2020

Quaternary Hydrides Pd1-y-zAgyCuzHx Embedded Atom Method Potentials for Hydrogen Energy Applications

Chaonan Zhang
zhang127@marshall.edu

QUATERNARY HYDRIDES $Pd_{1-y-z}Ag_yCu_zH_x$ EMBEDDED ATOM METHOD POTENTIALS FOR HYDROGEN ENERGY APPLICATIONS

A thesis submitted to
the Graduate College of
Marshall University
In partial fulfillment of
the requirements for the degree of
Master of Science
In
Mechanical Engineering
by
Chaonan Zhang
Approved by
Dr. Iyad Hijazi, Committee Chairperson
Dr. Gang Chen
Dr. Roozbeh Ross Salary

Marshall University
December 2020

APPROVAL OF THESIS

We, the faculty supervising the work of Chaonan Zhang, affirm that the thesis, "*Quaternary Hydrides $Pd_{1-y-z}Ag_yCu_zH_x$ Embedded Atom Potential Method Potentials for Hydrogen Energy Applications*", meets the high academic standards for original scholarship and creative work established by the College of Engineering and Computer Science for the Master of Science in Mechanical Engineering. This work also conforms to the editorial standards of our discipline and the Graduate College of Marshall University. With our signatures, we approve the manuscript for publication.

Dr. Iyad Hijazi, Weisberg Division of Engineering Committee Chairperson Date 12/04/2020

Dr. Gang Chen, Weisberg Division of Engineering Committee Member Date 12/4/2020

Dr. Roozbeh Ross Salary, Weisberg Division of Engineering Committee Member Date 12/04/2020

ACKNOWLEDGMENTS

This research will not be done without the great help and advice from my advisor Dr. Iyad Hijazi. Also, I would like to thank all the employees from Weisberg Division of Engineering and the Marshall University library who supported me during my thesis.

TABLE OF CONTENTS

List of Tables viii

List of Figures ix

Abstract xi

CHAPTER 1 Introduction 1

CHAPTER 2 Modeling 5

2.1 The Potentials 5

CHAPTER 3 Fitting and Validation of Palladium Silver Copper Alloys 8

3.1 Ag and Cu Fitting Parameters 8

3.2 Validation 10

CHAPTER 4 Fitting and Validation of Palladium Silver and Palladium Copper Hydrides 15

4.1 DFT Calculations 15

4.2 Ag-H and Cu-H Fitting 20

4.3 Validation 22

4.3.1 Lattice Constants and Cohesive Energy 23

4.3.2 Bulk Modulus and Elastic Constants 24

4.4 Additional Compositions 26

4.4.1 Dynamic Stability 30

4.4.2 Miscibility Gap and Gibbs Free Energy of Mixing 31

CHAPTER 5 Conclusion 34

References 36

APPENDIX A: Institutional Review Board Letter. 46

APPENDIX B: Calculations for Interstitial Solid Solution 47

Mixing Rule Pair Potential Derivates 50
Total Energy 51
Cohesive Energy 51
Bulk Modulus 52
Solid Solution Bulk Modulus 53
Stress 54
Solid Solution Stress 54
Elastic Constants 55
Vacancy Formation Energy 56
Gibbs Free Energy Mixing 56
APPENDIX C: Code Used to Calculate Data in this Thesis 58
SIESTA Code 58
$Pd_{0.75}Cu_{0.25}H_{1.00}$.fdf 58
LAMMPS Code 60
Elastic Constants Calculation 60
in.elastic_PdAgCuH0666 60
init.mod-PdAgCuH-0.666 64
potential.mod 65
NPT.mod 66
MATLAB Program for Fitting the Model Parameters 66
Fit.m 66
objfun.m 80
Check_Fit.m 133

MATLAB Program for Generating Dynamo Format Potential File..... 197

LIST OF TABLES

Table 1. Pd, Ag and Cu fitting parameters..... 8

Table 2. Fitting results for Ag and Cu 8

Table 3: Pd-Ag, Pd-Cu and Ag-Cu heat of solution and scaling factors from fitting..... 10

Table 4. Experimental values used in shifting ab initio data 17

Table 5. Shifting factors for Pd-Ag-H and Pd-Cu-H ab initio data 17

Table 6. Pd-Ag-H and Pd-Cu-H ab initio data, fitting results, and MD results..... 19

