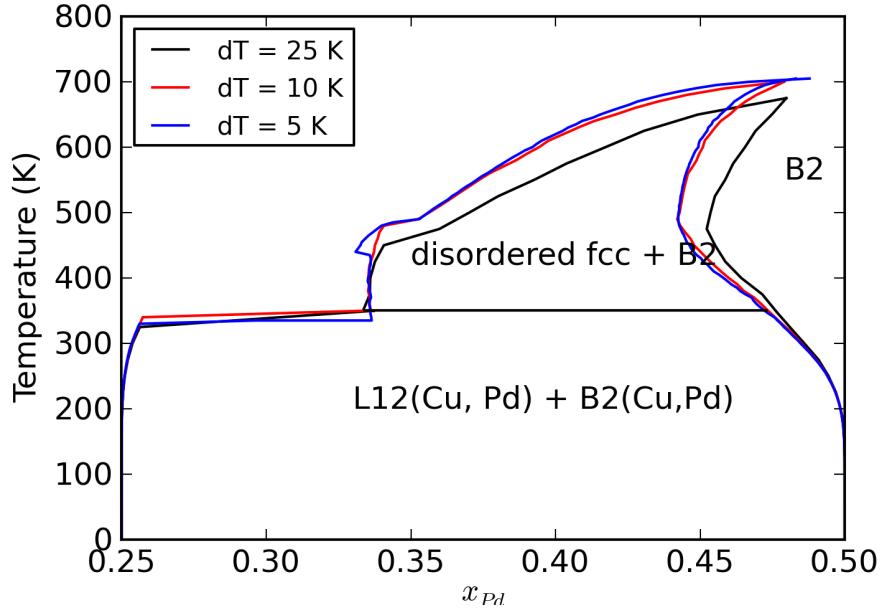


Figure 20: Temperature interval convergence of Phase diagram between L12 and B2 phases considering vibrational free energy.

### 2.8. Convergence of phase boundary on various super-cell size

Super-cell radius is evaluated to ensure the current super-cell size converge. Meanwhile, temperature interval is 10 K per step and other parameters are equal. As is shown in figure , as super-cell radius increases from 35 to 50 Å, the main feature of phase boundary does not change and order-disorder transition temperature only shift around 10K. Therefore, it is convincing that phase boundary converges at super-cell radius of 35 Å.

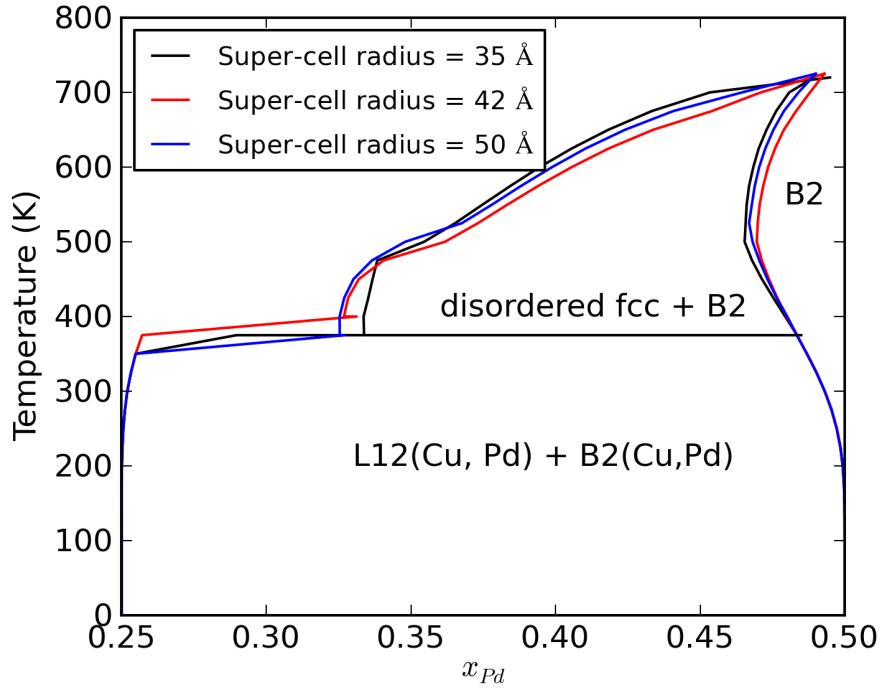


Figure 21: Effect of super-cell radius on phase boundary without considering vibrational free energy.

### 3. How to choose phases considered in phase diagram

The convex hull for both CuPd fcc and bcc lattice is compared in figure 22.

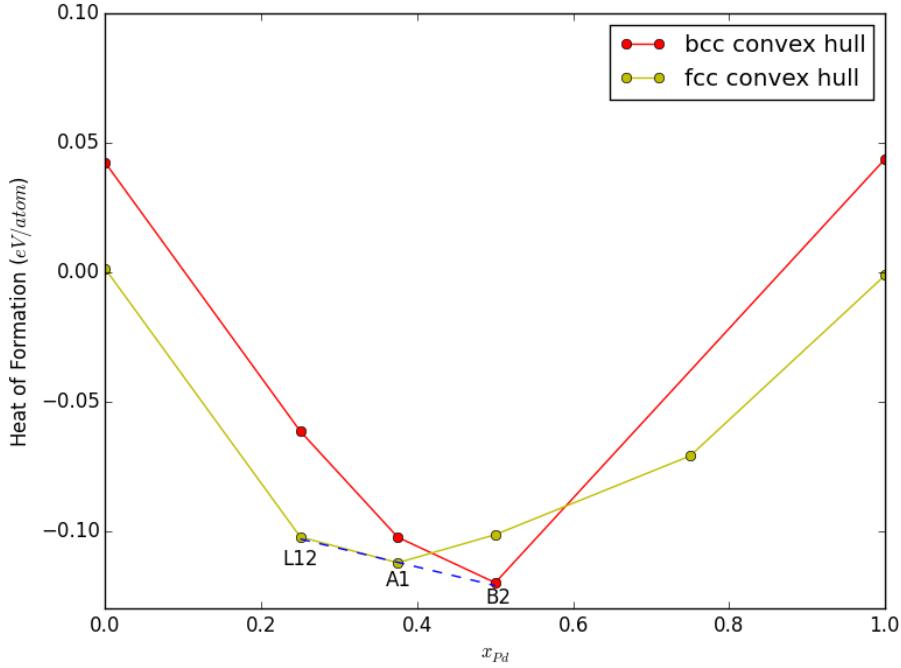


Figure 22: Ground state convex hull of CuPd bcc and fcc lattice. The blue dash line indicates the phase right on the line can be as stable as mixture of L12 phase and B2 phase.

As shown in Figure 22, the formation energy of  $\text{Cu}_5\text{Pd}_3$  A1 structure is right on the line between L12 structure and CuPd B2 structure. This indicates the  $\text{Cu}_5\text{Pd}_3$  A1 structure is no more stable than the mixture of L12 structure and CuPd B2 structure in 0 K. In previous literature [5, 6], disordered A1 phase only exists at temperature higher than 870 K. At lower temperature, L12 phase and B2 phase coexists. Therefore, we neglect the  $\text{Cu}_5\text{Pd}_3$  A1 phase at 0K and only compute the phase equilibrium between L12 and B2 structures in Cu-rich region at low temperature.

#### 4. Stretching and bending mode of force constant

In this section, a complete data of stiffness versus bond length is represented for both stretching and bending mode of vibration.

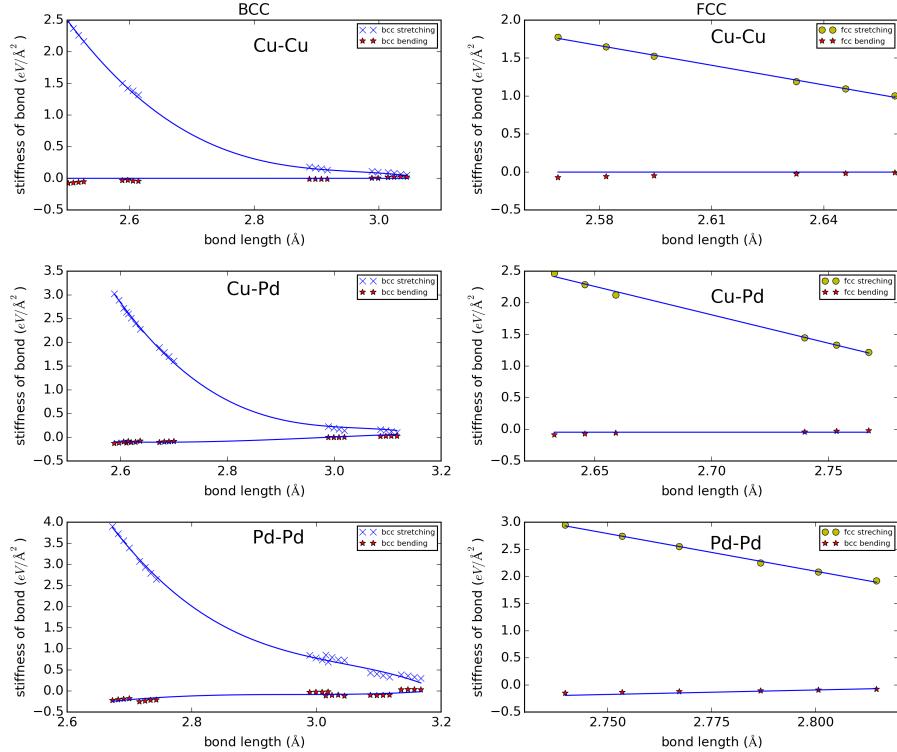


Figure 23: Nearest neighbor stretching and bending force constants versus bond length in bcc and fcc Cu-Pd system (Left three figures are in bcc lattice and three figures on the right are in fcc lattice).

Figure 23 shows stretching constants decrease monotonically with increasing bond length in both fcc and bcc lattice. Comparably, bending terms, by contrast, are relatively insensitive and magnitude is much lower than that of stretching term. In this case, only stretching constants are considered in our

calculation of vibrational free energy.

## 5. Effect of thermal expansion on phase diagram

Thermal expansion is considered in calculation of vibration free energy. The vibration free energies of both CuPd fcc and B2 phases are calculated at 0%, 1% and 2% volume increase. Super-cell radius is chosen to be 35 Å and temperature interval is 10 K per step. As is shown in figure 24, the thermal expansion has apparent influence on phase boundary at high temperature. large thermal expansion seems cause decreasing in order-disorder transition temperature and reduce the region of B2 phase. It indicates thermal expansion add up to the instability of B2 phase. This phenomenon can be explained by the changes of vibrational entropy due to the anharmonicity resulted from volume effect [7]. On the other hand, as thermal expansion is considered, stiffness of all bonds decreases as volume expands which may cause instability of bcc lattice as the stiffness of longer bond approaching to zero. Although thermal expansion cause noticeable change to phase diagram, this change in B2 region is still insufficient to reproduce the experimental phase diagram. But it is also important to note that volume expansion set in out calculation may be different from that in real world and it may be too difficult to evaluate its true effect unless the experimental data of thermal expansion is obtained.

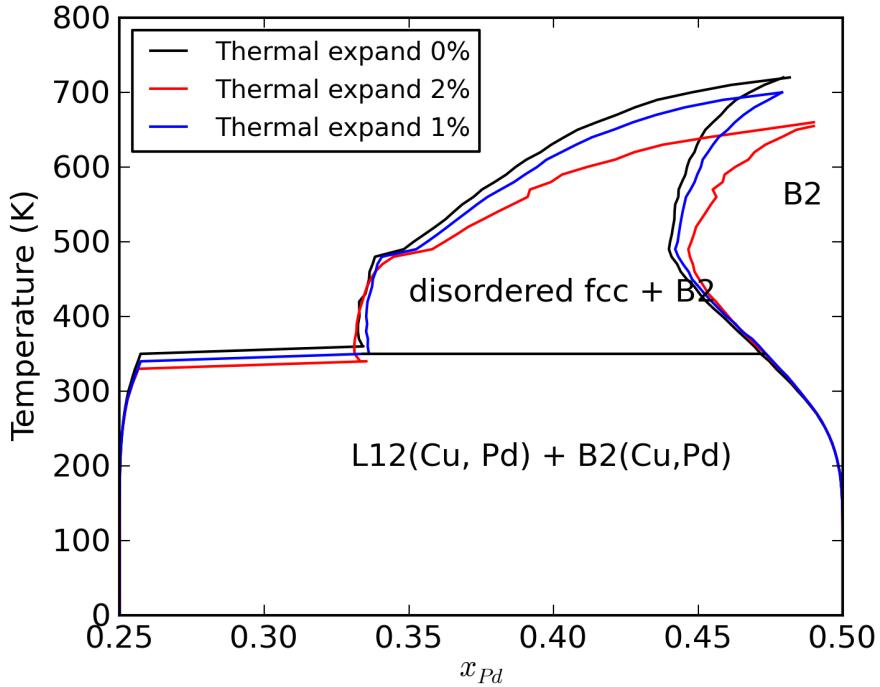


Figure 24: Phase diagram between L12 and B2 phases with different volume increase.

## 6. Effect of different functionals on ground state energy

Different functionals may influence the energy of ground state structures. Calculations based on various functionals were performed on most of the known ground state structures. For bcc, the structures are: Cu,  $\text{Cu}_6\text{Pd}_2$ ,  $\text{Cu}_5\text{Pd}_3$ , CuPd, Pd. As for fcc, structures are: Cu,  $\text{Cu}_3\text{Pd}$ ,  $\text{Cu}_5\text{Pd}_3$ ,  $\text{Cu}_4\text{Pd}_4$ ,  $\text{Cu}_3\text{Pd}_5$ ,  $\text{CuPd}_3$  and Pd.

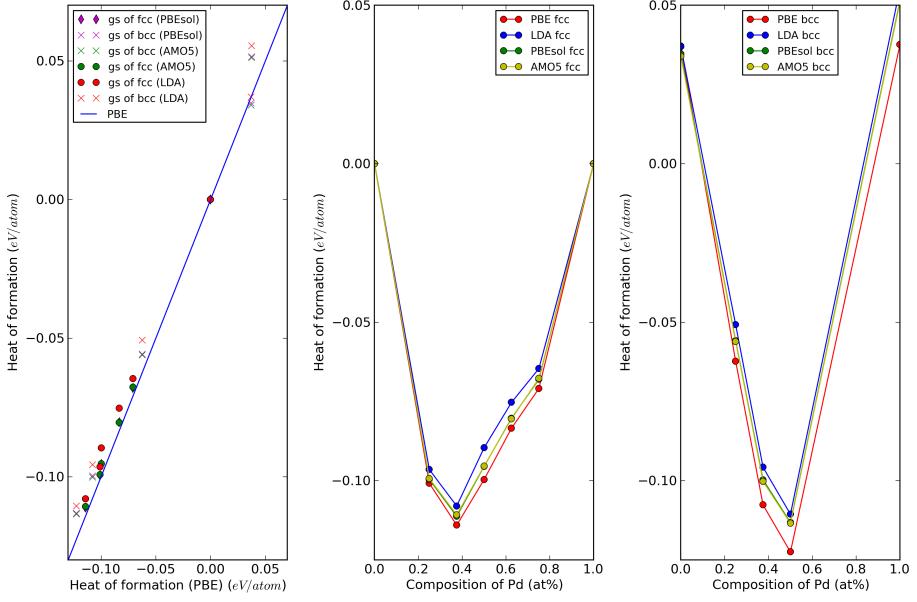


Figure 25: Left panel a shows the heat of formation of all ground state (gs) structures in different functionals. For each point, the x value is the heat of formation for each ground states using PBE functional. Each y value is the heat of formation for the same structure using PBEsol, AMO5 and LDA functionals. The blue line is the parity line comparing the results from various functionals. Middle panel shows the convex hull of fcc lattice utilizing various functionals. The right panel shows the convex hull of bcc lattice using various functionals.

Figure 25 shows the influence of functional choice on heat of formation of the same ground state structures. All the calculations converge to 1 meV/atom with high energy cutoff and density of  $k$ -points grid. The middle panel shows that the heat of formation increases as the functional changes from PBE to LDA. The right panel shows similar trend which indicates the energy differences between the most stable structure of CuPd bcc and fcc will change through various functionals. In this case, construction of cluster

expansion based on various functionals may have significant change to the order-disorder transition temperature and stability of CuPd bcc lattice.

## 7. Effect of error in DFT calculation on phase diagram

In order to evaluate whether energetic error of 1-2 meV/atom matters in phase boundary, we manually decrease energy of CuPd B2 ground state structure by 1-3 meV/atom. The resulting phase diagram is summarized in Figure 26.

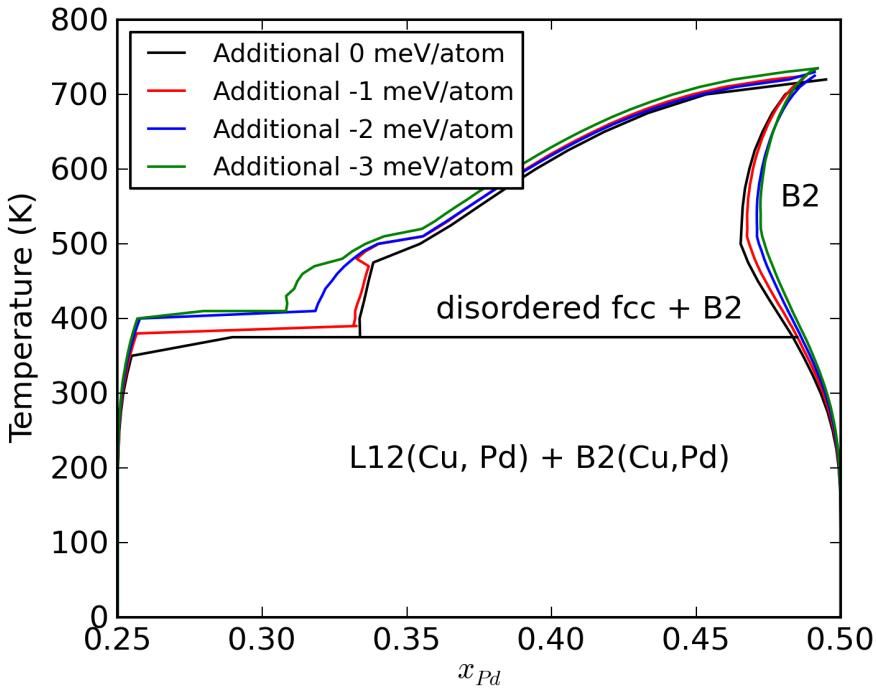


Figure 26: Effect of error in DFT calculation on transition temperature.

As is shown in figure 26, the order-disorder transition temperature increases by 5-15 K since the CuPd B2 energy decreases by 1-3 meV/atom.

Meanwhile, as energy of B2 phase is reduced, the B2 region remains its shape.

Therefore, it is evident that error in DFT energy does not significantly change the region of B2 phase.