Table 7. Fitting parameters for Pd-H, Cu-H and Ag-H 21

Table 8. Fitting parameters for H-H 21

LIST OF FIGURES

Figure 1: Ag and Cu cohesive energy and Rose et al. equation of state. 9

Figure 2: $Pd_{1-x}Ag_x$ and Pd_xCu_{1-x} alloys lattice constants and cohesive energies from MD, experiments, and DFT. 12

Figure 3: $Pd_{1-x}Ag_x$ and Pd_xCu_{1-x} alloys C_{11}, C_{12} elastic constant and bulk modulus from MD, experiment, and DFT. 13

Figure 4: $Pd_{1-x}Ag_x$ and Pd_xCu_{1-x} alloys C_{44} and C' elastic constant from MD, experiment, and DFT. 14

Figure 5: H (light grey) interstitial sites within the Pd (grey) and Ag or Cu (black) lattice. 16

Figure 6: $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ shifted DFT results comparison with Hale et al. 18

Figure 7: Pair potential functions of the fitted H-H, Pd-H, Ag-H and Cu-H potentials. 22

Figure 8: Pair potential functions of the fitted Pd-Pd, Ag-Ag, Cu-Cu, Pd-Ag, Pd-Cu and Ag-Cu potentials. 22

Figure 9: $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ lattice constants and cohesive energies from MD and fitting data. 24

Figure 10: $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ C_{11}, C_{12} elastic constant and bulk modulus from MD. 25

Figure 11: $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ C_{44} elastic constant and C' shear elastic constant from MD. 26

Figure 12: $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$, $Pd_{0.50}Ag_{0.50}H_x$, $Pd_{0.75}Cu_{0.25}H_x$, $Pd_{0.50}Cu_{0.50}H_x$ and $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ lattice constants and cohesive energies from MD. 27

Figure 13: $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$, $Pd_{0.50}Ag_{0.50}H_x$, $Pd_{0.75}Cu_{0.25}H_x$, $Pd_{0.50}Cu_{0.50}H_x$ and $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ C_{11}, C_{12} elastic constant and bulk modulus from MD. 28

Figure 14: $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$, $Pd_{0.50}Ag_{0.50}H_x$, $Pd_{0.75}Cu_{0.25}H_x$, $Pd_{0.50}Cu_{0.50}H_x$ and $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ C_{44} and C' elastic constant from MD. 29

Figure 15: $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$, $Pd_{0.50}Ag_{0.50}H_x$, $Pd_{0.75}Cu_{0.25}H_x$, $Pd_{0.50}Cu_{0.50}H_x$ and $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ young's modulus, shear modulus, and Poisson's ratio. 30

Figure 16: (a) TE structure before simulation. interstitial H (red) within the Pd (blue), Ag (green) and Cu (yellow) lattice. (b) OC structure after simulation. 31

Figure 17: Gibbs free energy plot for different structures at 0 K. 33

Figure 18: Gibbs free energy plot for different structures at 300 K. 33

ABSTRACT

The Pd-H system has attracted extensive attention. Pd can absorb considerable H at room temperature, this ability is reversible, so it is suitable for multiple energy applications. Pd-Ag alloys possess higher H permeability, solubility and narrower miscibility gap with better mechanical properties than pure Pd, but sulfur poisoning remains an issue. Pd-Cu alloys have excellent resistance to sulfur and carbon monoxide poisoning and hydrogen embrittlement, good mechanical properties, and broader temperature working environments over pure Pd, but relatively lower hydrogen permeability and solubility than pure Pd and Pd-Ag alloys. This suggests that alloying Pd with Ag and Cu to create Pd-Ag-Cu ternary alloys can optimize the overall performance and substantially lowers the cost. Thus, in this research, the first embedded atom method potentials for the quaternary hydrides $Pd_{1-y-z}Ag_yCu_zH_x$ were provided. The fully analytical potentials are fitted utilizing the central atom method without performing time-consuming molecular dynamics simulations.

CHAPTER 1

INTRODUCTION

The Pd-H system has attracted various studies [1-8]. Pd possesses the capacity to absorb a considerable H at room temperature and atmospheric pressure. It has excellent hydrogen selectivity, permeability and H diffusivity with high mobility [9-12]. Therefore, it has diverse applications such as fuel cells, hydrogen storage, refrigeration, hydrogen separation and purification, nuclear radiation adsorption and catalytic converter [9-15].