## 8. References

- [1] A. van de Walle, G. Ceder, [Automating first-principles phase diagram calculations](#), J. Phase Equilib. 23 (4) (2002) 348–359. doi:[10.1361/105497102770331596](https://doi.org/10.1361/105497102770331596).  
URL <http://dx.doi.org/10.1361/105497102770331596>
- [2] A. van de Walle, M. Asta, G. Ceder, [The Alloy Theoretic Automated Toolkit: A user guide](#), Calphad 26 (4) (2002) 539 – 553. doi:[10.1016/S0364-5916\(02\)80006-2](https://doi.org/10.1016/S0364-5916(02)80006-2).  
URL <http://www.sciencedirect.com/science/article/pii/S0364591602800062>
- [3] A. van de Walle, M. Asta, [Self-driven lattice-model monte carlo simulations of alloy thermodynamic properties and phase diagrams](#), Modelling Simul. Mater. Sci. Eng. 10 (5) (2002) 521–538. doi:[10.1088/0965-0393/10/5/304](https://doi.org/10.1088/0965-0393/10/5/304).  
URL <http://dx.doi.org/10.1088/0965-0393/10/5/304>
- [4] D. Alfe, [Phon: A program to calculate phonons using the small displacement method](#), Comput. Phys. Commun. 180 (12) (2009) 2622–2633. doi:[10.1016/j.cpc.2009.03.010](https://doi.org/10.1016/j.cpc.2009.03.010).  
URL <http://dx.doi.org/10.1016/j.cpc.2009.03.010>

- [5] P. Subramanian, D. Laughlin, Cu-Pd (copper-palladium), J. Phase Equilib. 12 (2) (1991) 231–243. doi:10.1007/BF02645723.  
 URL <http://dx.doi.org/10.1007/BF02645723>
- [6] J. Teeriniemi, P. Taskinen, K. Laasonen, First-principles investigation of the Cu-Ni, Cu-Pd, and Ni-Pd binary alloy systems, Intermetallics 57 (2015) 41–50. doi:10.1016/j.intermet.2014.09.006.  
 URL <http://dx.doi.org/10.1016/j.intermet.2014.09.006>
- [7] A. Van De Walle, G. Ceder, The effect of lattice vibrations on substitutional alloy thermodynamics, Rev. Mod. Phys. 74 (2002) 11–45. doi:10.1103/RevModPhys.74.11.  
 URL <http://link.aps.org/doi/10.1103/RevModPhys.74.11>

## 9. Appendix

### 9.1. Example of $k$ -points convergence in various functionals

#### 9.1.1. Cu fcc A1 PBE

---

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7
8
9  kpt = [10,12,14,16,18,20]# set the k-points density from 10X10X10 to 20X20X20
10
11 energy = []
12 with jasp('..../CuPd_fcc/0')as calc:#copy structure of Cu fcc from cluster expansion of CuPd fcc
```

```

13     atoms0=calc.get_atoms()
14
15     for a in kpt:
16         with jasp('.../.../convergence/Cu-fcc-PBE-kpt{0}'.format(a)),
17             xc = 'PBE',
18             kpts = (a,a,a),
19             encut = 500,
20             ediff = 1e-5,
21             isif = 7,
22             ibrion = 2,
23             nsw = 40,
24             atoms = atoms0) as calc:
25             atoms0.get_forces()
26         try:
27             energy.append(atoms0.get_potential_energy()/len(atoms0))
28         except (VaspSubmitted, VaspQueued):
29             pass
30
31     import matplotlib.pyplot as plt
32     plt.figure()
33     plt.plot(kpt,energy,'ro-')
34     plt.xlabel('Number of k-points per reciprocal atom')
35     plt.ylabel('Energy (eV/atom)')
36     plt.title(' Convergence of k-points density for Cu fcc structure')
37     plt.savefig('./images-si/kpt-Cu-fcc-PBE.png')
38     plt.show()

```

---

### 9.1.2. Cu fcc A1 PBEsol

---

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7
8
9  kpt = [10,11,12,13,14,15,16,17,18,19,20,21,22,23] # set the number of k-points from 10X10X10 to 23X23X23

```

```

10
11 energy = []
12 with jasp('../.../CuPd_fcc/0')as calc:# Copy the structure of Cu fcc from cluster expansion of CuPd fcc
13 atoms0 = calc.get_atoms()
14 for a in kpt:
15     with jasp('../.../convergence/Cufcc-PBESOL-kpt{0}'.format(a),
16             xc = 'LDA',
17             gga = 'PS',
18             kpts = (a,a,a),
19             encut = 500,
20             ediff = 1e-6,
21             isif = 7,
22             ibrion = 2,
23             nsw = 40,
24             atoms = atoms0) as calc:
25         atoms0.get_forces()
26
27     try:
28         energy.append(atoms0.get_potential_energy()/len(atoms0))
29     except (VaspSubmitted, VaspQueued):
30         pass
31
32 import matplotlib.pyplot as plt
33 plt.figure()
34 plt.plot(kpt,energy,'ro-')
35 plt.xlabel('K-points')
36 plt.ylabel('Energy (eV/atom)')
37 plt.title('K-points convergence for Cufcc')
38 plt.savefig('./images-si/kpt-converge-Cufcc-PBEsol.png')
39 plt.show()

```

---

### 9.1.3. Cu fcc A1 LDA

---

```

1 from ase import Atom, Atoms
2 import numpy as np
3 from jasp import *
4 from ase.visualize import view
5 from ase.lattice import bulk

```

```

6   import os
7
8
9
10  kpt = [10,11,12,13,14,15,16,17]# set the k-points density from 10X10X10 to 17X17X17
11
12  energy = []
13  with jasp('.../.../CuPd_fcc/0')as calc:# copy the structure of Cu from cluster expansion of CuPd fcc
14      atoms0 = calc.get_atoms()
15  for a in kpt:
16      with jasp('.../.../convergence/Cufcc-LDA-kpt{0}'.format(a),
17                  xc = 'LDA',
18                  kpts = (a,a,a),
19                  encut = 500,
20                  ediff = 1e-6,
21                  isif = 7,
22                  ibrion = 2,
23                  nsw = 40,
24                  atoms = atoms0) as calc:
25          atoms0.get_forces()
26
27  try:
28      energy.append(atoms0.get_potential_energy()/len(atoms0))
29  except (VaspSubmitted, VaspQueued):
30      pass
31
32  import matplotlib.pyplot as plt
33  plt.figure()
34  plt.plot(kpt,energy,'ro-')
35  plt.xlabel('K-points')
36  plt.ylabel('Energy (eV/atom)')
37  plt.title('K-points convergence for Cufcc')
38  plt.savefig('./images-si/kpt-converge-Cufcc-LDA.png')
39
40  plt.show()

```

---

#### 9.1.4. Cu fcc A1 AMO5

---

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7
8
9
10 kpt = [10,11,12,13,14,15,16,17,18,19,20,21,22]# set the k-points density from 10X10X10 to 22X22X22
11
12 energy = []
13 with jasp('.../.../CuPd_fcc/0')as calc:# copy the structure of Cu fcc from cluster expansion of CuPd fcc
14     atoms0 = calc.get_atoms()
15     for a in kpt:
16         with jasp('.../.../convergence/Cufcc-AM05-kpt{0}'.format(a),
17                    xc = 'LDA',
18                    gga = 'AM',
19                    kpts = (a,a,a),
20                    encut = 500,
21                    ediff = 1e-6,
22                    isif = 7,
23                    ibrion = 2,
24                    nsw = 40,
25                    atoms = atoms0) as calc:
26             atoms0.get_forces()
27
28         try:
29             energy.append(atoms0.get_potential_energy()/len(atoms0))
30         except (VaspSubmitted, VaspQueued):
31             pass
32
33 import matplotlib.pyplot as plt
34 plt.figure()
35 plt.plot(kpt,energy,'ro-')
36 plt.xlabel('K-points')
37 plt.ylabel('Energy (eV/atom)')
38 plt.title('K-points convergence check for Cu fcc')

```

```
39 plt.savefig('~/images-si/kpt-converge-Cufcc-AM05.png')
40 plt.show()
```

---

## 9.2. Examples of encut convergence in various functionals

### 9.2.1. Cufcc A1 PBESOL

```
1 from ase import Atom, Atoms
2 import numpy as np
3 from jasp import *
4 from ase.visualize import view
5 from ase.lattice import bulk
6 import os
7
8
9
10 Encut = [400,450,500,550,600] # sets of different energy cutoff
11
12 energy = []
13
14 with jasp('CuPd_fcc/0')as calc: # use the configuration 0 in cluster expansion of CuPd fcc as structure input
15     atoms0 = calc.get_atoms()
16
17 for a in Encut:
18     with jasp('convergence/Cufcc-PBESOL-encut{0}'.format(a),
19               xc = 'LDA',
20               gga = 'PS',
21               kpts = (13,13,13),
22               encut = a,
23               ediff = 1e-6,
24               isif = 7,
25               ibrion = 2,
26               nsw = 40,
27               atoms = atoms0) as calc:
28         atoms0.get_forces()
29
30     try:
31         atoms = calc.get_atoms()
```

```

32         energy.append(atoms0.get_potential_energy()/len(atoms0))
33     except (VaspSubmitted, VaspQueued):
34         pass
35
36     import matplotlib.pyplot as plt
37     plt.figure()
38     plt.plot(Encut,energy,'ro-')
39     plt.xlabel('Encut (eV)')
40     plt.ylabel('Energy (eV/atom)')
41     plt.title('Encut convergence for Cu fcc')
42     plt.savefig('./convergence/Encut converge-Cufcc-AMOF.png')
43     plt.show()

```

---

[./convergence/Encut converge-Cufcc-AMOF.png]]

### 9.2.2. Cufcc A1 LDA

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7
8
9
10 Encut = [300,350,400,450,500,550]# set all the energy cutoff number
11
12 energy = []
13 with jasp('CuPd_fcc/0')as calc:# copy the structure of Cu in Cluster expansion of CuPd fcc
14     atoms0 = calc.get_atoms()
15
16
17 for a in Encut:
18     with jasp('convergence/Cufcc-LDA-encut{0}'.format(a),# the calculation result is stored in Cufcc-LDA-encutXX
19         xc = 'LDA',
20         kpts = (13,13,13),
21         encut = a,

```

```

22         ediff = 1e-5,
23         isif = 7,
24         ibrion = 2,
25         nsw = 40,
26         atoms = atoms0) as calc:
27     atoms0.get_forces()
28
29     try:
30         energy.append(atom.get_potential_energy()/len(atom))
31     except (VaspSubmitted, VaspQueued):
32         pass
33
34     import matplotlib.pyplot as plt
35     plt.figure()
36     plt.plot(Encut,energy,'ro-')
37     plt.xlabel('Encut (eV)')
38     plt.ylabel('Energy (eV/atom)')
39     plt.title('Encut convergence for Cufcc')
40     plt.savefig('./convergence/Encut converge-Cufcc-AMOF.png')
41     plt.show()

```

---

[./convergence/Encut converge-Cufcc-AMOF.png]]

### 9.2.3. Cufcc A1 AMO5

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7
8
9
10 Encut = [350,400,450,500,550,600]# set the energy cutoff
11
12 energy = []
13 with jasp('CuPd_fcc/0')as calc:# copy the structure of Cu in cluster expansion of CuPd fcc to atom0

```

```

14     atoms0 = calc.get_atoms()
15
16 for a in Encut:
17     with jasp('convergence/Cufcc-AM05-encut{0}'.format(a),# the result is stored in this directory
18         xc = 'LDA',
19         gga = 'AM',
20         kpts = (13,13,13),
21         encut = a,
22         ediff = 1e-6,
23         isif = 7,
24         ibrion = 2,
25         nsw = 40,
26         atoms = atoms0) as calc:
27     atoms0.get_forces()
28
29 try:
30
31     energy.append(atoms0.get_potential_energy()/len(atoms0))
32 except (VaspSubmitted, VaspQueued):
33     pass
34
35 import matplotlib.pyplot as plt
36 plt.figure()
37 plt.plot(Encut,energy,'ro-')
38 plt.xlabel('Encut')
39 plt.ylabel('Energy (eV/atom)')
40 plt.title('encut convergence check for Cufcc')
41 plt.savefig('./convergence/Encut converge-Cufcc-AMOF.png')
42 plt.show()

```

---

[./convergence/Encut converge-Cufcc-AMOF.png]]

#### 9.2.4. Cufcc A1 PBE

---

```

1 from ase import Atom, Atoms
2 import numpy as np
3 from jasp import *
4 from ase.visualize import view

```

```

5   from ase.lattice import bulk
6   import os
7
8
9
10  Encut = [350,400,450,500,550,600]# set the energy cutoff
11
12  energy = []
13  with jasp('CuPd_fcc/0')as calc:# copy the structure of Cu in cluster expansion of CuPd fcc to atom0
14      atoms0 = calc.get_atoms()
15
16  for a in Encut:
17      with jasp('convergence/Cufcc-PBE-encut{0}'.format(a),# the result is stored in this directory
18          xc = 'PBE',
19          kpts = (13,13,13),
20          encut = a,
21          ediff = 1e-6,
22          isif = 7,
23          ibrion = 2,
24          nsw = 40,
25          atoms = atoms0) as calc:
26      atoms0.get_forces()
27
28  try:
29
30      energy.append(atoms0.get_potential_energy()/len(atoms0))
31  except (VaspSubmitted, VaspQueued):
32      pass
33
34  import matplotlib.pyplot as plt
35  plt.figure()
36  plt.plot(Encut,energy,'ro-')
37  plt.xlabel('Encut')
38  plt.ylabel('Energy (eV/atom)')
39  plt.title('Encut convergence for Cufcc')
40  plt.savefig('./convergence/Encut converge-Cufcc-PBE.png')
41  plt.show()

```

---

```
[./convergence/Encut converge-Cufcc-AMOF.png]]
```

### 9.3. Figure 1 result of cluster expansion

```
1  from vasp import Vasp
2  from ase.io import read
3  import os
4  import matplotlib.pyplot as plt
5  fig = plt.figure(figsize=(11,8))
6  ax = fig.add_subplot(221)
7  f = open('../../../CuPd_fcc_encut500_4000/gs.out','r')
8  lines = f.readlines()
9  f.close()
10
11 gs_fcc_concentration,gs_fcc_energy,gs_fcc_fitenergy,gs_fcc_index = [],[],[],[]
12 for line in lines:
13     fields = line.split()
14     if len(fields) == 4:
15         c,e,fe,i = line.split()
16     elif len(fields) == 6: #nmaps
17         c, e,fe,err,nu,i = line.split()
18         gs_fcc_concentration.append(float(c))
19         gs_fcc_energy.append(float(e))
20         gs_fcc_fitenergy.append(float(fe))
21         gs_fcc_index.append(int(i))
22
23
24
25
26
27 f = open('../../../CuPd_fcc_encut500_4000/fit.out','r')
28 lines = f.readlines()
29 f.close()
30
31 gs_c_fcc_concentration,gs_c_fcc_energy,gs_c_fcc_fitenergy,gs_c_fcc_index = [],[],[],[]
32 for line in lines:
33     fields = line.split()
34     if len(fields) == 4:
```

```

35         c,e,fe,i = line.split()
36     elif len(fields) == 6:  #mmaps
37         c, e,fe,err,nu,i = line.split()
38         gs_c_fcc_concentration.append(float(c))
39         gs_c_fcc_energy.append(float(e))
40         gs_c_fcc_fitenergy.append(float(fe))
41         gs_c_fcc_index.append(int(i))
42
43
44 f = open('../../../CuPd_bcc_encut500_3000/gs.out','r')
45 lines = f.readlines()
46 f.close()
47
48 gs_bcc_concentration,gs_bcc_energy,gs_bcc_fitenergy,gs_bcc_index = [],[],[],[]
49 for line in lines:
50     fields = line.split()
51     if len(fields) == 4:
52         c,e,fe,i = line.split()
53     elif len(fields) == 6:  #mmaps
54         c, e,fe,err,nu,i = line.split()
55         gs_bcc_concentration.append(float(c))
56         gs_bcc_energy.append(float(e))
57         gs_bcc_fitenergy.append(float(fe))
58         gs_bcc_index.append(int(i))
59
60
61
62 f = open('../../../CuPd_bcc_encut500_3000/fit.out','r')
63 lines = f.readlines()
64 f.close()
65
66 gs_c_bcc_concentration,gs_c_bcc_energy,gs_c_bcc_fitenergy,gs_c_bcc_index = [],[],[],[]
67 for line in lines:
68     fields = line.split()
69     if len(fields) == 4:
70         c,e,fe,i = line.split()
71     elif len(fields) == 6:  #mmaps
72         c, e,fe,err,nu,i = line.split()

```

```

73     gs_c_bcc_concentration.append(float(c))
74     gs_c_bcc_energy.append(float(e))
75     gs_c_bcc_fitenergy.append(float(fe))
76     gs_c_bcc_index.append(int(i))

77

78

79 import matplotlib.pyplot as plt
80 ax.plot([0,1],[0,0],'k--')
81 ax.plot(gs_c_fcc_concentration,gs_c_fcc_fitenergy,'gx',label=" Fitted Energies")
82 ax.plot(gs_fcc_concentration,gs_fcc_energy,'ko-',
83         label='Known Ground State',markersize=4,
84         markerfacecolor='blue',markeredgecolor='black')
85 ax.plot(gs_fcc_concentration,gs_fcc_fitenergy,'g+')
86 ax.plot(gs_c_fcc_concentration,gs_c_fcc_energy,'r+',label="known Structure")
87 ax.set_xlabel('$x_{\%s}$ \%Pd')
88 ax.set_ylabel('Heat Of Formation ($eV/atom$)')
89 ax.text(0.05,0.03,'(a)',size=20)
90 ax.legend(loc=4,prop={'size':10})
91 ax.set_ylim([-0.15,0.05])
92 ax = fig.add_subplot(222)
93 ax.plot(gs_c_bcc_concentration,gs_c_bcc_fitenergy,'gx',label=" Fitted Energies")
94 ax.plot(gs_bcc_concentration,gs_bcc_energy,'ko-',
95         label='Known Ground State',markersize=4,
96         markerfacecolor='blue',markeredgecolor='black')
97 ax.plot(gs_bcc_concentration,gs_bcc_fitenergy,'g+')
98 ax.plot(gs_c_bcc_concentration,gs_c_bcc_energy,'r+',label="known Structure")
99 ax.set_xlabel('$x_{\%s}$ \%Pd')
100 ax.set_ylabel('Heat Of Formation ($eV/atom$)')
101 ax.text(0.05,0.03, '(b)', size=20)
102 ax.set_ylim([-0.15,0.05])
103
104 plt.tight_layout()
105 ax.legend(loc=4,prop={'size':9})
106
107
108
109 cluseci = []
110 clusradii = []

```

```

111  f = open('../../../CuPd_fcc_enuc500_4000/clusinfo.out','r')
112  for line in f:
113      if line.startswith('#'):
114          continue
115      try:
116          n, radius, df, eci = line.split()
117          clusradii.append((int(n)-2)*20 + float(radius))
118          cluseci.append(float(eci))
119      except ValueError:
120          continue
121
122  f.close()
123
124  ax = fig.add_subplot(223)
125  ax.plot(clusradii,cluseci,'ko')
126  #ax = gca()
127
128  ax.set_xticks([0,5,10,15,
129                  20,25,30,35,
130                  40, 45, 50,55])
131
132  ax.set_xticklabels(['pairs','5','10','15',
133                      'trip','5','10','15',
134                      'quad','5','10','15'])
135
136  ax.set_xlabel('Cluster radius ($\AA$)')
137  ax.set_ylabel('ECI ($eV$)')
138  ax.set_ylim([-0.005,0.035])
139  ax.text(2,0.030,(c),size=20)
140  #ax.title('ECI vs. cluster radius')
141  plt.tight_layout()
142
143
144  cluseci = []
145  clusradii = []
146  f = open('../../../CuPd_bcc_enuc500_3000/clusinfo.out','r')
147  for line in f:
148      if line.startswith('#'):

```

```

149         continue
150     try:
151         n, radius, df, eci = line.split()
152         clusradii.append((int(n)-2)*20 + float(radius))
153         cluseci.append(float(eci))
154     except ValueError:
155         continue
156
157 f.close()
158
159 ax = fig.add_subplot(224)
160 ax.plot(clusradii,cluseci,'ko')
161 #ax = gca()
162
163 ax.set_xticks([0,5,10,15,
164                 20,25,30,35,
165                 40, 45, 50,55])
166
167 ax.set_xticklabels(['pairs','5','10','15',
168                      'trip','5','10','15',
169                      'quad','5','10','15'])
170
171 ax.set_xlabel('Cluster radius ($\AA$)')
172 ax.set_ylabel('ECI ($eV$)')
173 ax.set_ylim([-0.005,0.035])
174 ax.text(2,0.030,(d),size=20)
175 #ax.title('ECI vs. cluster radius')
176 plt.tight_layout()
177
178 str = './images-si/fcc-bcc-convexhull'
179 for ext in ['.png', '.eps']:
180     plt.savefig(str + ext, dpi=300)
181 plt.show()

```

---

## 9.4. Convergence of encut for force constant

### 9.4.1. Cu-Cu bond

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7  import matplotlib.pyplot as plt
8
9
10 filename = '../../../../../CuPd_bcc_dr0.02_<math>\text{encut}</math>425/f_Cu-Cu.dat'
11 with open(filename) as f:
12     lines = f.readlines()
13 bondL0 = []
14 strech = []
15 bend = []
16 for i,line in enumerate(lines):
17     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
18     strech += [np.array(line[0:].split(), dtype=float)[1]]
19     bend += [np.array(line[0:].split(), dtype=float)[2]]
20 plt.subplot(1,2,1)
21 plt.plot(bondL0,strech,'d',label='dr = 0.02 \u00a5 (encut 425)')
22 plt.subplot(1,2,2)
23 plt.plot(bondL0,bend,'d',label='dr = 0.02 \u00a5 (encut 425)')
24
25
26
27
28 filename = '../../../../../CuPd_bcc_dr0.02_<math>\text{encut}</math>500/f_Cu-Cu.dat'
29 with open(filename) as f:
30     lines = f.readlines()
31 bondL0 = []
32 strech = []
33 bend = []
34 for i,line in enumerate(lines):
35     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
36     strech += [np.array(line[0:].split(), dtype=float)[1]]
37     bend += [np.array(line[0:].split(), dtype=float)[2]]
38 plt.subplot(1,2,1)

```

```

39 plt.plot(bondL0,strech,'yd',label='dr = 0.02 $\AA$ (encut 500)')
40 plt.subplot(1,2,2)
41 plt.plot(bondL0,bend,'yd',label='dr = 0.02 $\AA$ (encut 500)')
42
43
44
45
46 # take Cu-Cu
47 filename = '../../../../../CuPd_bcc_encut500_dr0.02kpt/f_Cu-Cu.dat'
48 with open(filename) as f:
49     lines = f.readlines()
50 bondL0 = []
51 strech = []
52 bend = []
53 for i,line in enumerate(lines):
54     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
55     strech += [np.array(line[0:].split(), dtype=float)[1]]
56     bend += [np.array(line[0:].split(), dtype=float)[2]]
57 plt.subplot(1,2,1)
58 plt.plot(bondL0,strech,'ko',label='dr = 0.02 $\AA$ (encut 350)')
59 plt.xlabel('Bond length ($\AA$)')
60 plt.ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
61 plt.legend(loc='best',prop={'size':8})
62 plt.subplot(1,2,2)
63 plt.plot(bondL0,bend,'ko',label='dr = 0.02 $\AA$ (encut 350)')
64 plt.xlabel('Bond length ($\AA$)')
65 plt.ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
66 plt.legend(loc='best',prop={'size':8})
67 plt.tight_layout()
68 for ext in ['.png', '.eps']:
69     plt.savefig('./images-si/Cu-Cu-testencut' + ext, dpi=300)
70 #plt.savefig('Cu-Cu-testencut.png')
71
72 plt.show()

```

---

#### 9.4.2. Cu-Pd bond

---

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7  import matplotlib.pyplot as plt
8
9
10 filename = '../../../../../CuPd_bcc_dr0.02_<math>\text{encut}</math>425/f_Cu-Pd.dat'
11 with open(filename) as f:
12     lines = f.readlines()
13 bondL0 = []
14 strech = []
15 bend = []
16 for i,line in enumerate(lines):
17     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
18     strech += [np.array(line[0:].split(), dtype=float)[1]]
19     bend += [np.array(line[0:].split(), dtype=float)[2]]
20 plt.subplot(1,2,1)
21 plt.plot(bondL0,strech,'d',label='dr = 0.02 \u00a5 (encut 425)')
22 plt.subplot(1,2,2)
23 plt.plot(bondL0,bend,'d',label='dr = 0.02 \u00a5 (encut 425)')
24
25
26
27
28 filename = '../../../../../CuPd_bcc_dr0.02_<math>\text{encut}</math>500/f_Cu-Pd.dat'
29 with open(filename) as f:
30     lines = f.readlines()
31 bondL0 = []
32 strech = []
33 bend = []
34 for i,line in enumerate(lines):
35     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
36     strech += [np.array(line[0:].split(), dtype=float)[1]]
37     bend += [np.array(line[0:].split(), dtype=float)[2]]
38 plt.subplot(1,2,1)

```

```

39 plt.plot(bondL0,strech,'d',label='dr = 0.02 $\AA$ (encut 500)')
40 plt.subplot(1,2,2)
41 plt.plot(bondL0,bend,'d',label='dr = 0.02 $\AA$ (encut 500)')
42
43
44
45
46 # take Cu-Cu
47 filename = '../../../../../CuPd_bcc_encut500_dr0.02kpt/f_Cu-Pd.dat'
48 with open(filename) as f:
49     lines = f.readlines()
50 bondL0 = []
51 strech = []
52 bend = []
53 for i,line in enumerate(lines):
54     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
55     strech += [np.array(line[0:].split(), dtype=float)[1]]
56     bend += [np.array(line[0:].split(), dtype=float)[2]]
57 plt.subplot(1,2,1)
58 plt.plot(bondL0,strech,'ko',label='dr = 0.02 $\AA$ (encut 350)')
59 plt.xlabel('Bond length ($\AA$)')
60 plt.ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
61 plt.legend(loc='best',prop={'size':8})
62 plt.subplot(1,2,2)
63 plt.plot(bondL0,bend,'ko',label='dr = 0.02 $\AA$ (encut 350)')
64 plt.xlabel('Bond length ($\AA$)')
65 plt.ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
66 plt.legend(loc='best',prop={'size':8})
67 plt.tight_layout()
68 for ext in ['.png', '.eps']:
69     plt.savefig('./images-si/Cu-Pd-testencut' + ext, dpi=300)
70 plt.show()

```

---

#### 9.4.3. Pd-Pd bond

---

```

1 from ase import Atom, Atoms
2 import numpy as np
3 from jasp import *

```

```

4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7  import matplotlib.pyplot as plt
8
9
10 filename = '../../../../../CuPd_bcc_dr0.02_encyt425/f_Pd-Pd.dat'
11 with open(filename) as f:
12     lines = f.readlines()
13 bondL0 = []
14 strech = []
15 bend = []
16 for i,line in enumerate(lines):
17     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
18     strech += [np.array(line[0:].split(), dtype=float)[1]]
19     bend += [np.array(line[0:].split(), dtype=float)[2]]
20 plt.subplot(1,2,1)
21 plt.plot(bondL0,strech,'d',label='dr = 0.02 $\AA$ (encyt 425)')
22 plt.subplot(1,2,2)
23 plt.plot(bondL0,bend,'d',label='dr = 0.02 $\AA$ (encyt 425)')
24
25
26
27 filename = '../../../../../CuPd_bcc_dr0.02_encyt500/f_Pd-Pd.dat'
28 with open(filename) as f:
29     lines = f.readlines()
30 bondL0 = []
31 strech = []
32 bend = []
33 for i,line in enumerate(lines):
34     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
35     strech += [np.array(line[0:].split(), dtype=float)[1]]
36     bend += [np.array(line[0:].split(), dtype=float)[2]]
37 plt.subplot(1,2,1)
38 plt.plot(bondL0,strech,'yd',label='dr = 0.02 $\AA$ (encyt 500)')
39 plt.subplot(1,2,2)
40 plt.plot(bondL0,bend,'yd',label='dr = 0.02 $\AA$ (encyt 500)')
41

```

```

42
43
44
45 # take Cu-Cu
46 filename = '../../../../../CuPd_bcc_enuc500_dr0.02kpt/f_Pd-Pd.dat'
47 with open(filename) as f:
48     lines = f.readlines()
49 bondL0 = []
50 strech = []
51 bend = []
52 for i,line in enumerate(lines):
53     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
54     strech += [np.array(line[0:].split(), dtype=float)[1]]
55     bend += [np.array(line[0:].split(), dtype=float)[2]]
56 plt.subplot(1,2,1)
57 plt.plot(bondL0,strech,'ko',label='dr = 0.02 $\AA$ (encut 350)')
58 plt.xlabel('Bond length ($\AA$)')
59 plt.ylabel('Stiffness of bond ($eV/\AA^2$)')
60 plt.legend(loc='best',prop={'size':8})
61 plt.subplot(1,2,2)
62 plt.plot(bondL0,bend,'ko',label='dr = 0.02 $\AA$ (encut 350)')
63 plt.xlabel('Bond length ($\AA$)')
64 plt.ylabel('Stiffness of bond ($eV/\AA^2$)')
65 plt.legend(loc='best',prop={'size':8})
66 plt.tight_layout()
67 for ext in ['.png', '.eps']:
68     plt.savefig('./images-si/Pd-Pd-testencut'+ext, dpi=300)
69 plt.show()

```

---

## 9.5. Convergence of er for force constant

### 9.5.1. Cu-Cu bond

---

```

1 from ase import Atom, Atoms
2 import numpy as np
3 from jasp import *
4 from ase.visualize import view
5 from ase.lattice import bulk

```

```

6   import os
7   import matplotlib.pyplot as plt
8
9   from matplotlib.ticker import MaxNLocator
10  fig=plt.figure(figsize=(10,5))
11
12  filename = '../../../../../CuPd_fcc_encut500_is1_dr0.035_dr7/f_Cu-Cu.dat'
13  with open(filename) as f:
14      lines = f.readlines()
15  bondL0 = []
16  strech = []
17  bend = []
18  for i,line in enumerate(lines):
19      bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
20      strech += [np.array(line[0:].split(), dtype=float)[1]]
21      bend += [np.array(line[0:].split(), dtype=float)[2]]
22  ax = fig.add_subplot(1,2,1)
23  ax.plot(bondL0,strech,'d',label='er=7 $\AA$')
24  ax = fig.add_subplot(1,2,2)
25  ax.plot(bondL0,bend,'d',label='er=7 $\AA$')
26
27
28
29
30
31
32 # take Cu-Cu
33 filename = '../../../../../CuPd_fcc_encut500_is1_dr0.035/f_Cu-Cu.dat'
34 with open(filename) as f:
35     lines = f.readlines()
36  bondL0 = []
37  strech = []
38  bend = []
39  for i,line in enumerate(lines):
40      bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
41      strech += [np.array(line[0:].split(), dtype=float)[1]]
42      bend += [np.array(line[0:].split(), dtype=float)[2]]
43  ax = fig.add_subplot(1,2,1)

```

```

44 ax.plot(bondL0,strech,'ko',label='er = 8.4 $\AA$')
45 plt.xlabel('Bond length ($\AA$)')
46 plt.ylabel('Stiffness of bond ($eV/\{\AA\}^2$)')
47 plt.legend(loc='best',prop={'size':9})
48 ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
49 ax=fig.add_subplot(1,2,2)
50 ax.plot(bondL0,bend,'ko',label='er = 8.4 $\AA$')
51 plt.xlabel('Bond length ($\AA$)')
52 plt.ylabel('Stiffness of bond ($eV/\{\AA\}^2$)')
53 plt.legend(loc='best',prop={'size':9})
54 plt.tight_layout()
55
56 ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
57 for ext in ['.png', '.eps']:
58     plt.savefig('./images-si/Cu-Cu-tester' + ext, dpi=300)
59 plt.show()

```

---

### 9.5.2. Cu-Pd bond

```

1 from ase import Atom, Atoms
2 import numpy as np
3 from jasp import *
4 from ase.visualize import view
5 from ase.lattice import bulk
6 import os
7 import matplotlib.pyplot as plt
8
9 from matplotlib.ticker import MaxNLocator
10 fig=plt.figure(figsize=(10,5))
11
12 filename = '../../../../../CuPd_fcc_encut500_is1_dr0.035_dr7/f_Cu-Pd.dat'
13 with open(filename) as f:
14     lines = f.readlines()
15 bondL0 = []
16 strech = []
17 bend = []
18 for i,line in enumerate(lines):
19     bondL0 += [np.array(line[0:]).split(), dtype=float)[0]]

```

```

20     strech += [np.array(line[0:].split(), dtype=float)[1]]
21     bend += [np.array(line[0:].split(), dtype=float)[2]]
22 ax = fig.add_subplot(1,2,1)
23 ax.plot(bondL0,strech,'d',label='er=7 $\AA$')
24 ax = fig.add_subplot(1,2,2)
25 ax.plot(bondL0,bend,'d',label='er=7 $\AA$')
26
27
28
29
30
31
32 # take Cu-Cu
33 filename = '../../../../../CuPd_fcc_encut500_is1_dr0.035/f_Cu-Pd.dat'
34 with open(filename) as f:
35     lines = f.readlines()
36 bondL0 = []
37 strech = []
38 bend = []
39 for i,line in enumerate(lines):
40     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
41     strech += [np.array(line[0:].split(), dtype=float)[1]]
42     bend += [np.array(line[0:].split(), dtype=float)[2]]
43 ax = fig.add_subplot(1,2,1)
44 ax.plot(bondL0,strech,'ko',label='er = 8.4 $\AA$')
45 plt.xlabel('Bond length ($\AA$)')
46 plt.ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
47 plt.legend(loc='best',prop={'size':9})
48 ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
49 ax=fig.add_subplot(1,2,2)
50 ax.plot(bondL0,bend,'ko',label='er = 8.4 $\AA$')
51 plt.xlabel('Bond length ($\AA$)')
52 plt.ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
53 plt.legend(loc='best',prop={'size':9})
54 plt.tight_layout()
55
56 ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
57 for ext in ['.png', '.eps']:

```

```

58     plt.savefig('./images-si/Cu-Pd-tester' + ext, dpi=300)
59 #plt.savefig('Cu-Cu-testencut.png')
60
61 plt.show()

```

---

### 9.5.3. Pd-Pd bond

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7  import matplotlib.pyplot as plt
8
9  from matplotlib.ticker import MaxNLocator
10 fig=plt.figure(figsize=(10,5))
11
12 filename = '../../../../../CuPd_fcc_encut500_is1_dr0.035_dr7/f_Pd-Pd.dat'
13 with open(filename) as f:
14     lines = f.readlines()
15 bondL0 = []
16 strech = []
17 bend = []
18 for i,line in enumerate(lines):
19     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
20     strech += [np.array(line[0:].split(), dtype=float)[1]]
21     bend += [np.array(line[0:].split(), dtype=float)[2]]
22 ax = fig.add_subplot(1,2,1)
23 ax.plot(bondL0,strech,'d',label='er=7 $\AA$')
24 ax = fig.add_subplot(1,2,2)
25 ax.plot(bondL0,bend,'d',label='er=7 $\AA$')
26
27
28 # take Cu-Cu
29 filename = '../../../../../CuPd_fcc_encut500_is1_dr0.035/f_Pd-Pd.dat'
30 with open(filename) as f:
31     lines = f.readlines()

```

```

32 bondL0 = []
33 strecth = []
34 bend = []
35 for i,line in enumerate(lines):
36     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
37     strecth += [np.array(line[0:].split(), dtype=float)[1]]
38     bend += [np.array(line[0:].split(), dtype=float)[2]]
39 ax = fig.add_subplot(1,2,1)
40 ax.plot(bondL0,strecth,'ko',label='er = 8.4 $\AA$')
41 plt.xlabel('Bond length ($\AA$)')
42 plt.ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
43 plt.legend(loc='best',prop={'size':9})
44 ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
45 ax=fig.add_subplot(1,2,2)
46 ax.plot(bondL0,bend,'ko',label='er = 8.4 $\AA$')
47 plt.xlabel('Bond length ($\AA$)')
48 plt.ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
49 plt.legend(loc='best',prop={'size':9})
50 plt.tight_layout()
51
52 ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
53 for ext in ['.png', '.eps']:
54     plt.savefig('../images-si/Pd-Pd-tester' + ext, dpi=300)
55 #plt.savefig('Cu-Cu-testencut.png')
56
57 plt.show()

```

---

## 9.6. Convergence of $k$ -points for force constant

### 9.6.1. Cu-Cu bond

---

```

1 from ase import Atom, Atoms
2 import numpy as np
3 from jasp import *
4 from ase.visualize import view
5 from ase.lattice import bulk
6 import os
7 import matplotlib.pyplot as plt

```

```

8
9
10 filename = ' ../../CuPd_bcc_dr0.02_kpt11/f_Cu-Cu.dat'
11 with open(filename) as f:
12     lines = f.readlines()
13 bondL0 = []
14 strech = []
15 bend = []
16 for i,line in enumerate(lines):
17     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
18     strech += [np.array(line[0:].split(), dtype=float)[1]]
19     bend += [np.array(line[0:].split(), dtype=float)[2]]
20 plt.subplot(1,2,1)
21 plt.plot(bondL0,strech,'d',label='dr = 0.02 $\AA$ (kpoints 10x10x10)')
22 plt.subplot(1,2,2)
23 plt.plot(bondL0,bend,'d',label='dr = 0.02 $\AA$ (kpoints 10x10x10)')
24
25
26
27
28 filename = ' ../../CuPd_bcc_dr0.02_kpt10/f_Cu-Cu.dat'
29 with open(filename) as f:
30     lines = f.readlines()
31 bondL0 = []
32 strech = []
33 bend = []
34 for i,line in enumerate(lines):
35     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
36     strech += [np.array(line[0:].split(), dtype=float)[1]]
37     bend += [np.array(line[0:].split(), dtype=float)[2]]
38 plt.subplot(1,2,1)
39 plt.plot(bondL0,strech,'d',label='dr = 0.02 $\AA$ (kpoints 9x9x9)')
40 plt.subplot(1,2,2)
41 plt.plot(bondL0,bend,'d',label='dr = 0.02 $\AA$ (kpoints 9x9x9)')
42
43
44
45 filename = ' ../../CuPd_bcc_encut500_dr0.02kpt/f_Cu-Cu.dat'

```

```

46     with open(filename) as f:
47         lines = f.readlines()
48     bondL0 = []
49     strech = []
50     bend = []
51     for i,line in enumerate(lines):
52         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
53         strech += [np.array(line[0:].split(), dtype=float)[1]]
54         bend += [np.array(line[0:].split(), dtype=float)[2]]
55     plt.subplot(1,2,1)
56     plt.plot(bondL0,strech,'o',label='dr = 0.02 $\AA$ (kpoints 7x7x7)')
57     plt.subplot(1,2,2)
58     plt.plot(bondL0,bend,'o',label='dr = 0.02 $\AA$ (kpoints 7x7x7)')
59
60
61     filename = '../../../../../CuPd_bcc_dr0.02_kpt8/f_Cu-Cu.dat'
62     with open(filename) as f:
63         lines = f.readlines()
64     bondL0 = []
65     strech = []
66     bend = []
67     for i,line in enumerate(lines):
68         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
69         strech += [np.array(line[0:].split(), dtype=float)[1]]
70         bend += [np.array(line[0:].split(), dtype=float)[2]]
71     plt.subplot(1,2,1)
72     plt.plot(bondL0,strech,'o',label='dr = 0.02 $\AA$ (kpoints 8x8x8)')
73     plt.subplot(1,2,2)
74     plt.plot(bondL0,bend,'o',label='dr = 0.02 $\AA$ (kpoints 8x8x8)')
75
76
77     # take Cu-Cu
78     filename = '../../../../../CuPd_bcc_encut500_dr0.02/f_Cu-Cu.dat'
79     with open(filename) as f:
80         lines = f.readlines()
81     bondL0 = []
82     strech = []
83     bend = []

```

```

84     for i,line in enumerate(lines):
85         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
86         strech += [np.array(line[0:].split(), dtype=float)[1]]
87         bend += [np.array(line[0:].split(), dtype=float)[2]]
88     plt.subplot(1,2,1)
89     plt.plot(bondL0,strech,'ko',label='dr = 0.02 $\\AA$ (kpoints 5x5x5)')
90     plt.xlabel('Bond length ($\\AA$)')
91     plt.ylabel('Stiffness of bond $eV/\\AA^2$')
92     plt.legend(loc='best',prop={'size':8})
93     plt.subplot(1,2,2)
94     plt.plot(bondL0,bend,'ko',label='dr = 0.02 $\\AA$ (kpoints 5x5x5)')
95     plt.xlabel('Bond length ($\\AA$)')
96     plt.ylabel('Stiffness of bond ($eV/\\AA^2$)')
97     plt.legend(loc='best',prop={'size':8})
98     plt.tight_layout()
99     for ext in ['.png','.eps']:
100        plt.savefig('../images-si/Cu-Cu-testkpts'+ext,dpi=300)
101    plt.show()