However, the coexistence of low H alpha (α) and high H beta (β) phases in Pd hydride causes a miscibility gap. The strain caused by the lattice mismatch between the two phases increases the possibility of mechanical failure [16-18]. In addition, Pd is vulnerable to hydrogen embrittlement after some cycles of α phase and β phase transformations in Pd membranes or absorption/desorption cycles in hydrogen storage [18, 19]. Pd membranes are also prone to H_2S and CO poisoning [20-23]. Finally, palladium is an expensive noble metal, with a price currently higher than gold.

Pd can be alloyed with Ag and Cu to solve the above listed problems and reduce costs [24-26]. Although H is practically insoluble in Ag, Pd-Ag alloys possess higher H solubility than Pd [25], with the largest solubility in 20-40% Ag content, at 400 K and 1 atm pressure [27, 28]. As with Pd, Pd-Ag alloys are 100% selective when absorbing H, and therefore, are also ideal for separating H from mixture of gases [18]. Pd-Ag alloys also possess excellent H permeability, with the greatest permeability observed in the $Pd_{77}Ag_{23}$ structure [24, 29]. Depending on Ag concentration, H diffusion rate can be also increased in Pd-Ag alloys as compared with pure Pd [5]. Pd-Ag alloys are reported to possess more steady mechanical properties than pure Pd, and

better resistance to H embrittlement [18], however, strong poisoning by H_2S remains an issue [30].

Pd-Cu alloys are also known as one of the most effective materials in hydrogen separation field, as Cu is a much cheaper metal than Pd. Pd-Cu alloys also have better mechanical properties, high thermal stability [31], avoidance of hydrogen embrittlement at room temperature, improved sulfur poisoning resistance than Pd and Pd-Ag alloys [32], and more resistance to carbon monoxide poisoning than pure Pd [33, 34]. In addition, Pd-Cu alloys are also completely selective for hydrogen [35], and H diffusion rate in body-centered-cubic (bcc) Pd-Cu alloys is faster than those in pure Pd and face-centered-cubic (fcc) Pd-Cu alloys [36, 37]. However, H permeation rate in bcc Pd-Cu has been found to be slower than in pure Pd, but faster than in fcc Pd-Cu [38], with the optimal permeation value at Cu ~ 47.5 % [39]. In addition, H solubility in Pd-Cu alloy is lower than pure Pd, and decreases strongly as Cu content increases [38].

Since Pd-Ag alloys are known for their excellent H selectivity, permeability and solubility, and Pd-Cu alloys have excellent resistance to sulfur and carbon monoxide poisoning and hydrogen embrittlement, good mechanical properties, broader working conditions than pure Pd and relatively low cost, one way to optimize the overall performance and substantially lower the cost is alloying Pd with Ag and Cu. It has been found the H selectivity in H_2/N_2 mixture gas is higher in ternary Pd-Ag-Cu alloys than those in binary fcc Pd-Cu [40], the H permeability increases when substituting small amount of Ag for Cu in Pd-Cu alloy [41]. Ling et al. concluded H permeability increases with the addition of Ag for a fixed Cu concentration [42]. The H permeability for Pd-Ag-Cu is approximately twice the reported permeability for fcc Pd-Cu alloys with a similar Pd composition [43]. Ling et al. reported the H solubility increase in Pd-Ag-Cu

alloys with increasing Ag concentration for a fixed concentration of Cu, however, the H solubility decreases as the Cu content is increased [42]. Ling et al. also concluded that H diffusion rate in Pd-Cu-Ag alloys decreases with increasing Ag concentration in Pd-Ag-Cu alloys with low Cu content. However, for alloys with high Cu content, diffusivity first increases with Ag concentration, then decreases at high Ag concentration [42]. As with Pd-Cu, some Pd-Ag-Cu alloy compositions have good sulfur resistance [44]. Zhao et al. reported the addition of Ag to Pd-Cu membranes always weakens the sulfur resistance of Pd-Ag-Cu alloys, but for some Pd-Ag-Cu compositions, H permeance remains at least 50 % higher than in Pd-Cu membranes under the same H_2S testing conditions [44]. This indicates that efficient ternary Pd-Ag-Cu membranes, which exhibit significantly improved H_2 permeation rates in the presence of H_2S, can be obtained through careful balancing of Cu and Ag content. Finally, results show that Pd-Ag-Cu membranes possess stable H permeation flux at 400 °C after 300 h on stream, indicating that Pd-Ag-Cu alloys have good thermal stability [40].