```

---

### 9.6.2. Cu-Pd bond

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7  import matplotlib.pyplot as plt
8
9
10 filename = '../../../../../CuPd_bcc_dr0.02_kpt11/f_Cu-Pd.dat'
11 with open(filename) as f:
12     lines = f.readlines()
13 bondL0 = []
14 strech = []
15 bend = []
16 for i,line in enumerate(lines):
17     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]

```

```

18     strech += [np.array(line[0:].split(), dtype=float)[1]]
19     bend += [np.array(line[0:].split(), dtype=float)[2]]
20     plt.subplot(1,2,1)
21     plt.plot(bondL0,strech,'d',label='dr = 0.02 $\AA$ (kpoints 10x10x10)')
22     plt.subplot(1,2,2)
23     plt.plot(bondL0,bend,'d',label='dr = 0.02 $\AA$ (kpoints 10x10x10)')
24
25
26
27
28     filename = '../.../CuPd_bcc_dr0.02_kpt10/f_Cu-Pd.dat'
29     with open(filename) as f:
30         lines = f.readlines()
31     bondL0 = []
32     strech = []
33     bend = []
34     for i,line in enumerate(lines):
35         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
36         strech += [np.array(line[0:].split(), dtype=float)[1]]
37         bend += [np.array(line[0:].split(), dtype=float)[2]]
38     plt.subplot(1,2,1)
39     plt.plot(bondL0,strech,'d',label='dr = 0.02 $\AA$ (kpoints 9x9x9)')
40     plt.subplot(1,2,2)
41     plt.plot(bondL0,bend,'d',label='dr = 0.02 $\AA$ (kpoints 9x9x9)')
42
43
44
45     filename = '../.../CuPd_bcc_enucut500_dr0.02kpt/f_Cu-Pd.dat'
46     with open(filename) as f:
47         lines = f.readlines()
48     bondL0 = []
49     strech = []
50     bend = []
51     for i,line in enumerate(lines):
52         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
53         strech += [np.array(line[0:].split(), dtype=float)[1]]
54         bend += [np.array(line[0:].split(), dtype=float)[2]]
55     plt.subplot(1,2,1)

```

```

56     plt.plot(bondL0,strech,'o',label='dr = 0.02 $\AA$ (kpoints 7x7x7)')
57     plt.subplot(1,2,2)
58     plt.plot(bondL0,bend,'o',label='dr = 0.02 $\AA$ (kpoints 7x7x7)')
59
60
61     filename = '../../../../../CuPd_bcc_dr0.02_kpt8/f_Cu-Pd.dat'
62     with open(filename) as f:
63         lines = f.readlines()
64     bondL0 = []
65     strech = []
66     bend = []
67     for i,line in enumerate(lines):
68         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
69         strech += [np.array(line[0:].split(), dtype=float)[1]]
70         bend += [np.array(line[0:].split(), dtype=float)[2]]
71     plt.subplot(1,2,1)
72     plt.plot(bondL0,strech,'o',label='dr = 0.02 $\AA$ (kpoints 8x8x8)')
73     plt.subplot(1,2,2)
74     plt.plot(bondL0,bend,'o',label='dr = 0.02 $\AA$ (kpoints 8x8x8)')
75
76
77     # take Cu-Cu
78     filename = '../../../../../CuPd_bcc_enucut500_dr0.02/f_Cu-Pd.dat'
79     with open(filename) as f:
80         lines = f.readlines()
81     bondL0 = []
82     strech = []
83     bend = []
84     for i,line in enumerate(lines):
85         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
86         strech += [np.array(line[0:].split(), dtype=float)[1]]
87         bend += [np.array(line[0:].split(), dtype=float)[2]]
88     plt.subplot(1,2,1)
89     plt.plot(bondL0,strech,'ko',label='dr = 0.02 $\AA$ (kpoints 5x5x5)')
90     plt.xlabel('Bond length ($\AA$)')
91     plt.ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
92     plt.legend(loc='best',prop={'size':8})
93     plt.subplot(1,2,2)

```

```

94 plt.plot(bondL0,bend,'ko',label='dr = 0.02 $\AA$ (kpoints 5x5x5)')
95 plt.xlabel('Bond length ($\AA$)')
96 plt.ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
97 plt.legend(loc='best',prop={'size':8})
98 plt.tight_layout()
99 for ext in ['.png','.eps']:
100 plt.savefig('../images-si/Cu-Pd-testkpts'+ext,dpi=300)
101 plt.show()

```

---

### 9.6.3. Pd-Pd bond

```

1 from ase import Atom, Atoms
2 import numpy as np
3 from jasp import *
4 from ase.visualize import view
5 from ase.lattice import bulk
6 import os
7 import matplotlib.pyplot as plt
8
9
10 filename = '../CuPd_bcc_dr0.02_kpt11/f_Pd-Pd.dat'
11 with open(filename) as f:
12     lines = f.readlines()
13 bondL0 = []
14 strech = []
15 bend = []
16 for i,line in enumerate(lines):
17     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
18     strech += [np.array(line[0:].split(), dtype=float)[1]]
19     bend += [np.array(line[0:].split(), dtype=float)[2]]
20 plt.subplot(1,2,1)
21 plt.plot(bondL0,strech,'d',label='dr = 0.02 $\AA$ (kpoints 10x10x10)')
22 plt.subplot(1,2,2)
23 plt.plot(bondL0,bend,'d',label='dr = 0.02 $\AA$ (kpoints 10x10x10)')
24
25
26
27

```

```

28     filename = '../../../../../CuPd_bcc_dr0.02_kpt10/f_Pd-Pd.dat'
29     with open(filename) as f:
30         lines = f.readlines()
31     bondL0 = []
32     strech = []
33     bend = []
34     for i,line in enumerate(lines):
35         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
36         strech += [np.array(line[0:].split(), dtype=float)[1]]
37         bend += [np.array(line[0:].split(), dtype=float)[2]]
38     plt.subplot(1,2,1)
39     plt.plot(bondL0,strech,'d',label='dr = 0.02 $\AA$ (kpoints 9x9x9)')
40     plt.subplot(1,2,2)
41     plt.plot(bondL0,bend,'d',label='dr = 0.02 $\AA$ (kpoints 9x9x9)')
42
43
44
45     filename = '../../../../../CuPd_bcc_enucut500_dr0.02kpt/f_Pd-Pd.dat'
46     with open(filename) as f:
47         lines = f.readlines()
48     bondL0 = []
49     strech = []
50     bend = []
51     for i,line in enumerate(lines):
52         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
53         strech += [np.array(line[0:].split(), dtype=float)[1]]
54         bend += [np.array(line[0:].split(), dtype=float)[2]]
55     plt.subplot(1,2,1)
56     plt.plot(bondL0,strech,'o',label='dr = 0.02 $\AA$ (kpoints 7x7x7)')
57     plt.subplot(1,2,2)
58     plt.plot(bondL0,bend,'o',label='dr = 0.02 $\AA$ (kpoints 7x7x7)')
59
60
61     filename = '../../../../../CuPd_bcc_dr0.02_kpt8/f_Pd-Pd.dat'
62     with open(filename) as f:
63         lines = f.readlines()
64     bondL0 = []
65     strech = []

```

```

66     bend = []
67     for i,line in enumerate(lines):
68         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
69         strech += [np.array(line[0:].split(), dtype=float)[1]]
70         bend += [np.array(line[0:].split(), dtype=float)[2]]
71     plt.subplot(1,2,1)
72     plt.plot(bondL0,strech,'o',label='dr = 0.02 $\AA$ (kpoints 8x8x8)')
73     plt.subplot(1,2,2)
74     plt.plot(bondL0,bend,'o',label='dr = 0.02 $\AA$ (kpoints 8x8x8)')
75
76
77 # take Cu-Cu
78 filename = '../../../../../CuPd_bcc_encut500_dr0.02/f_Pd-Pd.dat'
79 with open(filename) as f:
80     lines = f.readlines()
81 bondL0 = []
82 strech = []
83 bend = []
84 for i,line in enumerate(lines):
85     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
86     strech += [np.array(line[0:].split(), dtype=float)[1]]
87     bend += [np.array(line[0:].split(), dtype=float)[2]]
88 plt.subplot(1,2,1)
89 plt.plot(bondL0,strech,'ko',label='dr = 0.02 $\AA$ (kpoints 5x5x5)')
90 plt.xlabel('Bond length ($\AA$)')
91 plt.ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
92 plt.legend(loc='best',prop={'size':8})
93 plt.subplot(1,2,2)
94 plt.plot(bondL0,bend,'ko',label='dr = 0.02 $\AA$ (kpoints 5x5x5)')
95 plt.xlabel('Bond length ($\AA$)')
96 plt.ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
97 plt.legend(loc='best',prop={'size':8})
98 plt.tight_layout()
99 for ext in ['.png','.eps']:
100    plt.savefig('../images-si/Pd-Pd-testkpts'+ext,dpi=300)
101 plt.show()

```

---

## 9.7. Convergence of displacement of atom for force constant

### 9.7.1. bcc

#### 1. Cu-Cu bond

---

```
1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7  import matplotlib.pyplot as plt
8
9
10 from matplotlib.ticker import MaxNLocator
11 fig = plt.figure()
12
13
14 # displacement of atom = 0.005 Ang
15 filename = '../.../CuPd_bcc_500_is1_dr0.005/f_Cu-Cu.dat'
16 with open(filename) as f:
17     lines = f.readlines()
18 bondL0 = []
19 strech = []
20 bend = []
21 for i,line in enumerate(lines):
22     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
23     strech += [np.array(line[0:].split(), dtype=float)[1]]
24     bend += [np.array(line[0:].split(), dtype=float)[2]]
25 ax = fig.add_subplot(121)
26 ax.plot(bondL0,strech,'x',label='dr = 0.005 $\AA$')
27 ax = fig.add_subplot(122)
28 ax.plot(bondL0,bend,'x',label='dr = 0.005 $\AA$')
29
30 # displacement of atom = 0.04 Ang
31 filename = '../.../CuPd_bcc_500_dr0.04/f_Cu-Cu.dat'
32 with open(filename) as f:
33     lines = f.readlines()
```

```

34 bondL0 = []
35 strech = []
36 bend = []
37 for i,line in enumerate(lines):
38     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
39     strech += [np.array(line[0:].split(), dtype=float)[1]]
40     bend += [np.array(line[0:].split(), dtype=float)[2]]
41 ax = fig.add_subplot(121)
42 ax.plot(bondL0,strech,'d',label='dr = 0.04 $\AA$')
43 ax = fig.add_subplot(122)
44 ax.plot(bondL0,bend,'d',label='dr = 0.04 $\AA$')
45
46
47 # displacement of atom = 0.02 Ang
48 filename = '../.../CuPd_bcc_500_is1_dr0.02/f_Cu-Cu.dat'
49 with open(filename) as f:
50     lines = f.readlines()
51 bondL0 = []
52 strech = []
53 bend = []
54 for i,line in enumerate(lines):
55     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
56     strech += [np.array(line[0:].split(), dtype=float)[1]]
57     bend += [np.array(line[0:].split(), dtype=float)[2]]
58 ax = fig.add_subplot(121)
59 ax.plot(bondL0,strech,'d',label='dr = 0.02 $\AA$')
60 ax=fig.add_subplot(122)
61 ax.plot(bondL0,bend,'d',label='dr = 0.02 $\AA$')
62
63 # displacement of atom = 0.03 Ang
64 filename = '../.../CuPd_bcc_500_dr0.03/f_Cu-Cu.dat'
65 with open(filename) as f:
66     lines = f.readlines()
67 bondL0 = []
68 strech = []
69 bend = []
70 for i,line in enumerate(lines):
71     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]

```

```

72     strech += [np.array(line[0:].split(), dtype=float)[1]]
73     bend += [np.array(line[0:].split(), dtype=float)[2]]
74     ax= fig.add_subplot(121)
75     ax.plot(bondL0,strech,'o',label='dr = 0.03 $\AA$')
76     ax=fig.add_subplot(122)
77     ax.plot(bondL0,bend,'o',label='dr = 0.03 $\AA$')
78
79
80 # take Cu-Cu
81 filename = '../../../../../CuPd_bcc_encut500_is1_dr0.01/f_Cu-Cu.dat'
82 with open(filename) as f:
83     lines = f.readlines()
84 bondL0 = []
85 strech = []
86 bend = []
87 for i,line in enumerate(lines):
88     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
89     strech += [np.array(line[0:].split(), dtype=float)[1]]
90     bend += [np.array(line[0:].split(), dtype=float)[2]]
91 ax=fig.add_subplot(121)
92 ax.plot(bondL0,strech,'yo',label='dr = 0.01 $\AA$ ')
93 ax.set_xlabel('Bond length ($\AA$)')
94 ax.set_ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
95 ax.legend(loc='best',prop={'size':10})
96 ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
97 ax=fig.add_subplot(122)
98 ax.plot(bondL0,bend,'yo',label='dr = 0.01 $\AA$')
99 ax.set_xlabel('Bond length ($\AA$)')
100 ax.set_ylabel('Stiffness of bond $eV/\{\AA^2\}$')
101 ax.legend(loc='best',prop={'size':10})
102 plt.tight_layout()
103
104 ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
105 for exp in ['.png','.eps']:
106     plt.savefig('./images-si/Cu-Cu-testradius'+exp,dpi=300)
107
108 plt.show()

```

---

## 2. Cu-Pd bond

---

```
1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7  import matplotlib.pyplot as plt
8  from matplotlib.ticker import MaxNLocator
9  fig = plt.figure()
10
11
12 # displacement of atom = 0.005 Ang
13 filename = '../.../CuPd_bcc_enucut500_is1_dr0.005/f_Cu-Pd.dat'
14 with open(filename) as f:
15     lines = f.readlines()
16 bondL0 = []
17 strech = []
18 bend = []
19 for i,line in enumerate(lines):
20     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
21     strech += [np.array(line[0:].split(), dtype=float)[1]]
22     bend += [np.array(line[0:].split(), dtype=float)[2]]
23 ax = fig.add_subplot(121)
24 ax.plot(bondL0,strech,'x',label='dr = 0.005 $\AA$')
25 ax = fig.add_subplot(122)
26 ax.plot(bondL0,bend,'x',label='dr = 0.005 $\AA$')
27
28 # displacement of atom is 0.04 Ang
29 filename = '../.../CuPd_bcc_enucut500_dr0.04/f_Cu-Pd.dat'
30 with open(filename) as f:
31     lines = f.readlines()
32 bondL0 = []
33 strech = []
34 bend = []
35 for i,line in enumerate(lines):
36     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
```

```

37     strech += [np.array(line[0:].split(), dtype=float)[1]]
38     bend += [np.array(line[0:].split(), dtype=float)[2]]
39     ax = fig.add_subplot(121)
40     ax.plot(bondL0,strech,'d',label='dr = 0.04 $\AA$')
41     ax = fig.add_subplot(122)
42     ax.plot(bondL0,bend,'d',label='dr = 0.04 $\AA$')
43
44
45 # displacement of atom = 0.02 Ang
46 filename = '../.../CuPd_bcc_enucut500_is1_dr0.02/f_Cu-Pd.dat'
47 with open(filename) as f:
48     lines = f.readlines()
49 bondL0 = []
50 strech = []
51 bend = []
52 for i,line in enumerate(lines):
53     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
54     strech += [np.array(line[0:].split(), dtype=float)[1]]
55     bend += [np.array(line[0:].split(), dtype=float)[2]]
56 ax = fig.add_subplot(121)
57 ax.plot(bondL0,strech,'d',label='dr = 0.02 $\AA$')
58 ax=fig.add_subplot(122)
59 ax.plot(bondL0,bend,'d',label='dr = 0.02 $\AA$')
60
61 # displacement of atom = 0.03 Ang
62 filename = '../.../CuPd_bcc_enucut500_dr0.03/f_Cu-Pd.dat'
63 with open(filename) as f:
64     lines = f.readlines()
65 bondL0 = []
66 strech = []
67 bend = []
68 for i,line in enumerate(lines):
69     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
70     strech += [np.array(line[0:].split(), dtype=float)[1]]
71     bend += [np.array(line[0:].split(), dtype=float)[2]]
72 ax= fig.add_subplot(121)
73 ax.plot(bondL0,strech,'o',label='dr = 0.03 $\AA$')
74 ax=fig.add_subplot(122)

```

```

75     ax.plot(bondL0,bend,'o',label='dr = 0.03 $\AA$')
76
77
78     # displacement of atom = 0.01 Ang
79     filename = '../CuPd_bcc_encut500_is1_dr0.01/f_Cu-Pd.dat'
80     with open(filename) as f:
81         lines = f.readlines()
82     bondL0 = []
83     strech = []
84     bend = []
85     for i,line in enumerate(lines):
86         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
87         strech += [np.array(line[0:].split(), dtype=float)[1]]
88         bend += [np.array(line[0:].split(), dtype=float)[2]]
89     ax=fig.add_subplot(121)
90     ax.plot(bondL0,strech,'yo',label='dr = 0.01 $\AA$ ')
91     ax.set_xlabel('Bond length ($\AA$)')
92     ax.set_ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
93     ax.legend(loc='best',prop={'size':10})
94     ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
95     ax=fig.add_subplot(122)
96     ax.plot(bondL0,bend,'yo',label='dr = 0.01 $\AA$ ')
97     ax.set_xlabel('Bond length ($\AA$)')
98     ax.set_ylabel('Stiffness of bond $eV/\{\AA^2\}$')
99     ax.legend(loc='best',prop={'size':10})
100    plt.tight_layout()
101
102    ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
103    for ext in ['.png','.eps']:
104        plt.savefig('./images-si/Cu-Pd-testradius'+ext,dpi=300)
105
106    plt.show()

```

---

### 3. Pd-Pd bond

---

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *

```

```

4   from ase.visualize import view
5   from ase.lattice import bulk
6   import os
7   import matplotlib.pyplot as plt
8   from matplotlib.ticker import MaxNLocator
9   fig = plt.figure()
10
11
12  # displacement of atom = 0.005 Ang
13  filename = '../.../CuPd_bcc_enuc500_is1_dr0.005/f_Pd-Pd.dat'
14  with open(filename) as f:
15      lines = f.readlines()
16  bondL0 = []
17  strech = []
18  bend = []
19  for i,line in enumerate(lines):
20      bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
21      strech += [np.array(line[0:].split(), dtype=float)[1]]
22      bend += [np.array(line[0:].split(), dtype=float)[2]]
23  ax = fig.add_subplot(121)
24  ax.plot(bondL0,strech,'x',label='dr = 0.005 $\AA$')
25  ax = fig.add_subplot(122)
26  ax.plot(bondL0,bend,'x',label='dr = 0.005 $\AA$')
27
28  # displacement of atom is 0.04 Ang
29  filename = '../.../CuPd_bcc_enuc500_dr0.04/f_Pd-Pd.dat'
30  with open(filename) as f:
31      lines = f.readlines()
32  bondL0 = []
33  strech = []
34  bend = []
35  for i,line in enumerate(lines):
36      bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
37      strech += [np.array(line[0:].split(), dtype=float)[1]]
38      bend += [np.array(line[0:].split(), dtype=float)[2]]
39  ax = fig.add_subplot(121)
40  ax.plot(bondL0,strech,'d',label='dr = 0.04 $\AA$')
41  ax = fig.add_subplot(122)

```

```

42     ax.plot(bondL0,bend,'d',label='dr = 0.04 $\AA$')
43
44
45 # displacement of atom = 0.02 Ang
46 filename = '../.../CuPd_bcc_enucut500_is1_dr0.02/f_Pd-Pd.dat'
47 with open(filename) as f:
48     lines = f.readlines()
49 bondL0 = []
50 strecth = []
51 bend = []
52 for i,line in enumerate(lines):
53     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
54     strecth += [np.array(line[0:].split(), dtype=float)[1]]
55     bend += [np.array(line[0:].split(), dtype=float)[2]]
56 ax = fig.add_subplot(121)
57 ax.plot(bondL0,strecth,'d',label='dr = 0.02 $\AA$')
58 ax=fig.add_subplot(122)
59 ax.plot(bondL0,bend,'d',label='dr = 0.02 $\AA$')
60
61 # displacement of atom = 0.03 Ang
62 filename = '../.../CuPd_bcc_enucut500_dr0.03/f_Pd-Pd.dat'
63 with open(filename) as f:
64     lines = f.readlines()
65 bondL0 = []
66 strecth = []
67 bend = []
68 for i,line in enumerate(lines):
69     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
70     strecth += [np.array(line[0:].split(), dtype=float)[1]]
71     bend += [np.array(line[0:].split(), dtype=float)[2]]
72 ax= fig.add_subplot(121)
73 ax.plot(bondL0,strecth,'o',label='dr = 0.03 $\AA$')
74 ax=fig.add_subplot(122)
75 ax.plot(bondL0,bend,'o',label='dr = 0.03 $\AA$')
76
77
78 # displacement of atom = 0.01 Ang
79 filename = '../.../CuPd_bcc_enucut500_is1_dr0.01/f_Pd-Pd.dat',

```

```

80  with open(filename) as f:
81      lines = f.readlines()
82  bondL0 = []
83  strech = []
84  bend = []
85  for i,line in enumerate(lines):
86      bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
87      strech += [np.array(line[0:].split(), dtype=float)[1]]
88      bend += [np.array(line[0:].split(), dtype=float)[2]]
89  ax=fig.add_subplot(121)
90  ax.plot(bondL0,strech,'yo',label='dr = 0.01 $\AA$')
91  ax.set_xlabel('Bond length ($\AA$)')
92  ax.set_ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
93  ax.legend(loc='best',prop={'size':10})
94  ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
95  ax=fig.add_subplot(122)
96  ax.plot(bondL0,bend,'yo',label='dr = 0.01 $\AA$')
97  ax.set_xlabel('Bond length ($\AA$)')
98  ax.set_ylabel('Stiffness of bond $eV/\{\AA^2\}$')
99  ax.legend(loc='best',prop={'size':10})
100 plt.tight_layout()
101
102
103 ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
104 for ext in ['.png','.eps']:
105     plt.savefig('./images-si/Pd-Pd-testradius'+ext,dpi=300)
106
107 plt.show()

```

---

### 9.7.2. fcc

#### 1. Cu-Cu bond

---

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk

```

```

6   import os
7   import matplotlib.pyplot as plt
8   from matplotlib.ticker import MaxNLocator
9   fig = plt.figure()
10
11
12  # displacement of atom = 0.05 Ang
13  filename = '../.../CuPd_fcc_enucut500_is1_dr0.05/f_Cu-Cu.dat'
14  with open(filename) as f:
15      lines = f.readlines()
16  bondL0 = []
17  strecth = []
18  bend = []
19  for i,line in enumerate(lines):
20      bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
21      strecth += [np.array(line[0:].split(), dtype=float)[1]]
22      bend += [np.array(line[0:].split(), dtype=float)[2]]
23  ax = fig.add_subplot(121)
24  ax.plot(bondL0,strecth,'d',label='dr = 0.05 $\AA$')
25  ax=fig.add_subplot(122)
26  ax.plot(bondL0,bend,'d',label='dr = 0.05 $\AA$')
27
28  # displacement of atom is 0.2 Ang
29  filename = '../.../CuPd_fcc_enucut500_vib2_hp/f_Cu-Cu.dat'
30  with open(filename) as f:
31      lines = f.readlines()
32  bondL0 = []
33  strecth = []
34  bend = []
35  for i,line in enumerate(lines):
36      bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
37      strecth += [np.array(line[0:].split(), dtype=float)[1]]
38      bend += [np.array(line[0:].split(), dtype=float)[2]]
39  ax=fig.add_subplot(121)
40  ax.plot(bondL0,strecth,'d',label='dr = 0.2 $\AA$')
41  ax=fig.add_subplot(122)
42  ax.plot(bondL0,bend,'d',label='dr = 0.2 $\AA$')
43

```

```

44  # displacement of atom = 0.035 Ang
45  filename = '../.../CuPd_fcc_enuct500_is1_dr0.035/f_Cu-Cu.dat'
46  with open(filename) as f:
47      lines = f.readlines()
48  bondL0 = []
49  strech = []
50  bend = []
51  for i,line in enumerate(lines):
52      bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
53      strech += [np.array(line[0:].split(), dtype=float)[1]]
54      bend += [np.array(line[0:].split(), dtype=float)[2]]
55  ax=fig.add_subplot(121)
56  ax.plot(bondL0,strech,'o',label='dr = 0.035 $\AA$')
57  ax=fig.add_subplot(122)
58  ax.plot(bondL0,bend,'o',label='dr = 0.035 $\AA$')
59
60
61  # displacement of atom is 0.02 Ang
62  filename = '../.../CuPd_fcc_enuct500_is1_dr0.02/f_Cu-Cu.dat'
63  with open(filename) as f:
64      lines = f.readlines()
65  bondL0 = []
66  strech = []
67  bend = []
68  for i,line in enumerate(lines):
69      bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
70      strech += [np.array(line[0:].split(), dtype=float)[1]]
71      bend += [np.array(line[0:].split(), dtype=float)[2]]
72  ax=fig.add_subplot(1,2,1)
73  ax.plot(bondL0,strech,'yo',label='dr = 0.02 $\AA$ ')
74  ax.set_xlabel('Bond length ($\AA$)')
75  ax.set_ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
76  ax.legend(loc='best',prop={'size':10})
77  ax.text(2.56,1.86,'(a)',color='black',fontsize='19')
78
79
80  ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
81

```

```

82     ax=fig.add_subplot(122)
83     ax.plot(bondL0,bend,'yo',label='dr = 0.02 $\AA$')
84     ax.set_xlabel('Bond length ($\AA$)')
85     ax.set_ylabel('Stiffness of bond ($eV/\{\AA^2\}$)')
86     ax.legend(loc=4,prop={'size':10})
87     ax.text(2.56,-0.004,'(b)',color='black',fontsize=19)
88     plt.tight_layout()
89
90     ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
91     for ext in ['.png','.eps']:
92         plt.savefig('./images-si/Cu-Cu-fccradius'+ext,dpi=300)
93     plt.show()

```

---

## 2. Cu-Pd bond

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7  import matplotlib.pyplot as plt
8  from matplotlib.ticker import MaxNLocator
9  fig = plt.figure()
10
11
12 # displacement of atom = 0.05 Ang
13 filename = '../CuPd_fcc_encut500_is1_dr0.05/f_Cu-Pd.dat'
14 with open(filename) as f:
15     lines = f.readlines()
16     bondL0 = []
17     strech = []
18     bend = []
19     for i,line in enumerate(lines):
20         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
21         strech += [np.array(line[0:].split(), dtype=float)[1]]
22         bend += [np.array(line[0:].split(), dtype=float)[2]]
23     ax = fig.add_subplot(121)

```

```

24     ax.plot(bondL0,strech,'d',label='dr = 0.05 $\AA$')
25     ax=fig.add_subplot(122)
26     ax.plot(bondL0,bend,'d',label='dr = 0.05 $\AA$')
27
28     # displacement of atom = 0.2 Ang
29     filename = '../.../CuPd_fcc_enucut500_vib2_hp/f_Cu-Pd.dat'
30     with open(filename) as f:
31         lines = f.readlines()
32     bondL0 = []
33     strech = []
34     bend = []
35     for i,line in enumerate(lines):
36         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
37         strech += [np.array(line[0:].split(), dtype=float)[1]]
38         bend += [np.array(line[0:].split(), dtype=float)[2]]
39     ax=fig.add_subplot(121)
40     ax.plot(bondL0,strech,'d',label='dr = 0.2 $\AA$')
41     ax=fig.add_subplot(122)
42     ax.plot(bondL0,bend,'d',label='dr = 0.2 $\AA$')
43
44     # displacement of atom = 0.035 Ang
45     filename = '../.../CuPd_fcc_enucut500_is1_dr0.035/f_Cu-Pd.dat'
46     with open(filename) as f:
47         lines = f.readlines()
48     bondL0 = []
49     strech = []
50     bend = []
51     for i,line in enumerate(lines):
52         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
53         strech += [np.array(line[0:].split(), dtype=float)[1]]
54         bend += [np.array(line[0:].split(), dtype=float)[2]]
55     ax=fig.add_subplot(121)
56     ax.plot(bondL0,strech,'o',label='dr = 0.035 $\AA$')
57     ax=fig.add_subplot(122)
58     ax.plot(bondL0,bend,'o',label='dr = 0.035 $\AA$')
59
60
61     # displacement of atom = 0.02 Ang

```

```

62     filename = '../CuPd_fcc_encut500_is1_dr0.02/f_Cu-Pd.dat'
63     with open(filename) as f:
64         lines = f.readlines()
65     bondL0 = []
66     strech = []
67     bend = []
68     for i,line in enumerate(lines):
69         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
70         strech += [np.array(line[0:].split(), dtype=float)[1]]
71         bend += [np.array(line[0:].split(), dtype=float)[2]]
72     ax=fig.add_subplot(1,2,1)
73     ax.plot(bondL0,strech,'yo',label='dr = 0.02 $\AA$')
74     ax.set_xlabel('bond length ($\AA$)')
75     ax.set_ylabel('stiffness of bond ($eV/\{\AA^2\}$)')
76     ax.legend(loc='best',prop={'size':10})
77     ax.text(2.64,2.5,'(a)',color='black',fontsize='19')
78
79
80     ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
81
82     ax=fig.add_subplot(122)
83     ax.plot(bondL0,bend,'yo',label='dr = 0.02 $\AA$')
84     ax.set_xlabel('bond length ($\AA$)')
85     ax.set_ylabel('stiffness of bond ($eV/\{\AA^2\}$)')
86     ax.legend(loc=4,prop={'size':10})
87     ax.text(2.64,-0.018,'(b)',color='black',fontsize='19')
88     plt.tight_layout()
89
90     ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
91     for ext in ['.png','.eps']:
92         plt.savefig('./images-si/Cu-Pd-fcctestradius'+ext,dpi=300)
93     plt.show()

```

---

### 3. Pd-Pd bond

---

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *

```

```

4   from ase.visualize import view
5   from ase.lattice import bulk
6   import os
7   import matplotlib.pyplot as plt
8   from matplotlib.ticker import MaxNLocator
9   fig = plt.figure()
10
11
12  # displacement of atom = 0.05 Ang
13  filename = '../.../CuPd_fcc_enucut500_is1_dr0.05/f_Pd-Pd.dat'
14  with open(filename) as f:
15      lines = f.readlines()
16  bondL0 = []
17  strech = []
18  bend = []
19  for i,line in enumerate(lines):
20      bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
21      strech += [np.array(line[0:].split(), dtype=float)[1]]
22      bend += [np.array(line[0:].split(), dtype=float)[2]]
23  ax = fig.add_subplot(121)
24  ax.plot(bondL0,strech,'d',label='dr = 0.05 $\AA$')
25  ax=fig.add_subplot(122)
26  ax.plot(bondL0,bend,'d',label='dr = 0.05 $\AA$')
27
28  # displacement of atom = 0.2 Ang
29  filename = '../.../CuPd_fcc_enucut500_vib2_hp/f_Pd-Pd.dat'
30  with open(filename) as f:
31      lines = f.readlines()
32  bondL0 = []
33  strech = []
34  bend = []
35  for i,line in enumerate(lines):
36      bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
37      strech += [np.array(line[0:].split(), dtype=float)[1]]
38      bend += [np.array(line[0:].split(), dtype=float)[2]]
39  ax=fig.add_subplot(121)
40  ax.plot(bondL0,strech,'d',label='dr = 0.2 $\AA$')
41  ax=fig.add_subplot(122)

```

```

42     ax.plot(bondL0,bend,'d',label='dr = 0.2 $\AA$')
43
44     # displacement of atom = 0.035 Ang
45     filename = '../.../CuPd_fcc_enucut500_is1_dr0.035/f_Pd-Pd.dat'
46     with open(filename) as f:
47         lines = f.readlines()
48     bondL0 = []
49     strecth = []
50     bend = []
51     for i,line in enumerate(lines):
52         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
53         strecth += [np.array(line[0:].split(), dtype=float)[1]]
54         bend += [np.array(line[0:].split(), dtype=float)[2]]
55     ax=fig.add_subplot(121)
56     ax.plot(bondL0,strecth,'o',label='dr = 0.035 $\AA$')
57     ax=fig.add_subplot(122)
58     ax.plot(bondL0,bend,'o',label='dr = 0.035 $\AA$')
59
60
61     # displacement of atom = 0.02 Ang
62     filename = '../.../CuPd_fcc_enucut500_is1_dr0.02/f_Pd-Pd.dat'
63     with open(filename) as f:
64         lines = f.readlines()
65     bondL0 = []
66     strecth = []
67     bend = []
68     for i,line in enumerate(lines):
69         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
70         strecth += [np.array(line[0:].split(), dtype=float)[1]]
71         bend += [np.array(line[0:].split(), dtype=float)[2]]
72     ax=fig.add_subplot(1,2,1)
73     ax.plot(bondL0,strecth,'yo',label='dr = 0.02 $\AA$ ')
74     ax.set_xlabel('bond length ($\AA$)')
75     ax.set_ylabel('stiffness of bond ($eV/\{\AA^2\}$)')
76     ax.legend(loc='best',prop={'size':10})
77     ax.text(2.74,3.1,'(a)',color='black',fontsize='19')
78
79

```

```

80     ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
81
82     ax=fig.add_subplot(122)
83     ax.plot(bondL0,bend,'yo',label='dr = 0.02 $\AA$')
84     ax.set_xlabel('bond length ($\AA$)')
85     ax.set_ylabel('stiffness of bond (eV/\AA^2)')
86     ax.legend(loc=4,prop={'size':10})
87     ax.text(2.74,-0.07,'(b)',color='black',fontsize=19)
88     plt.tight_layout()
89
90     ax.xaxis.set_major_locator(MaxNLocator(nbins=5))
91     for ext in ['.png','.eps']:
92         plt.savefig('./images-si/Pd-Pd-fccstraduis'+ext,dpi=300)
93     plt.show()

```

---

### 9.8. SI plot stiffness vs bond length together

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7  import matplotlib.pyplot as plt
8  from matplotlib.ticker import MaxNLocator
9  fig = plt.figure(figsize=(12,10))
10
11 ax = fig.add_subplot(321)
12 def function(x):# fitted result from fitsusl
13
14     return -0.13061*x**6+3.44714*x**4-30.41924*x**2+89.85356
15
16
17 bondL0 = []
18 strech = []
19 bend=[]
20
21 filename = '../../../../../CuPd_bcc_enuct500_is1_dr0.02/f_Cu-Cu.dat'

```

```

22     with open(filename) as f:
23         lines = f.readlines()
24
25     for i,line in enumerate(lines):
26         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
27         strech += [np.array(line[0:].split(), dtype=float)[1]]
28         bend += [np.array(line[0:].split(), dtype=float)[2]]
29
30     coefs = np.polyfit(bondL0,strech,7)
31
32     x = np.linspace(min(bondL0),max(bondL0))
33     fit = np.polyval(coefs[::-1],x)
34
35     ax.plot(bondL0,strech,'x',label='bcc stretching')
36     ax.plot(bondL0,bend,'r*',label='bcc bending')
37     ax.plot(x,[function(i) for i in x],'b-')
38     ax.plot(x,[0 for i in x],'b-')
39     ax.set_xlabel('bond length ($\AA$)')
40     ax.set_ylabel('stiffness of bond (eV/$\AA^2$)')
41     ax.legend(loc='best',prop={'size':10})
42     ax.text(2.8,2,'Cu-Cu',fontsize=17)
43     ax.set_title('BCC')
44     ax.xaxis.set_major_locator(MaxNLocator(nbins=4))
45
46
47
48
49
50     ax = fig.add_subplot(322)
51
52     def function(x):
53
54         return -8.618*x+23.898 # fitted result from fitsusl
55
56     filename = '../../../../../CuPd_fcc_encut500_is1_dr0.035/f_Cu-Cu.dat'
57     with open(filename) as f:
58         lines = f.readlines()
59     bondL0 = []

```

```

60     strech = []
61     bend = []
62     for i,line in enumerate(lines):
63         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
64         strech += [np.array(line[0:].split(), dtype=float)[1]]
65         bend += [np.array(line[0:].split(), dtype=float)[2]]
66
67     ax.plot(bondL0,strech,'yo',label='fcc stretching ')
68     x = np.linspace(min(bondL0),max(bondL0))
69     fit = np.polyval(coefs[:, :],x)
70
71     ax.set_title('FCC')
72     ax.plot(x,[function(i) for i in x],'b-')
73     ax.plot(x,[0 for i in x],'b-')
74     ax.plot(bondL0,bend,'r*',label='fcc bending')
75     ax.set_xlabel('bond length ($\AA$)')
76     ax.set_ylabel('stiffness of bond ($eV/\AA^2$)')
77     ax.legend(loc='best',prop={'size':10})
78     ax.text(2.61,1.7,'Cu-Cu',fontsize=17)
79
80     ax.xaxis.set_major_locator(MaxNLocator(nbins=4))
81
82
83
84
85     ### Cu-Pd bond
86
87     ax = fig.add_subplot(323)
88     def function(x):
89
90         return -0.14861*x**6+4.14352*x**4-38.59792*x**2+120.31631 # fitted result from fitsusl
91
92
93     bondL0=[]
94     strech = []
95     bend = []
96     filename = '../CuPd_bcc_encut500_is1_dr0.02/f_Cu-Pd.dat'
97     with open(filename) as f:

```

```

98     lines = f.readlines()
99
100    for i,line in enumerate(lines):
101        bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
102        strech += [np.array(line[0:].split(), dtype=float)[1]]
103        bend += [np.array(line[0:].split(), dtype=float)[2]]
104
105    coefs = np.polyfit(bondL0,strech,6)
106
107    x = np.linspace(min(bondL0),max(bondL0))
108    fit = np.polyval(coefs[:],x)
109    #plt.subplot(1,3,2)
110    ax.plot(bondL0,strech,'x',label='bcc stretching')
111    ax.plot(bondL0,bend,'r*',label='bcc bending')
112    ax.plot(x,[function(i) for i in x],'b-')
113    ax.plot(x,[-0.00739*i**6+0.16496*i**4-6.54186*i+11.65808 for i in x],'b-')
114    ax.set_xlabel('bond length ($\AA$)')
115    ax.set_ylabel('stiffness of bond ($eV/\{\AA^2\}$)')
116    ax.text(2.8,3,'Cu-Pd',fontsize=17)
117    ax.legend(loc=1,prop={'size':10})
118
119    ax.xaxis.set_major_locator(MaxNLocator(nbins=4))
120
121
122    ax = fig.add_subplot(324)
123    def function(x):
124
125        return -8.975*x+26.043 # fitted result from fitsusl
126
127    filename = '../../../../../CuPd_fcc_encut500_is1_dr0.035/f_Cu-Pd.dat'
128    with open(filename) as f:
129        lines = f.readlines()
130    bondL0 = []
131    strech = []
132    bend = []
133    for i,line in enumerate(lines):
134        bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
135        strech += [np.array(line[0:].split(), dtype=float)[1]]
```

```

136     bend += [np.array(line[0:].split(), dtype=float)[2]]
137
138 ax.plot(bondL0,strech,'yo',label='fcc streching ')
139 x = np.linspace(min(bondL0),max(bondL0))
140 ax.plot(bondL0,bend,'r*',label='fcc bending')
141 ax.plot(x,[function(i) for i in x],'b-')
142 ax.plot(x,[-0.04554 for i in x],'b-')
143 ax.set_xlabel('bond length ($\AA$)')
144 ax.set_ylabel('stiffness of bond ($eV/\{\AA^2\}$)')
145 ax.text(2.7,2,'Cu-Pd',fontsize=17)
146 ax.legend(loc=1,prop={'size':10})
147
148 ax.xaxis.set_major_locator(MaxNLocator(nbins=4))
149
150
151
152
153 ax=fig.add_subplot(325)
154 def function(x):
155
156     return -0.22303*x**6+6.21903*x**4-58.27893*x**2+184.13498 # fitted result from fitsvsl
157
158
159 bondL0 = []
160 strech = []
161 bend = []
162 filename = '../CuPd_bcc_encut500_is1_dr0.02/f_Pd-Pd.dat'
163 with open(filename) as f:
164     lines = f.readlines()
165
166 for i,line in enumerate(lines):
167     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
168     strech += [np.array(line[0:].split(), dtype=float)[1]]
169     bend += [np.array(line[0:].split(), dtype=float)[2]]
170
171 coefs = np.polyfit(bondL0,strech,6)
172
173 x = np.linspace(min(bondL0),max(bondL0))