To accelerate the search for Pd-Ag-Cu alloys with optimal hydrogen selectivity, permeability, diffusivity, absorption, thermal stability and resistance to sulfur and CO poisoning, an accurate and efficient modeling of the Pd-Ag-Cu-H system is needed. The most reliable simulation technique utilizes first principle (ab initio) calculations, but its high processing costs make it infeasible for simulations containing a large number of atoms. Molecular dynamics (MD) simulations using the embedded atom method (EAM) offers an efficient way to investigate alloys with larger atomic structures. EAM is well suitable to model binary and ternary hydrides with metallic crystal structures and interstitial H atoms. An EAM Pd-H potential which can predict the miscibility gap was formulated by Zhou et al. [4]. The Pd-H EAM potential was then expanded into Pd-Ag-H ternary potential by Hale et al. [5]. Their Pd-Ag-H potential predicted

smooth changes in structures parameters, elastic properties and energy with increasing H concentrations, H sites occupation shift and the disappearance of the miscibility gap by the addition of Ag at 300 K. However, their model was built on Foiles et al.'s Pd potential, which is available in a tabular form but does not include a full explanation of the analytical form and parameters [45], and therefore impeding further improvement of the Pd-Ag-H ternary system.

In previous work, Pd-H and Pd-Ag-H EAM potentials with fewer fitting parameters than Zhou et al. [4] and Hale et al. [5] were fitted by Hijazi et al. [46] and Fuller and Hijazi [47] that can predict many of the properties of the Pd-H and Pd-Ag-H structures accurately. In this research, the previously developed EAM Pd-H and Pd-Ag-H potentials are expanded into the Pd-Ag-Cu-H quaternary EAM potentials. First principles simulations using SIESTA were carried out to obtain the fitting parameters. The central atom method is used during fitting procedure, without performing time-consuming MD simulations.

CHAPTER 2

MODELING

2.1 The Potentials

A total of 18 functions are needed to create the Pd-Ag-Cu-H quaternary atomic EAM potentials, which include 4 embedding energy functions, F_{Pd}, F_{Ag}, F_{Cu} and F_H, 4 electron density functions, ρ_{Pd}, ρ_{Ag}, ρ_{Cu} and ρ_H, and 10 pair functions, φ_{Pd-Pd}, φ_{Ag-Ag}, φ_{Cu-Cu}, φ_{H-H}, φ_{Pd-Ag}, φ_{Pd-Cu}, φ_{Ag-Cu}, $\varphi_{Pd-H,}$ φ_{Ag-H} and φ_{Cu-H}. In EAM, each atom is embedded into a lattice that includes all host atoms. The pair potential between atoms, and the energy related to embedding an atom inside the host lattice is modeled. The total energy E_{tot} of an EAM system is given by [48]:

$$E_{tot} = \sum_{i=1}^{N} F_i(\rho_i) + \frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N} \varphi_{ij}(r_{ij}) \tag{1}$$

$$\rho_i = \sum_{j\neq i}^{N} f_j(r_{ij}) \tag{2}$$

where ρ_i is the electron density for atom i, F_i is the embedding energy, ϕ_{ij} is the pair potential between atom i and atom j, r_{ij} represents the distance from atom i and to atom j, f_j is the electron density function of distance from the center of atom j. The EAM model by Hijazi and Park [49-53] was used in fitting the Pd-Pd, Ag-Ag, Cu-Cu, Pd-Ag, Pd-Cu and Ag-Cu potentials. The embedding function is represented by:

$$F(\rho) = F(\rho_e)\left[1 - \eta \ln\left(\frac{\rho}{\rho_e}\right)\right]\left(\frac{\rho}{\rho_e}\right)^{\eta} \tag{3}$$

The host electron density is given by:

$$f = f_e e^{-\chi(r-r_e)} \tag{4}$$

where f_e is a scaling factor that can be obtained from $f_e = E_c/\Omega$, Ω is the atomic volume and E_c is the cohesive energy, r_e is the equilibrium closest distance, and a fitting parameter χ. The pair potential function is the modified potential created by Rose et al. [54] and has the form:

$$\phi = -\phi_e\left[1+\delta(r/r_e-1)\right]e^{-\beta(r/r_e-1)} \tag{5}$$

where ϕ_e, δ, and β are the 3 adjustable parameters. Therefore, for an fcc metal, there are 6 fitting parameters χ, ϕ_e, δ, β, η, and ρ_e.