```

```

174     fit = np.polyval(coefs[:, :], x)
175
176     #ax = fig.add_subplot(1, 3, 3)
177     ax.plot(bondL0, strech, 'x', label='bcc stretching')
178     ax.plot(bondL0, bend, 'r*', label='bcc bending')
179     ax.plot(x, [function(i) for i in x], 'b-')
180     ax.plot(x, [0.0335*i**6-0.88482*i**4+7.79392*i**2-22.97742 for i in x], 'b-')
181     ax.set_xlabel('bond length ($\AA$)')
182     ax.set_ylabel('stiffness of bond ($eV/\{\AA^2\}$)')
183     ax.legend(loc=1, prop={'size': 10})
184     ax.text(2.9, 3.5, 'Pd-Pd', fontsize=17)
185     plt.tight_layout()
186
187     ax.xaxis.set_major_locator(MaxNLocator(nbins=4))
188
189     ax=fig.add_subplot(326)
190
191     def function(x):
192         return -13.876*x+40.95
193         # return -2.7236*x**2+23.45347
194     filename = '../CuPd_fcc_encut500_is1_dr0.035/f_Pd-Pd.dat'
195     with open(filename) as f:
196         lines = f.readlines()
197     bondL0 = []
198     strech = []
199     bend = []
200     for i, line in enumerate(lines):
201         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
202         strech += [np.array(line[0:].split(), dtype=float)[1]]
203         bend += [np.array(line[0:].split(), dtype=float)[2]]
204
205     x = np.linspace(min(bondL0), max(bondL0))
206     ax.plot(bondL0, strech, 'yo', label='fcc stretching')
207     ax.plot(bondL0, bend, 'r*', label='bcc bending')
208     ax.plot(x, [function(i) for i in x], 'b-')
209     ax.plot(x, [0.29828*i**2-2.42971 for i in x], 'b-')
210     ax.set_xlabel('bond length ($\AA$)')
211     ax.set_ylabel('stiffness of bond ($eV/\{\AA^2\}$)')

```

```

212     ax.legend(loc=1,prop={'size':10})
213     ax.text(2.775,2.5,'Pd-Pd',fontsize=17)
214     plt.tight_layout()
215
216     ax.xaxis.set_major_locator(MaxNLocator(nbins=4))
217
218     for ext in ['.png', '.eps']:
219         plt.savefig('./images-si/main-fit' + ext, dpi=300)
220
221     plt.show()

```

---

*9.9. Figure. 3 plot stiffness of stretching mode vs bond length*

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7  import matplotlib.pyplot as plt
8  from matplotlib.ticker import MaxNLocator
9  fig = plt.figure()
10
11 ax = fig.add_subplot(131)
12 def function(x):
13
14     return -0.13061*x**6+3.44714*x**4-30.41924*x**2+89.85356
15
16 filename = '../CuPd_fcc_encut500_is1_dr0.035/f_Cu-Cu.dat'
17 with open(filename) as f:
18     lines = f.readlines()
19 bondL0 = []
20 strech = []
21 bend = []
22 for i,line in enumerate(lines):
23     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
24     strech += [np.array(line[0:].split(), dtype=float)[1]]
25     bend += [np.array(line[0:].split(), dtype=float)[2]]

```

```

26
27 ax.plot(bondL0,streich,'yo',label='fcc stretching ')
28
29 bondL0 = []
30 streich = []
31 bend=[]
32
33 filename = '.../.../CuPd_bcc_enucut500_is1_dr0.02/f_Cu-Cu.dat'
34 with open(filename) as f:
35     lines = f.readlines()
36
37 for i,line in enumerate(lines):
38     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
39     streich += [np.array(line[0:].split(), dtype=float)[1]]
40     bend += [np.array(line[0:].split(), dtype=float)[2]]
41
42 coefs = np.polyfit(bondL0,streich,7)
43
44 x = np.linspace(min(bondL0),max(bondL0))
45 fit = np.polyval(coefs[:,],x)
46
47 ax.plot(bondL0,streich,'x',label='bcc stretching ')
48 ax.plot(x,[function(i) for i in x],'b-')
49 ax.set_xlabel('bond length ($\AA$)')
50 ax.set_ylabel('stiffness of bond ($eV/\{\AA^2\}$)')
51 ax.legend(loc='best',prop={'size':7})
52 ax.text(2.8,2.2,'Cu-Cu',fontsize=15)
53
54 ax.xaxis.set_major_locator(MaxNLocator(nbins=4))
55
56 ### Cu-Pd bond
57
58 def function(x):
59
60     return -0.14861*x**6+4.14352*x**4-38.59792*x**2+120.31631
61
62 ax=fig.add_subplot(1,3,2)
63

```

```

64
65 filename = '../../../../../CuPd_fcc_encut500_is1_dr0.035/f_Cu-Pd.dat'
66 with open(filename) as f:
67     lines = f.readlines()
68 bondL0 = []
69 strech = []
70 bend = []
71 for i,line in enumerate(lines):
72     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
73     strech += [np.array(line[0:].split(), dtype=float)[1]]
74     bend += [np.array(line[0:].split(), dtype=float)[2]]
75
76 ax.plot(bondL0,strech,'yo',label='fcc stretching ')
77 bondL0=[]
78 strech = []
79 bend = []
80 filename = '../../../../../CuPd_bcc_encut500_is1_dr0.02/f_Cu-Pd.dat'
81 with open(filename) as f:
82     lines = f.readlines()
83
84 for i,line in enumerate(lines):
85     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
86     strech += [np.array(line[0:].split(), dtype=float)[1]]
87     bend += [np.array(line[0:].split(), dtype=float)[2]]
88
89 coefs = np.polyfit(bondL0,strech,6)
90
91 x = np.linspace(min(bondL0),max(bondL0))
92 fit = np.polyval(coefs[:],x)
93 #plt.subplot(1,3,2)
94 ax.plot(bondL0,strech,'x',label='bcc stretching ')
95 ax.plot(x,[function(i) for i in x],'b-')
96 ax.set_xlabel('bond length ($\AA$)')
97 ax.set_ylabel('stiffness of bond ($eV/\{\AA^2\}$)')
98 ax.text(2.9,3,'Cu-Pd',fontsize=15)
99 ax.legend(loc=1,prop={'size':7})
100
101 ax.xaxis.set_major_locator(MaxNLocator(nbins=4))

```

```

102
103     ### Pd-Pd bond
104     ax=fig.add_subplot(133)
105     def function(x):
106
107         return -0.22303*x**6+6.21903*x**4-58.27893*x**2+184.13498
108
109     filename = '../CuPd_fcc_encut500_is1_dr0.035/f_Pd-Pd.dat'
110     with open(filename) as f:
111         lines = f.readlines()
112     bondL0 = []
113     strech = []
114     bend = []
115     for i,line in enumerate(lines):
116         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
117         strech += [np.array(line[0:].split(), dtype=float)[1]]
118         bend += [np.array(line[0:].split(), dtype=float)[2]]
119
120     plt.plot(bondL0,strech,'yo',label='fcc stretching ')
121     bondL0 = []
122     strech = []
123     bend = []
124     filename = '../CuPd_bcc_encut500_is1_dr0.02/f_Pd-Pd.dat'
125     with open(filename) as f:
126         lines = f.readlines()
127
128     for i,line in enumerate(lines):
129         bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
130         strech += [np.array(line[0:].split(), dtype=float)[1]]
131         bend += [np.array(line[0:].split(), dtype=float)[2]]
132
133     coefs = np.polyfit(bondL0,strech,6)
134
135     x = np.linspace(min(bondL0),max(bondL0))
136     fit = np.polyval(coefs[:],x)
137
138     ax.plot(bondL0,strech,'x',label='bcc stretching ')
139     ax.plot(x,[function(i) for i in x],'b-')

```

```

140 ax.set_xlabel('bond length ($\AA$)')
141 ax.set_ylabel('stiffness of bond ($eV/\{\AA^2\}$)')
142 ax.legend(loc=1,prop={'size':7})
143 ax.text(2.9,3.5,'Pd-Pd',fontsize=15)
144 plt.tight_layout()
145
146 ax.xaxis.set_major_locator(MaxNLocator(nbins=4))
147 for ext in ['.png', '.eps']:
148     plt.savefig('./images-si/combine-fit' + ext, dpi=300)
149
150 plt.show()

```

---

### 9.10. Figure. 5 plot effect of vibrational free energy

```

1 from ase import Atom, Atoms
2 import numpy as np
3 from jasp import *
4 from ase.visualize import view
5 from ase.lattice import bulk
6 import os
7 import matplotlib.pyplot as plt
8
9
10 plt.subplot(121)
11 # get reference energy
12 filename = '../../../../../CuPd_fcc_encut500_4000/ref_energy.in'
13 with open(filename) as f:
14     lines = f.readlines()
15 E_ref = []
16 for line in lines:
17     E_ref += [np.array(line[0:]).split(), dtype=float)[0]]
18
19 Cu_ref = E_ref[0]
20 Pd_ref = E_ref[1]
21
22
23 # CuPd_bcc
24 n = ['0', '614', '479', '3', '1']

```

```

25 energy = []
26
27 for a in n:
28     with jasp('.../CuPd_bcc_encut500_3000/{}'.format(a)) as calc:
29         atom = calc.get_atoms()
30         n = len(atom)
31         composition = 1.0
32         ele = atom.get_chemical_symbols()
33
34
35     for e in ele:
36         if e == 'Cu':
37             composition = composition - 1.0/n
38
39     try:
40         # print atoms0.get_forces()
41         energy.append((atom.get_potential_energy() - Cu_ref*len(atom)*(1-composition) - Pd_ref*composition*len(atom)))
42     except (VaspSubmitted, VaspQueued):
43         print '{0} not finished!'.format(a)
44
45 # obtain vibrational free energy
46 def Gfree_bcc(n):
47     filename = '.../final-data/vib-nobend-kpt4000/bcc-{}-fvib'.format(n)
48     with open(filename) as f:
49         lines = f.readlines()
50     G = []
51     for i,line in enumerate(lines):
52         if i==0 or i==300 or i ==600 or i==900:
53             G += [np.array(line[0:].split(), dtype=float)[0]]
54     return G
55
56 def Gfree_ref(n):
57     filename = '.../final-data/vib-nobend-kpt4000/fcc-{}-fvib'.format(n)
58     with open(filename) as f:
59         lines = f.readlines()
60     G = []
61     for i,line in enumerate(lines):
62         if i==0 or i==300 or i == 600 or i==900:
63             G += [np.array(line[0:].split(), dtype=float)[0]]
64     return G

```

```

63     ref_Cu = Gfree_ref(0)
64     ref_Pd = Gfree_ref(1)
65
66     n = [0,614,479,3,1]
67     composition = [0,0.25,0.375,0.5,1]
68     natom = [1,8,8,2,1]
69     G_bcc = []
70     for i in n:
71         G_bcc.append(Gfree_bcc(i))
72     # when T = OK
73     G_vib_0 = [G_bcc[i][0] for i in range(len(n))]
74     for i in range(len(G_vib_0)):
75         G_vib_0[i]=(G_vib_0[i]-(1-composition[i])*ref_Cu[0]-(composition[i])*ref_Pd[0])
76     energy_0 = np.array(energy)+np.array(G_vib_0)
77
78     # when T = 300K
79     G_vib_300 = [G_bcc[i][1] for i in range(len(n))]
80     for i in range(len(G_vib_300)):
81         G_vib_300[i]=(G_vib_300[i]-(1-composition[i])*ref_Cu[1]-(composition[i])*ref_Pd[1])
82     energy_300 = np.array(energy)+np.array(G_vib_300)
83
84     # when T = 600K
85     G_vib_600 = [G_bcc[i][2] for i in range(len(n))]
86     for i in range(len(G_vib_600)):
87         G_vib_600[i]=(G_vib_600[i]-(1-composition[i])*ref_Cu[2]-(composition[i])*ref_Pd[2])
88     energy_600 = np.array(energy)+np.array(G_vib_600)
89
90     # when T = 900K
91     G_vib_900 = [G_bcc[i][3] for i in range(len(n))]
92     for i in range(len(G_vib_900)):
93         G_vib_900[i]=(G_vib_900[i]-(1-composition[i])*ref_Cu[3]-(composition[i])*ref_Pd[3])
94     energy_900 = np.array(energy)+np.array(G_vib_900)
95
96     import matplotlib.pyplot as plt
97     compo = [0,0.25,0.375,0.5,0.75,1]
98     compo2 = [0,0.25,0.375,0.5,1]
99
100    plt.plot(compo2,energy,'x-',label='B2 without vibrational contribution (OK)')

```

```

101 plt.plot(compo2, energy_0,'x-',label='B2 with vibrational contribution (OK)')
102
103 plt.plot(compo2, energy_300,'x-',label='B2 with vibrational contribution (300K)')
104
105 plt.plot(compo2, energy_600,'x-',label='B2 with vibrational contribution (600K)')
106
107 plt.plot(compo2, energy_900,'x-',label='B2 with vibrational contribution (900K)')
108
109
110
111 plt.legend()
112 plt.xlabel('Pd composition')
113 plt.ylabel('Heat of formation (eV/atom)')
114
115 # draw fcc part
116 # get reference energy
117 filename = '../CuPd_fcc_encut500_4000/ref_energy.in'
118 with open(filename) as f:
119     lines = f.readlines()
120 E_ref = []
121 for line in lines:
122     E_ref += [np.array(line[0:]).split(), dtype=float)[0]]
123
124 Cu_ref = E_ref[0]
125 Pd_ref = E_ref[1]
126
127 # higher accuracy CuPd_fcc
128 n = ['0', '27', '455', '457', '28', '1']
129 energy = []
130
131 for a in n:
132     with jasp('../CuPd_fcc_encut500_4000/{0}'.format(a)) as calc:
133         atom = calc.get_atoms()
134         n = len(atom)
135         composition = 1.0
136         ele = atom.get_chemical_symbols()
137
138

```

```

139     for e in ele:
140         if e == 'Cu':
141             composition = composition - 1.0/n
142     try:
143         #      print atoms0.get_forces()
144         energy.append((atom.get_potential_energy() - Cu_ref*len(atom)*(1-composition)-Pd_ref*composition*len(atom)))
145     except (VaspSubmitted, VaspQueued):
146         print '{0} not finished!'.format(a)
147
148
149 # obtain vibrational free energy
150 def Gfree_fcc(n):
151     filename = '.../.../final-data/vib-nobend-kpt4000/fcc-{}-fvib'.format(n)
152     with open(filename) as f:
153         lines = f.readlines()
154     G = []
155     for i,line in enumerate(lines):
156         if i==0 or i==300 or i == 600 or i==900:
157             G += [np.array(line[0:]).split(), dtype=float)[0]]
158     return G
159
160
161 ref_Cu = Gfree_fcc(0)
162 ref_Pd = Gfree_fcc(1)
163
164 n = [0,27,455,457,28,1]
165 composition = [0,0.25,0.375,0.5,0.75,1]
166 natom = [1,4,8,8,4,1]
167 G_fcc = []
168 for i in n:
169     G_fcc.append(Gfree_fcc(i))
170 # when T = OK
171 G_vib_0 = [G_fcc[i][0] for i in range(len(n))]
172 for i in range(len(G_vib_0)):
173     G_vib_0[i]=(G_vib_0[i]-(1-composition[i])*ref_Cu[0]-(composition[i])*ref_Pd[0])
174 energy_0 = np.array(energy)+np.array(G_vib_0)
175
176 # when T = 300K

```

```

177 G_vib_300 = [G_fcc[i][1] for i in range(len(n))]
178 for i in range(len(G_vib_300)):
179     G_vib_300[i]=(G_vib_300[i]-(1-composition[i])*ref_Cu[1]-(composition[i])*ref_Pd[1])
180 energy_300 = np.array(energy)+np.array(G_vib_300)
181
182 # when T = 600K
183 G_vib_600 = [G_fcc[i][2] for i in range(len(n))]
184 for i in range(len(G_vib_600)):
185     G_vib_600[i]=(G_vib_600[i]-(1-composition[i])*ref_Cu[2]-(composition[i])*ref_Pd[2])
186 energy_600 = np.array(energy)+np.array(G_vib_600)
187
188 # when T = 900K
189 G_vib_900 = [G_fcc[i][3] for i in range(len(n))]
190 for i in range(len(G_vib_900)):
191     G_vib_900[i]=(G_vib_900[i]-(1-composition[i])*ref_Cu[3]-(composition[i])*ref_Pd[3])
192 energy_900 = np.array(energy)+np.array(G_vib_900)
193
194 import matplotlib.pyplot as plt
195 compo = [0,0.25,0.375,0.5,0.75,1]
196 compo2 = [0,0.25,0.375,0.5,1]
197
198
199 plt.plot(compo, energy,'o-',label='fcc without F_{vib} (OK)')
200
201 plt.plot(compo, energy_0,'o-',label='fcc with F_{vib} (OK)')
202
203 plt.plot(compo, energy_300,'o-',label='fcc with F_{vib} (300K)')
204
205 plt.plot(compo, energy_600,'o-',label='fcc with F_{vib} (600K)')
206
207 plt.plot(compo, energy_900,'o-',label='fcc with F_{vib} (900K)')
208
209
210 plt.legend(loc='best',prop={'size':10})
211 plt.xlabel('Pd composition')
212 plt.ylabel('Heat of formation (eV/atom)')
213
214
```

```

215 plt.subplot(1,2,2)
216 def Gfree_bcc(n):
217     filename = ' ../../final-data/vib-nobend-kpt4000/bcc-{}-fvib'.format(n)
218     with open(filename) as f:
219         lines = f.readlines()
220     G = []
221     for i,line in enumerate(lines):
222         G += [np.array(line[0:].split(), dtype=float)[0]]
223     return G
224 def Gfree_ref(n):
225     filename = ' ../../final-data/vib-nobend-kpt4000/fcc-{}-fvib'.format(n)
226     with open(filename) as f:
227         lines = f.readlines()
228     G = []
229     for i,line in enumerate(lines):
230         G += [np.array(line[0:].split(), dtype=float)[0]]
231     return G
232
233 ref_Cu = np.array(Gfree_ref(0))
234 ref_Pd = np.array(Gfree_ref(1))
235
236
237 composition = [0,0.25,0.375,0.5,1]
238 G_bcc = np.array(Gfree_bcc(3))
239
240 G_l12 = np.array(Gfree_ref(27))
241 G_614 = np.array(Gfree_bcc(614))
242 G_fcc05 = np.array(Gfree_ref(457))
243 G_bcc = G_bcc
244 G_l12 = G_l12
245 G_fcc05 = G_fcc05
246 G_614 = G_614
247 import matplotlib.pyplot as plt
248 T = np.linspace(0,1000,1001)
249
250 plt.plot(T,G_bcc,'b-')
251 plt.plot(T,G_fcc05,'g-')
252 plt.plot(T,G_l12,'r-')

```

```

253 plt.plot(T,G_614,'y-')
254 plt.legend(['B2 lattice (50 at% Pd)', 'fcc lattice (50 at% Pd)', 'fcc lattice (25 at% Pd)', 'B2 lattice (25 at% Pd)'], loc=1)
255 plt.xlabel('Temperature (K)')
256 plt.ylabel('Normalized vibrational free energy (eV/atom)')
257 plt.tight_layout()
258 for ext in ['.png', '.eps']:
259     plt.savefig('./images-si/subplot' + ext, dpi=300)
260
261
262 plt.show()

```

---

### 9.11. SI plot different functionals

```

1  #+BEGIN_SRC python
2
3  from ase import Atom, Atoms
4  import numpy as np
5  from jasp import *
6  from ase.visualize import view
7  from ase.lattice import bulk
8  import os
9  import matplotlib.pyplot as plt
10 f=plt.figure(figsize=(12,8))
11
12 # get reference energy
13 with jasp('../.../convergence/Cu-fcc-PBE-kppra8000') as calc:
14     atom = calc.get_atoms()
15     Cu_ref = atom.get_potential_energy()
16 with jasp('../.../convergence/Pd-fcc-PBE-kppra10000') as calc:
17     atom = calc.get_atoms()
18     Pd_ref = atom.get_potential_energy()
19
20
21
22 # higher accuracy CuPd_fcc
23 n = ['Cu-fcc-PBE-kppra8000', 'Cu3Pd-fcc-PBE-kppra10000', 'CuPd455-fcc-PBE-kppra10000', 'CuPd457-fcc-PBE-kppra10000', 'CuPd
24 PBE_fcc = []
25

```

```

26     for a in n:
27         with jasp('.../.../convergence/{0}'.format(a)) as calc:
28             atom = calc.get_atoms()
29             n = len(atom)
30             composition = 1.0
31             ele = atom.get_chemical_symbols()
32
33
34             for e in ele:
35                 if e == 'Cu':
36                     composition = composition - 1.0/n
37             try:
38
39                 PBE_fcc.append((atom.get_potential_energy() - Cu_ref*len(atom)*(1-composition)-Pd_ref*composition*len(atom)))
40             except (VaspSubmitted, VaspQueued):
41                 print '{0} not finished!'.format(a)
42
43     # CuPd_bcc
44     n = ['Cu-bcc-PBE-kppra8000', 'CuPd614-bcc-PBE-kppra8000', 'CuPd494-bcc-PBE-kppra8000', 'CuPd-bcc-PBE-kppra8000', 'Pd-bcc-PB
45     PBE_bcc = []
46
47     for a in n:
48         with jasp('.../.../convergence/{0}'.format(a)) as calc:
49             atom = calc.get_atoms()
50             n = len(atom)
51             composition = 1.0
52             ele = atom.get_chemical_symbols()
53
54
55             for e in ele:
56                 if e == 'Cu':
57                     composition = composition - 1.0/n
58             try:
59
60                 PBE_bcc.append((atom.get_potential_energy() - Cu_ref*len(atom)*(1-composition)-Pd_ref*composition*len(atom)))
61             except (VaspSubmitted, VaspQueued):
62                 print '{0} not finished!'.format(a)
63

```

```

64  # get reference energy
65  with jasp('.../.../convergence/Cufcc-PBESOL-kpt23') as calc:
66      atom = calc.get_atoms()
67      Cu_ref = atom.get_potential_energy()
68  with jasp('.../.../convergence/Pdfcc-PBESOL-kpt24') as calc:
69      atom = calc.get_atoms()
70      Pd_ref = atom.get_potential_energy()
71
72
73
74  # higher accuracy CuPd_fcc
75  n = ['Cufcc-PBESOL-kpt23', 'Cu3Pdfcc-PBESOL-kpt17', 'Cu5Pd3fcc-isif3-PBESOL-kpt17', 'Cu4Pd4A1fcc-PBESOL-kpt16', 'Cu3Pd5A1f'
76  PBESol_fcc = []
77
78  for a in n:
79      with jasp('.../.../convergence/{0}'.format(a)) as calc:
80          atom = calc.get_atoms()
81          n = len(atom)
82          composition = 1.0
83          ele = atom.get_chemical_symbols()
84
85
86          for e in ele:
87              if e == 'Cu':
88                  composition = composition - 1.0/n
89
90      try:
91          # print atoms0.get_forces()
92          PBESol_fcc.append((atom.get_potential_energy() - Cu_ref*len(atom)*(1-composition) - Pd_ref*composition*len(at
93      except (VaspSubmitted, VaspQueued):
94          print '{0} not finished!'.format(a)
95
96  # CuPd_bcc
97  n = ['Cubcc-PBESol-kpt22', 'CuPdbcc-614-PBESol-kpt17', 'CuPdbcc-494-PBESol-kpt14', 'CuPdbcc-3-PBESol-kpt13', 'Pdbcc-PBESol
98
99  PBESol_bcc = []
100
101  for a in n:
102      with jasp('.../.../convergence/{0}'.format(a)) as calc:
103          atom = calc.get_atoms()

```

```

102     n = len(atom)
103     composition = 1.0
104     ele = atom.get_chemical_symbols()
105
106
107     for e in ele:
108         if e == 'Cu':
109             composition = composition - 1.0/n
110
111     try:
112
113         PBEsol_bcc.append((atom.get_potential_energy() -Cu_ref*len(atom)*(1-composition)-Pd_ref*composition*len(at
114     except (VaspSubmitted, VaspQueued):
115         print '{0} not finished!'.format(a)
116
117     # get reference energy
118     with jasp('.../.../.../convergence/Cufcc-AM05-kpt22') as calc:
119         atom = calc.get_atoms()
120         Cu_ref = atom.get_potential_energy()
121     with jasp('.../.../.../convergence/Pdfcc-AM05-kpt24') as calc:
122         atom = calc.get_atoms()
123         Pd_ref = atom.get_potential_energy()
124
125
126
127     n = ['Cufcc-AM05-kpt22', 'Cu3Pdfcc-AM05-kpt20', 'Cu5Pd3fcc-isif3-AM05-kpt15', 'Cu4Pd4A1fcc-AM05-kpt14', 'Cu3Pd5A1fcc-AM05-k
128     AM05_fcc = []
129
130     for a in n:
131         with jasp('.../.../.../convergence/{0}'.format(a)) as calc:
132             atom = calc.get_atoms()
133             n = len(atom)
134             composition = 1.0
135             ele = atom.get_chemical_symbols()
136
137
138             for e in ele:
139                 if e == 'Cu':

```



```

178 LDA_fcc = []
179
180 for a in n:
181     with jasp('.../.../.../convergence/{0}'.format(a)) as calc:
182         atom = calc.get_atoms()
183         n = len(atom)
184         composition = 1.0
185         ele = atom.get_chemical_symbols()
186
187
188     for e in ele:
189         if e == 'Cu':
190             composition = composition - 1.0/n
191     try:
192         #      print atoms0.get_forces()
193         LDA_fcc.append((atom.get_potential_energy() - Cu_ref*len(atom)*(1-composition)-Pd_ref*composition*len(atom)))
194     except (VaspSubmitted, VaspQueued):
195         print '{0} not finished!'.format(a)
196
197 n = ['Cubcc-LDA-kpt24', 'CuPd614bcc-LDA-kpt16', 'CuPd494bcc-LDA-kpt19', 'CuPdbcc-LDA-kpt13', 'Pdbcc-LDA-kpt23']
198 LDA_bcc = []
199
200 for a in n:
201     with jasp('.../.../.../convergence/{0}'.format(a)) as calc:
202         atom = calc.get_atoms()
203         n = len(atom)
204         composition = 1.0
205         ele = atom.get_chemical_symbols()
206
207
208     for e in ele:
209         if e == 'Cu':
210             composition = composition - 1.0/n
211     try:
212         #      print atoms0.get_forces()
213         LDA_bcc.append((atom.get_potential_energy() - Cu_ref*len(atom)*(1-composition)-Pd_ref*composition*len(atom)))
214     except (VaspSubmitted, VaspQueued):
215         print '{0} not finished!'.format(a)

```

```

216
217
218
219
220
221 import matplotlib.pyplot as plt
222 ax = f.add_subplot(131)
223 ax.plot(PBE_fcc, PBEsol_fcc,'md',label='gs of fcc (PBEsol)')
224
225 ax.plot(PBE_bcc, PBEsol_bcc,'mx',label='gs of bcc (PBEsol)')
226 ax.plot(PBE_bcc, AM05_bcc,'gx',label= 'gs of bcc (AM05)')
227 ax.plot(PBE_fcc,AM05_fcc,'go',label='gs of fcc (AM05)')
228 ax.plot(PBE_fcc,LDA_fcc,'ro',label='gs of fcc (LDA)')
229 ax.plot(PBE_bcc,LDA_bcc,'rx',label='gs of bcc (LDA)')
230
231
232 ax.plot(np.linspace(0.07,-0.13),np.linspace(0.07,-0.13),'b-',label='PBE')
233 ax.set_xlim([-0.13,0.07])
234 ax.set_ylim([-0.13,0.07])
235 plt.legend(loc='best',prop={'size':8})
236
237 ax.set_xlabel('Heat of formation (PBE) ($eV/atom$)',fontsize=12)
238 ax.set_ylabel('Heat of formation ($eV/atom$)',fontsize=12)
239
240 plt.tight_layout()
241 ax=f.add_subplot(132)
242 composition_fcc = [0,0.25,0.375,0.5,0.625,0.75,1]
243 composition_bcc = [0,0.25,0.375,0.5,1]
244 ax.plot(composition_fcc, PBE_fcc,'ro-',label='PBE fcc')
245 ax.plot(composition_fcc,LDA_fcc,'bo-',label='LDA fcc')
246 ax.plot(composition_fcc,PBEsol_fcc,'go-',label='PBEsol fcc')
247 ax.plot(composition_fcc,AM05_fcc,'yo-',label='AM05 fcc')
248 ax.set_xlabel('Composition of Pd (at%)')
249 ax.set_ylabel('Heat of formation ($eV/atom$)')
250 ax.set_ylim([-0.125,0.05])
251 plt.tight_layout()
252 plt.legend(loc='best',prop={'size':10})
253 ax=f.add_subplot(133)

```

```

254 ax.plot(composition_bcc,PBE_bcc,'ro-',label='PBE bcc')
255 ax.plot(composition_bcc,LDA_bcc,'bo-',label='LDA bcc')
256 ax.plot(composition_bcc,PBESol_bcc,'go-',label='PBESol bcc')
257 ax.plot(composition_bcc,AM05_bcc,'yo-',label='AM05 bcc')
258 ax.set_xlabel('Composition of Pd (at%)')
259 ax.set_ylabel('Heat of formation ($eV/atom$)')
260 ax.set_ylim([-0.125,0.05])
261 plt.tight_layout()
262 plt.legend(loc='best',prop={'size':10})
263
264 for ext in ['.png','.eps']:
265     plt.savefig('./images-si/different-functional'+ext,dpi=300)
266 plt.show()

```

---

## 9.12. SI use more accurate force calculation based on $Ismear=1$

### 9.12.1. Cu-Cu bond

```

1 from ase import Atom, Atoms
2 import numpy as np
3 from jasp import *
4 from ase.visualize import view
5 from ase.lattice import bulk
6 import os
7 import matplotlib.pyplot as plt
8
9 # displacement of atom = 0.02 Ang
10 filename = '../CuPd_bcc_dr0.02_kpt10/f_Cu-Cu.dat'
11 with open(filename) as f:
12     lines = f.readlines()
13 bondL0 = []
14 strech = []
15 bend = []
16 for i,line in enumerate(lines):
17     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
18     strech += [np.array(line[0:].split(), dtype=float)[1]]
19     bend += [np.array(line[0:].split(), dtype=float)[2]]
20 plt.subplot(1,2,1)

```

```

21 plt.plot(bondL0,strech,'d',label='dr = 0.02 $\AA$ (ismear=-5)')
22 plt.subplot(1,2,2)
23 plt.plot(bondL0,bend,'d',label='dr = 0.02 $\AA$ (ismear=-5)')
24
25
26
27
28
29 # displacement of atom is 0.02 Ang
30 filename = '../../../../../CuPd_bcc_dr0.02_ismear1/f_Cu-Cu.dat'
31 with open(filename) as f:
32     lines = f.readlines()
33 bondL0 = []
34 strech = []
35 bend = []
36 for i,line in enumerate(lines):
37     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
38     strech += [np.array(line[0:].split(), dtype=float)[1]]
39     bend += [np.array(line[0:].split(), dtype=float)[2]]
40 plt.subplot(1,2,1)
41 plt.plot(bondL0,strech,'ko',label='dr = 0.02 $\AA$ (ismear=1)')
42 plt.xlabel('bond length ($\AA$)')
43 plt.ylabel('stiffness of bond ($eV/\AA^2$)')
44 plt.legend(loc='best',prop={'size':10})
45 plt.subplot(1,2,2)
46 plt.plot(bondL0,bend,'ko',label='dr = 0.02 $\AA$ (ismear=1)')
47 plt.xlabel('bond length ($\AA$)')
48 plt.ylabel('stiffness of bond ($eV/\AA^2$)')
49 plt.legend(loc='best',prop={'size':10})
50 plt.tight_layout()
51
52
53
54 for ext in ['.png','.eps']:
55     plt.savefig('../images-si/Cu-Cu-ismear'+ext,dpi=300)
56 plt.show()

```

---

### 9.12.2. Cu-Pd bond

```
1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7  import matplotlib.pyplot as plt
8
9  # displacement of atom = 0.02 Ang
10 filename = '../../../../../CuPd_bcc_dr0.02_kpt10/f_Cu-Pd.dat'
11 with open(filename) as f:
12     lines = f.readlines()
13 bondL0 = []
14 strech = []
15 bend = []
16 for i,line in enumerate(lines):
17     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
18     strech += [np.array(line[0:].split(), dtype=float)[1]]
19     bend += [np.array(line[0:].split(), dtype=float)[2]]
20 plt.subplot(1,2,1)
21 plt.plot(bondL0,strech,'d',label='dr = 0.02 $\\AA$ (ismear=-5)')
22 plt.subplot(1,2,2)
23 plt.plot(bondL0,bend,'d',label='dr = 0.02 $\\AA$ (ismear=-5)')
24
25
26
27
28
29 # displacement of atom is 0.02 Ang
30 filename = '../../../../../CuPd_bcc_dr0.02_ismear1/f_Cu-Pd.dat'
31 with open(filename) as f:
32     lines = f.readlines()
33 bondL0 = []
34 strech = []
35 bend = []
36 for i,line in enumerate(lines):
```

```

37     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
38     strech += [np.array(line[0:].split(), dtype=float)[1]]
39     bend += [np.array(line[0:].split(), dtype=float)[2]]
40     plt.subplot(1,2,1)
41     plt.plot(bondL0,strech,'ko',label='dr = 0.02 $\\AA$ (ismear=1)')
42     plt.xlabel('bond length ($\\AA$)')
43     plt.ylabel('stiffness of bond ($eV/\\AA^2$)')
44     plt.legend(loc='best',prop={'size':10})
45     plt.subplot(1,2,2)
46     plt.plot(bondL0,bend,'ko',label='dr = 0.02 $\\AA$ (ismear=1)')
47     plt.xlabel('bond length ($\\AA$)')
48     plt.ylabel('stiffness of bond ($eV/\\AA^2$)')
49     plt.legend(loc='best',prop={'size':10})
50     plt.tight_layout()
51
52
53
54     for ext in ['.png','.eps']:
55         plt.savefig('./images-si/Cu-Pd-ismear'+ext,dpi=300)
56     plt.show()

```

---

### 9.12.3. Pd-Pd bond

```

1  from ase import Atom, Atoms
2  import numpy as np
3  from jasp import *
4  from ase.visualize import view
5  from ase.lattice import bulk
6  import os
7  import matplotlib.pyplot as plt
8
9  # displacement of atom = 0.02 Ang
10 filename = '../../../../../CuPd_bcc_dr0.02_kpt10/f_Pd-Pd.dat'
11 with open(filename) as f:
12     lines = f.readlines()
13 bondL0 = []
14 strech = []
15 bend = []

```

```

16   for i,line in enumerate(lines):
17       bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
18       strech += [np.array(line[0:].split(), dtype=float)[1]]
19       bend += [np.array(line[0:].split(), dtype=float)[2]]
20   plt.subplot(1,2,1)
21   plt.plot(bondL0,strech,'d',label='dr = 0.02 $\\AA$ (ismear=-5)')
22   plt.subplot(1,2,2)
23   plt.plot(bondL0,bend,'d',label='dr = 0.02 $\\AA$ (ismear=-5)')
24
25
26
27
28 filename = '../../../../../CuPd_bcc_dr0.02_ismear1/f_Pd-Pd.dat'
29 with open(filename) as f:
30     lines = f.readlines()
31 bondL0 = []
32 strech = []
33 bend = []
34 for i,line in enumerate(lines):
35     bondL0 += [np.array(line[0:].split(), dtype=float)[0]]
36     strech += [np.array(line[0:].split(), dtype=float)[1]]
37     bend += [np.array(line[0:].split(), dtype=float)[2]]
38 plt.subplot(1,2,1)
39 plt.plot(bondL0,strech,'ko',label='dr = 0.02 $\\AA$ (ismear=1)')
40 plt.xlabel('bond length ($\\AA$)')
41 plt.ylabel('stiffness of bond ($eV/\\AA^2$)')
42 plt.legend(loc='best',prop={'size':10})
43 plt.subplot(1,2,2)
44 plt.plot(bondL0,bend,'ko',label='dr = 0.02 $\\AA$ (ismear=1)')
45 plt.xlabel('bond length ($\\AA$)')
46 plt.ylabel('stiffness of bond ($eV/\\AA^2$)')
47 plt.legend(loc='best',prop={'size':10})
48 plt.tight_layout()
49
50
51
52 for ext in ['.png','.eps']:
53     plt.savefig('../images-si/Pd-Pd-ismear'+ext,dpi=300)

```

```
54 plt.show()
```

---

### 9.13. SI Compare convex hull of fcc to bcc

```
1 from ase.io import read
2 import os
3 f = open('../../../CuPd_bcc_enuct500_3000/gs.out', 'r')
4 lines = f.readlines()
5 f.close()
6
7 gs_bcc_concentration,gs_bcc_energy,gs_bcc_fitenergy,gs_bcc_index = [],[],[],[]
8 for line in lines:
9     fields = line.split()
10    if len(fields) == 4:
11        c,e,fe,i = line.split()
12    elif len(fields) == 7: #nmaps
13        c, c1,c2,e,fe,err,i = line.split()
14        gs_bcc_concentration.append(float(c))
15        gs_bcc_energy.append(float(e))
16        gs_bcc_fitenergy.append(float(fe))
17        gs_bcc_index.append(int(i))
18
19
20 f = open('../../../CuPd_fcc_enuct500_4000/gs.out', 'r')
21 lines = f.readlines()
22 f.close()
23
24 gs_fcc_concentration,gs_fcc_energy,gs_fcc_fitenergy,gs_fcc_index = [],[],[],[]
25 for line in lines:
26     fields = line.split()
27    if len(fields) == 4:
28        c,e,fe,i = line.split()
29    elif len(fields) == 7:
30        c, c1,c2,e,fe,err,i = line.split()
31        gs_fcc_concentration.append(float(c))
32        gs_fcc_energy.append(float(e))
33        gs_fcc_fitenergy.append(float(fe))
34        gs_fcc_index.append(int(i))
```

```

35
36
37 import matplotlib.pyplot as plt
38
39 plt.plot(gs_bcc_concentration,gs_bcc_fitenergy,'ro-',label="bcc convex hull")
40
41 plt.plot(gs_fcc_concentration,gs_fcc_fitenergy,'yo-',label="fcc convex hull")
42 plt.plot([0.25,0.5],[-0.103, -0.121],'b--')
43
44 plt.text(0.23,-0.113,'L12')
45 plt.text(0.36,-0.121,'A1')
46 plt.text(0.49,-0.128,'B2')
47 plt.xlabel('$x_{\text{Pd}}$')
48 plt.ylabel('Heat of Formation (eV/atom)')
49 plt.ylim([-0.13,0.1])
50 plt.tight_layout()
51 plt.legend(loc='best')
52 str = './images-si/CuPd-convexhull-compare.png'
53 plt.savefig(str)
54 plt.show()

```

---

## 9.14. Construction of phase diagram

### 9.14.1. without vibrational free energy

```

1 from vaspy import Vasp
2 from ase.io import read
3 import os
4 import matplotlib.pyplot as plt
5 plt.figure(figsize=(10,8))
6 f = open('../raw-data/novib-0-0.5.txt','r')
7 lines = f.readlines()
8 f.close()
9
10 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
11 for line in lines:
12     fields = line.split()
13     if len(fields) == 4:

```