The generalized Morse potential function proposed by Zhou et al. [4] and Hale et al. [5] was used in fitting the H-H, Pd-H, Ag-H and Cu-H pair potentials, and has the form:

$$\varphi_{HH}(r) = D_{HH}(\beta_{HH}e^{-\alpha_{HH}(r-r_{0,HH})} - \alpha_{HH}e^{\beta_{HH}(r-r_{0,HH})}) \tag{6}$$

where D, α, β, and r_0 are fitting parameters, r_0 defines the interatomic spacing between two atoms, and D ($\beta - \alpha$) is the binding energy. The H electron density function is given by:

$$\rho_H(r) = C_H e^{-\delta_H \cdot r} \tag{7}$$

which has 2 fitting parameters C_H and δ_H, while the embedding function for H has the form:

$$F_{H,u}(\rho) = -C_H \cdot \left(\frac{1}{2+d_H}\cdot\left(\rho+\varepsilon_H\right)^{2+d_H} - \frac{a_H+b_H}{1+d_H}\cdot\left(\rho+\varepsilon_H\right)^{1+dH} + \frac{a_H\cdot b_H}{d_H}\cdot\left(\rho+\varepsilon_H\right)^{d_H} \right) \tag{8}$$

where a_H, b_H, c_H, d_H are fitting parameters, and ε_H = 0.0540638.

The EAM total energy is a linear summation of the embedding energy and the pair potentials. A unique feature of the elemental EAM potential is that it will not change due to the transformation of the embedded energy functions. Thus, the embedding and pair potentials for Pd-H can be transformed utilizing the two equations below:

$$F_i^{Final}(\rho_i) = F_i^{initial}(\rho_i) + k\rho_i \tag{9}$$

$$\varphi_{ij}^{Final}(r_{ij}) = \varphi_{ij}^{initial}(r_{ij}) + 2k\rho_i(r_{ij}) \tag{10}$$

k represents an arbitrary constant. The embedding and pair potentials for Pd-H were thus converted in this pattern according to the method of Zhou et al. [4].

CHAPTER 3

FITTING AND VALIDATION OF PALLADIUM SILVER COPPER ALLOYS

3.1 Ag and Cu Fitting Parameters

For the pure metal Cu, the EAM potentials were fitted the same way as previously done for Pd and Ag [46, 47]. For each metal, the six fitting parameters included a_o, E_c, C_{11}, C_{12}, C_{44}, and E_{vf}. Where a_o is the lattice constant, E_c is the cohesive energy, C_{11}, C_{12}, and C_{44} are three elastic constants, and E_{vf} is the vacancy formation energy. Table 1 lists the Pd, Ag and Cu fitting parameters.

Table 1. Pd, Ag and Cu fitting parameters [46, 47]

	χ	φ_e	δ	β	η	ρ_e
Pd	**2.054085**	**0.216817**	**8.414105**	**7.221224**	**0.999999**	**3.316887**
Ag	**1.584768**	**0.154164**	**8.491335**	**7.183185**	**1.022270**	**2.213230**
Cu	**2.504500**	**0.175425**	**8.713725**	**6.906629**	**0.560027**	**3.648665**

As can be seen from Table 2, the calculated fitting results were almost identical to the experimental values [48] and those obtained by Foiles et al. [55]. In addition, as with Pd [46], the plots of cohesive energy vs. lattice constant for Ag [47] and Cu were also in very good consistency with the equation of state obtained from Rose et al. [54] as shown in Figure 1.