```

14         c,e,fe,i = line.split()
15     else:
16         print "error"
17     x_l12.append(float(c))
18     T_l12.append(float(e))
19     x_b2.append(float(fe))
20     T_b2.append(int(i))
21 f = open('../raw-data/novib-0.5-0.75.txt','r')
22 lines=f.readlines()
23 f.close()
24
25 x_3,T_3,x_4,T_4 = [],[],[],[]
26 for line in lines:
27     fields = line.split()
28     if len(fields) == 4:
29         c,e,fe,i = line.split()
30     else:
31         print "error"
32     x_3.append(float(c))
33     T_3.append(float(e))
34     x_4.append(float(fe))
35     T_4.append(int(i))
36
37 plt.plot(x_l12,T_l12,'k-')
38 plt.plot(x_b2,T_b2,'k-')
39 plt.plot(x_3,T_3,'k-')
40 plt.plot(x_4,T_4,'k-')
41 plt.plot([0.499999,0.70943],[400,400],'k-',label='CE+MC without $F_{vib}$')
42 plt.plot([0.4834,0.289685],[375,375],'k-')
43 plt.xlim([0.25,0.75])
44
45 ##### plot experimental data
46 L1x = [0.41194,0.45075,0.47463,0.49552,0.53433,0.6,0.62985,0.74328]
47 L1y = [860.89552,844.77612,817.91045,791.04478,726.56716,570.74627,495.52239,200]
48 L2x = [0.41194,0.43284,0.44776,0.4597,0.48358,0.49254,0.49851,0.49851]
49 L2y = [860,828.65672,791.04478,742.68657,656.71642,511.64179,425.67164,200]
50 L3x = [0.41194,0.38209,0.3403,0.42687]
51 L3y = [860,823.28358,667.46269,200]

```

```

52 L4x = [0.41194,0.3791,0.34328,0.32239,0.31045,0.30149,0.28358,0.27761,0.2806]
53 L4y = [860,855.52239,807.16418,769.55224,737.31343,705.07463,667.46269,667.46269,200]
54 L5x = [0.28358,0.3403]
55 L5y = [667.46269,667.46269]
56 plt.plot(L1x,L1y,'r-',label='Mei Li et al (2008)')
57 plt.plot(L2x,L2y,'r-')
58 plt.plot(L3x,L3y,'r-')
59 plt.plot(L4x,L4y,'r-')
60 plt.plot(L5x,L5y,'r-')
61 plt.xlabel('$x_{\%s}$' %'Pd')
62 plt.ylabel('Temperature (K)')
63 plt.legend()
64 # add label
65
66 plt.text(0.3,200,'L12(Cu, Pd) + B2(Cu,Pd)')
67 plt.text(0.53,200,'L12(Cu,Pd) + B2(Cu,Pd)')
68 plt.text(0.34,400,'disordered fcc + B2')
69 plt.text(0.52,430,'disordered fcc + B2')
70 plt.text(0.48,550,'B2')
71
72 for ex in ['.png','.eps']:
73     plt.savefig('./images-si/figure1'+ex,dpi=300)
74 plt.show()

```

---

### 9.14.2. with vibrational free energy

```

1 from vasp import Vasp
2 from ase.io import read
3 import os
4 import matplotlib.pyplot as plt
5 from matplotlib.legend_handler import HandlerLine2D
6 fig,ax=plt.subplots(figsize=(7,6))
7
8 f = open('../raw-data/novib-0-0.5.txt','r')
9 lines = f.readlines()
10 f.close()
11
12 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]

```

```

13   for line in lines:
14       fields = line.split()
15       if len(fields) == 4:
16           c,e,fe,i = line.split()
17       else:
18           print "error"
19       x_l12.append(float(c))
20       T_l12.append(float(e))
21       x_b2.append(float(fe))
22       T_b2.append(int(i))
23
24
25 ax.plot(x_l12,T_l12,'k-')
26 ax.plot(x_b2,T_b2,'k-')
27 ax.plot([0.499999,0.70943],[400,400],'k-',label='CE+MC without $F_{vib}$')
28 plt.xlim([0.25,0.5])
29
30 ##### plot phase diagram with vibrational free energy
31
32
33 f = open('../raw-data/dt10-vib.txt','r')
34 lines = f.readlines()
35 f.close()
36
37 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
38 for line in lines:
39     fields = line.split()
40     if len(fields) == 4:
41         c,e,fe,i = line.split()
42     else:
43         print "error"
44     x_l12.append(float(c))
45     T_l12.append(float(e))
46     x_b2.append(float(fe))
47     T_b2.append(int(i))
48
49
50 ax.plot(x_l12,T_l12,'r-')

```

```

51 ax.plot(x_b2,T_b2,'r-')
52 ax.plot([0.33,0.4827],[375,375],'k-')
53 ax.plot([0.334,0.473],[350,350],'r-',label='CE+MC with $F_{vib}$')
54 line3=ax.plot(0.479,700,'ro',label='Critical point with $F_{vib}$')
55 line4=ax.plot(0.49,720,'ko',label = 'Critical point without $F_{vib}$')
56 plt.xlim([0.25,0.5])
57
58
59 ax.set_xlabel('$x_{\{s\}}$ %Pd')
60 ax.set_ylabel('Temperature (K)')
61 plt.legend(loc='best',handler_map={line3: HandlerLine2D(numpoints=1),line4: HandlerLine2D(numpoints=1)},prop={'size':12})
62 # add label
63 ax.text(0.33,200,'L12(Cu, Pd) + B2(Cu,Pd)')
64 ax.text(0.35,420,'disordered fcc + B2')
65 ax.text(0.48,550,'B2')
66
67 for ex in ['.png','.eps']:
68     plt.savefig('../images-si/figure3'+ex,dpi=300)
69 plt.show()

```

---

### 9.14.3. effect of thermal expansion

```

1 from vaspy import Vasp
2 from ase.io import read
3 import os
4 import matplotlib.pyplot as plt
5 from matplotlib.legend_handler import HandlerLine2D
6 fig,ax=plt.subplots(figsize=(7,6))
7
8 f = open('../raw-data/thermalexpand0%.txt','r')
9 lines = f.readlines()
10 f.close()
11
12 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
13 for line in lines:
14     fields = line.split()
15     if len(fields) == 4:
16         c,e,fe,i = line.split()

```

```

17     else:
18         print "error"
19     x_l12.append(float(c))
20     T_l12.append(float(e))
21     x_b2.append(float(fe))
22     T_b2.append(int(i))
23
24
25 ax.plot(x_l12,T_l12,'k-',label='Thermal expand 0%')
26 ax.plot(x_b2,T_b2,'k-')
27 plt.xlim([0.25,0.5])
28
29 ##### plot phase diagram with vibrational free energy
30
31
32 f = open('../raw-data/thermalexpand2%.txt','r')
33 lines = f.readlines()
34 f.close()
35
36 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
37 for line in lines:
38     fields = line.split()
39     if len(fields) == 4:
40         c,e,fe,i = line.split()
41     else:
42         print "error"
43     x_l12.append(float(c))
44     T_l12.append(float(e))
45     x_b2.append(float(fe))
46     T_b2.append(int(i))
47
48
49 ax.plot(x_l12,T_l12,'r-',label='Thermal expand 2%')
50 ax.plot(x_b2,T_b2,'r-')
51 plt.xlim([0.25,0.5])
52
53
54

```

```

55     f = open('../raw-data/dt10-vib.txt','r')
56     lines = f.readlines()
57     f.close()
58
59     x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
60
61     for line in lines:
62         fields = line.split()
63         if len(fields) == 4:
64             c,e,fe,i = line.split()
65         else:
66             print "error"
67         x_l12.append(float(c))
68         T_l12.append(float(e))
69         x_b2.append(float(fe))
70         T_b2.append(int(i))
71
72     ax.plot(x_l12,T_l12,'b-',label='Thermal expand 1%')
73     ax.plot(x_b2,T_b2,'b-')
74     plt.xlim([0.25,0.5])
75     plt.legend(loc='best')
76     ax.plot([0.334,0.473],[350,350],'k-')
77
78
79     ax.text(0.33,200,'L12(Cu, Pd) + B2(Cu,Pd)')
80     ax.text(0.35,420,'disordered fcc + B2')
81     ax.text(0.48,550,'B2')
82
83
84
85     ax.set_xlabel('$x_{\text{$_{\%s$}}}$ %Pd')
86     ax.set_ylabel('Temperature (K)')
87
88     # add label
89
90
91     for ex in ['.png','.eps']:
92         plt.savefig('../images-si/thermalexpand'+ex,dpi=300)

```

```
93 plt.show()
```

---

#### 9.14.4. effect of temperature interval

##### 1. without vibrational free energy

```
1  from vasp import Vasp
2  from ase.io import read
3  import os
4  import matplotlib.pyplot as plt
5  from matplotlib.legend_handler import HandlerLine2D
6  fig,ax=plt.subplots(figsize=(7,6))
7
8  f = open('../raw-data/novib-0-0.5.txt','r')
9  lines = f.readlines()
10 f.close()
11
12 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
13 for line in lines:
14     fields = line.split()
15     if len(fields) == 4:
16         c,e,fe,i = line.split()
17     else:
18         print "error"
19     x_l12.append(float(c))
20     T_l12.append(float(e))
21     x_b2.append(float(fe))
22     T_b2.append(int(i))
23
24
25 ax.plot(x_l12,T_l12,'k-',label='dT = 25 K')
26 ax.plot(x_b2,T_b2,'k-')
27 plt.xlim([0.25,0.5])
28
29 ##### plot phase diagram with vibrational free energy
30
31
32 f = open('../raw-data/novib-dt10.txt','r')
```

```

33     lines = f.readlines()
34     f.close()
35
36     x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
37     for line in lines:
38         fields = line.split()
39         if len(fields) == 4:
40             c,e,fe,i = line.split()
41         else:
42             print "error"
43         x_l12.append(float(c))
44         T_l12.append(float(e))
45         x_b2.append(float(fe))
46         T_b2.append(int(i))
47
48
49     ax.plot(x_l12,T_l12,'r-',label='dT = 10 K')
50     ax.plot(x_b2,T_b2,'r-')
51     plt.xlim([0.25,0.5])
52
53
54
55     f = open('../raw-data/novib-dt5.txt','r')
56     lines = f.readlines()
57     f.close()
58
59     x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
60     for line in lines:
61         fields = line.split()
62         if len(fields) == 4:
63             c,e,fe,i = line.split()
64         else:
65             print "error"
66         x_l12.append(float(c))
67         T_l12.append(float(e))
68         x_b2.append(float(fe))
69         T_b2.append(int(i))
70

```

```

71
72     ax.plot(x_l12,T_l12,'b-',label='dT = 5 K')
73     ax.plot(x_b2,T_b2,'b-')
74     plt.xlim([0.25,0.5])
75     plt.legend(loc='best')
76     ax.plot([0.334,0.485],[375,375],'k-')
77     ax.text(0.33,200,'L12(Cu, Pd) + B2(Cu,Pd)')
78     ax.text(0.36,400,'disordered fcc + B2')
79     ax.text(0.48,550,'B2')
80
81     ax.set_xlabel('$x_{\$s\$} \%Pd$')
82     ax.set_ylabel('Temperature (K)')
83
84     # add label
85
86
87     for ex in ['.png','.eps']:
88         plt.savefig('./images-si/novib-dT'+ex,dpi=300)
89     plt.show()

```

---

## 2. with vibrational free energy

```

1  from vasp import Vasp
2  from ase.io import read
3  import os
4  import matplotlib.pyplot as plt
5  from matplotlib.legend_handler import HandlerLine2D
6  fig,ax=plt.subplots(figsize=(7,6))
7
8  f = open('../raw-data/vib-dt25.txt','r')
9  lines = f.readlines()
10 f.close()
11
12 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
13 for line in lines:
14     fields = line.split()
15     if len(fields) == 4:
16         c,e,fe,i = line.split()

```

```

17     else:
18         print "error"
19     x_l12.append(float(c))
20     T_l12.append(float(e))
21     x_b2.append(float(fe))
22     T_b2.append(int(i))
23
24
25 ax.plot(x_l12,T_l12,'k-',label='dT = 25 K')
26 ax.plot(x_b2,T_b2,'k-')
27 plt.xlim([0.25,0.5])
28
29 ##### plot phase diagram with vibrational free energy
30
31
32 f = open('../raw-data/dt10-vib.txt','r')
33 lines = f.readlines()
34 f.close()
35
36 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
37 for line in lines:
38     fields = line.split()
39     if len(fields) == 4:
40         c,e,fe,i = line.split()
41     else:
42         print "error"
43     x_l12.append(float(c))
44     T_l12.append(float(e))
45     x_b2.append(float(fe))
46     T_b2.append(int(i))
47
48
49 ax.plot(x_l12,T_l12,'r-',label='dT = 10 K')
50 ax.plot(x_b2,T_b2,'r-')
51 plt.xlim([0.25,0.5])
52
53
54

```

```

55     f = open('../raw-data/vib-dt5.txt','r')
56     lines = f.readlines()
57     f.close()
58
59     x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
60
61     for line in lines:
62         fields = line.split()
63         if len(fields) == 4:
64             c,e,fe,i = line.split()
65         else:
66             print "error"
67         x_l12.append(float(c))
68         T_l12.append(float(e))
69         x_b2.append(float(fe))
70         T_b2.append(int(i))
71
72     ax.plot(x_l12,T_l12,'b-',label='dT = 5 K')
73     ax.plot(x_b2,T_b2,'b-')
74     plt.xlim([0.25,0.5])
75     plt.legend(loc='best')
76     ax.plot([0.334,0.473],[350,350],'k-')
77
78
79     ax.text(0.33,200,'L12(Cu, Pd) + B2(Cu,Pd)')
80     ax.text(0.35,420,'disordered fcc + B2')
81     ax.text(0.48,550,'B2')
82
83
84
85     ax.set_xlabel('$x_{\text{--}}$ %Pd')
86     ax.set_ylabel('Temperature (K)')
87
88     # add label
89
90
91     for ex in ['.png','.eps']:
92         plt.savefig('./images-si/temperature-interval'+ex,dpi=300)

```

```
93 plt.show()
```

---

#### 9.14.5. effect of super-cell radius

```
1 from vasp import Vasp
2 from ase.io import read
3 import os
4 import matplotlib.pyplot as plt
5 from matplotlib.legend_handler import HandlerLine2D
6 fig,ax=plt.subplots(figsize=(7,6))
7
8 f = open('../raw-data/novib-0-0.5.txt','r')
9 lines = f.readlines()
10 f.close()
11
12 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
13 for line in lines:
14     fields = line.split()
15     if len(fields) == 4:
16         c,e,fe,i = line.split()
17     else:
18         print "error"
19     x_l12.append(float(c))
20     T_l12.append(float(e))
21     x_b2.append(float(fe))
22     T_b2.append(int(i))
23
24
25 ax.plot(x_l12,T_l12,'k-',label='Super-cell radius = 35 $\AA$')
26 ax.plot(x_b2,T_b2,'k-')
27 plt.xlim([0.25,0.5])
28
29 ##### plot phase diagram with vibrational free energy
30
31
32 f = open('../raw-data/novib-er42.txt','r')
33 lines = f.readlines()
34 f.close()
```

```

35
36 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
37 for line in lines:
38     fields = line.split()
39     if len(fields) == 4:
40         c,e,fe,i = line.split()
41     else:
42         print "error"
43     x_l12.append(float(c))
44     T_l12.append(float(e))
45     x_b2.append(float(fe))
46     T_b2.append(int(i))
47
48
49 ax.plot(x_l12,T_l12,'r-',label='Super-cell radius = 42 $\AA$')
50 ax.plot(x_b2,T_b2,'r-')
51 plt.xlim([0.25,0.5])
52
53
54
55 f = open('../raw-data/novib-er50.txt','r')
56 lines = f.readlines()
57 f.close()
58
59 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
60 for line in lines:
61     fields = line.split()
62     if len(fields) == 4:
63         c,e,fe,i = line.split()
64     else:
65         print "error"
66     x_l12.append(float(c))
67     T_l12.append(float(e))
68     x_b2.append(float(fe))
69     T_b2.append(int(i))
70
71
72 ax.plot(x_l12,T_l12,'b-',label='Super-cell radius = 50 $\AA$')

```

```

73     ax.plot(x_b2,T_b2,'b-')
74     plt.xlim([0.25,0.5])
75     plt.legend(loc='best')
76     ax.plot([0.334,0.485],[375,375],'k-')
77     ax.text(0.33,200,'L12(Cu, Pd) + B2(Cu,Pd)')
78
79     ax.text(0.36,400,'disordered fcc + B2')
80
81     ax.text(0.48,550,'B2')
82     plt.legend(loc=2,prop={'size':12})
83     ax.set_xlabel('$x_{\%s}$' %'Pd')
84     ax.set_ylabel('Temperature (K)')
85
86     # add label
87
88
89     for ex in ['.png','.eps']:
90         plt.savefig('./images-si/er'+ex,dpi=300)
91     plt.show()

```

---

#### 9.14.6. effect of error in DFT calculation

```

1  from vaspy import Vasp
2  from ase.io import read
3  import os
4  import matplotlib.pyplot as plt
5  from matplotlib.legend_handler import HandlerLine2D
6  fig,ax=plt.subplots(figsize=(7,5))
7
8  f = open('../git/fgeng/raw-data/novib-0-0.5.txt','r')
9  lines = f.readlines()
10 f.close()
11
12 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
13 for line in lines:
14     fields = line.split()
15     if len(fields) == 4:
16         c,e,fe,i = line.split()

```

```

17     else:
18         print "error"
19     x_l12.append(float(c))
20     T_l12.append(float(e))
21     x_b2.append(float(fe))
22     T_b2.append(int(i))
23
24
25 ax.plot(x_l12,T_l12,'k-',label='Additional 0 meV/atom')
26 ax.plot(x_b2,T_b2,'k-')
27 plt.xlim([0.25,0.5])
28
29 ##### plot phase diagram with vibrational free energy
30
31
32 f = open('../git/fgeng/raw-data/additionimeV.txt','r')
33 lines = f.readlines()
34 f.close()
35
36 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
37 for line in lines:
38     fields = line.split()
39     if len(fields) == 4:
40         c,e,fe,i = line.split()
41     else:
42         print "error"
43     x_l12.append(float(c))
44     T_l12.append(float(e))
45     x_b2.append(float(fe))
46     T_b2.append(int(i))
47
48
49 ax.plot(x_l12,T_l12,'r-',label='Additional -1 meV/atom')
50 ax.plot(x_b2,T_b2,'r-')
51 plt.xlim([0.25,0.5])
52
53
54

```

```

55 f = open('../git/fgeng/raw-data/addition2meV.txt','r')
56 lines = f.readlines()
57 f.close()
58
59 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
60 for line in lines:
61     fields = line.split()
62     if len(fields) == 4:
63         c,e,fe,i = line.split()
64     else:
65         print "error"
66     x_l12.append(float(c))
67     T_l12.append(float(e))
68     x_b2.append(float(fe))
69     T_b2.append(int(i))
70
71
72 ax.plot(x_l12,T_l12,'b-',label='Additional -2 meV/atom')
73 ax.plot(x_b2,T_b2,'b-')
74 plt.xlim([0.25,0.5])
75
76
77 f = open('../git/fgeng/raw-data/addition3meV.txt','r')
78 lines = f.readlines()
79 f.close()
80
81 x_l12,T_l12,x_b2,T_b2 = [],[],[],[]
82 for line in lines:
83     fields = line.split()
84     if len(fields) == 4:
85         c,e,fe,i = line.split()
86     else:
87         print "error"
88     x_l12.append(float(c))
89     T_l12.append(float(e))
90     x_b2.append(float(fe))
91     T_b2.append(int(i))
92

```

```
93
94 ax.plot(x_l12,T_l12,'g-',label='Additional -3 meV/atom')
95 ax.plot(x_b2,T_b2,'g-')
96 plt.legend(loc=2,prop={'size':12})
97 ax.plot([0.334,0.485],[375,375],'k-')
98 ax.text(0.33,200,'L12(Cu, Pd) + B2(Cu,Pd)')
99 ax.text(0.36,400,'disordered fcc + B2')
100 ax.text(0.48,550,'B2')
101
102 ax.set_xlabel('$x_{\%s}$' %'Pd')
103 ax.set_ylabel('Temperature (K)')
104
105 for ex in ['.png','.eps']:
106     plt.savefig('./images-si/changeb2'+ex,dpi=300)
107 plt.show()
```

---