Table 2. Fitting results for Ag and Cu [47, 55]

Ag	a_o **(nm)**	E_c **(eV)**	C_{11} **(GPa)**	C_{12} **(GPa)**	C_{44} **(GPa)**	B **(GPa)**	E_{vf} **(eV)**
Calculations	**0.409**	**2.85**	**123.1**	**94.4**	**46.9**	**104**	**1.08**
Simulation	**0.409**	**2.85**	**123.1**	**94.4**	**46.9**	**104**	**1.08**
Foiles et al.	**0.409**	**2.85**	**124.0**	**91.0**	**57.0**	**102**	**0.97**
Experimental	**0.409**	**2.85**	**124.0**	**93.4**	**46.1**	**104**	**1.10**
Cu							
Calculations	**0.3615**	**3.54**	**168.4**	**122.8**	**76.7**	**138**	**1.30**
Simulation	**0.3615**	**3.54**	**168.4**	**122.8**	**76.7**	**138**	**1.30**
Foiles et al.	**0.3615**	**3.54**	**167.0**	**124.0**	**76.0**	**138**	**1.28**
Experimental	**0.3615**	**3.54**	**170.0**	**122.5**	**75.8**	**138**	**1.30**

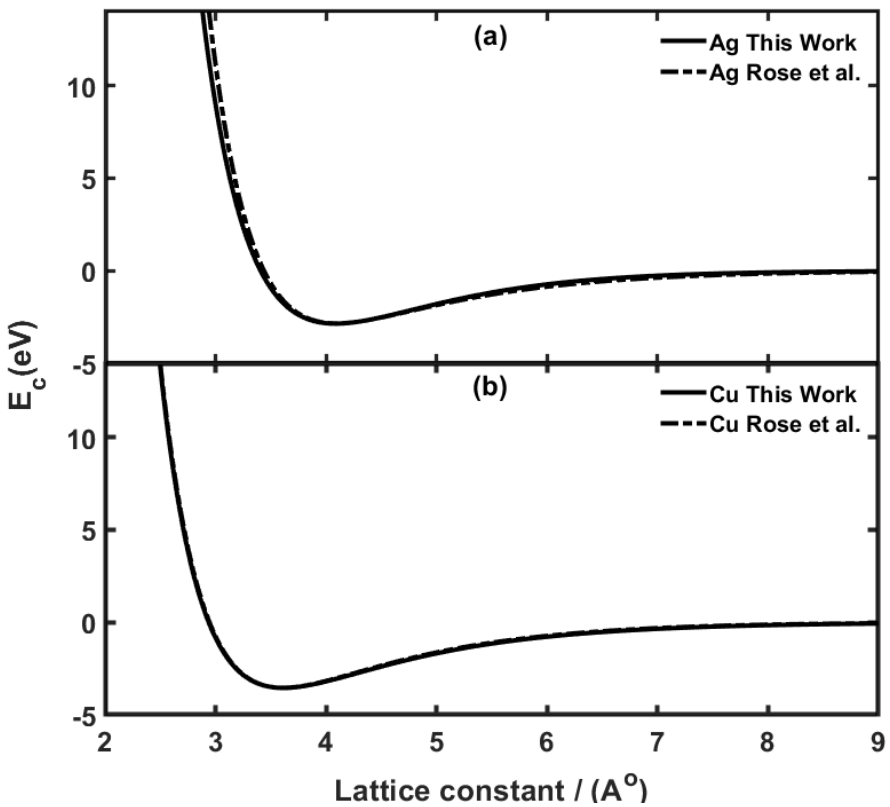

Figure 1: Ag and Cu cohesive energy and Rose et al. equation of state [54].

To account for the pair potential interactions for Pd-Ag, Pd-Cu and Ag-Cu alloys, the mixing rule between a type-a and a type-b atom interaction introduced by Johnson [56] was applied, and is given by the equation:

$$\phi_{ab}(r) = \frac{1}{2}\left[\frac{f_b(r)}{f_a(r)}\phi_{aa}(r) + \frac{f_a(r)}{f_b(r)}\phi_{bb}(r)\right] \tag{11}$$

For each type in the alloy, the electron density parameter can be calculated from the equation $f_e = S(E_c/\Omega)$, where Ω is the atomic volume and S is a scaling factor, with S = 1 for pure metals. For type-a atom as a host (solvent) and type-b as impurity (solute), the unrelaxed dilute limit heat of solution can be determined by the five steps given below:

(a) Remove host: $F^b(\rho^{-a}) + \sum_{i\neq 1}\phi^{ab}\left(r_{li}^a\right)$

(b) Add impurity: $-F^a(\rho^{-a}) - \sum_{i\neq 1}\phi^{aa}\left(r_{li}^a\right)$

(c) Adjust neighbors:

$$-\sum_{i\neq 1} F^{a}(\rho^{-a}) - \sum_{i\neq 1} F_{i}^{a}(\rho^{-a} - f^{a}\left(r_{li}^{a}\right) + f^{b}\left(r_{li}^{a}\right))$$

(d) Adjust cohesive energy: (e) Relaxation energy:

$$-E_c^a + E_c^b \qquad E_r = \left[1.167(\Omega_b / \Omega_a - 1)\right]^2$$

where ρ^a is the expression of electron density for type-a atom, r^a is the distance to its closest neighbor and E_r is the drop in total energy caused by relaxation and is predominantly dependent on the unit cell volume mismatch.

The electron density scaling factors for type-a and type-b atoms, S_a and S_b for the Pd-Ag, Pd-Cu and Ag-Cu pair potentials, obtained from fitting the experimental heat of solutions, are listed in Table 3 along with the calculated heat of solution values for each metal. The values for the relaxed heat of solution are very consistent with experimental obtained data and overall better than those obtained by Foiles et al. [55] and Hijazi and Park. [49].

Table 3: Pd-Ag, Pd-Cu and Ag-Cu heat of solution and scaling factors from fitting

$\Delta H_{solution}$ (eV)	This work (relaxed)	Foiles et al. (relaxed)	Hijazi & Park (relaxed)	Exp. (relaxed)	Scaling Factor (S)
Pd in Ag	**-0.186**	**-0.36**	**-0.23**	**-0.11**	**1.1063**
Ag in Pd	**-0.207**	**-0.24**	**-0.17**	**-0.29**	**1.8319**
Pd in Cu	**-0.364**	**-0.33**	**-0.36**	**-0.39**	**1.7097**
Cu in Pd	**-0.461**	**-0.34**	**-0.46**	**-0.44**	**1.3419**
Ag in Cu	**0.257**	**0.18**	**0.19**	**0.25**	**1.4626**
Cu in Ag	**0.389**	**0.11**	**0.41**	**0.39**	**1.6275**

3.2 Validation

The fitted parameters for Pd-Pd, Ag-Ag, Cu-Cu, Pd-Ag, Pd-Cu and Ag-Cu have been applied to create a tabulated EAM potential file in DYNAMO *setfl* format for the ternary Pd-Ag-

Cu system. Utilizing the tabulated EAM potential file, MD simulations for Ag-Ag, Cu-Cu, Pd-Ag, Pd-Cu and Ag-Cu structures were performed using a LAMMPS script code [57]. The MD simulation results for Ag-Ag, Cu-Cu were in excellent agreement with the calculated fitting results, as can be seen from Table 2, and the MD results proved the reliability of the Pd-Ag, Pd-Cu and Ag-Cu EAM potentials as can be seen from Figures 2-4.

Figure 2(a) and (b) show our lattice constant and cohesive energy results for the Pd_xAg_{1-x} and Pd_xCu_{1-x} ($0 \leq x \leq 1$) structures are almost identical with the experimental data [58]. For the Pd_xAg_{1-x} structures, the lattice constant results are closer to the experimental values than the results calculated using the Hale et al.'s EAM potentials with Morse pair function [5]. On the other hand, the density functional theory (DFT) data collected from Løvvik and Olsen [59] reveal a similar trend but overestimate the values for all compositions. Figure 2(b) shows that our cohesive energy values for Pd-Ag are very much in line with the values that are derived using the Hale et al. with Morse function [5]. The Hale et al. EAM potentials were obtained from the Interatomic Potential Repository [60]. However, their EAM potential with the hybrid model produced erratic results and was not included with the figures.

For the Pd_xCu_{1-x} structures, the lattice constant values from MD simulations are almost identical with the experimental values and those obtained by Kart and Erbay [31] as shown in Figure 2(a). For the cohesive energies for Pd_xCu_{1-x}, our MD values have an increasing trend similar to the data obtained by Kart and Erbay [31], as can be seen in Figure 2(b).

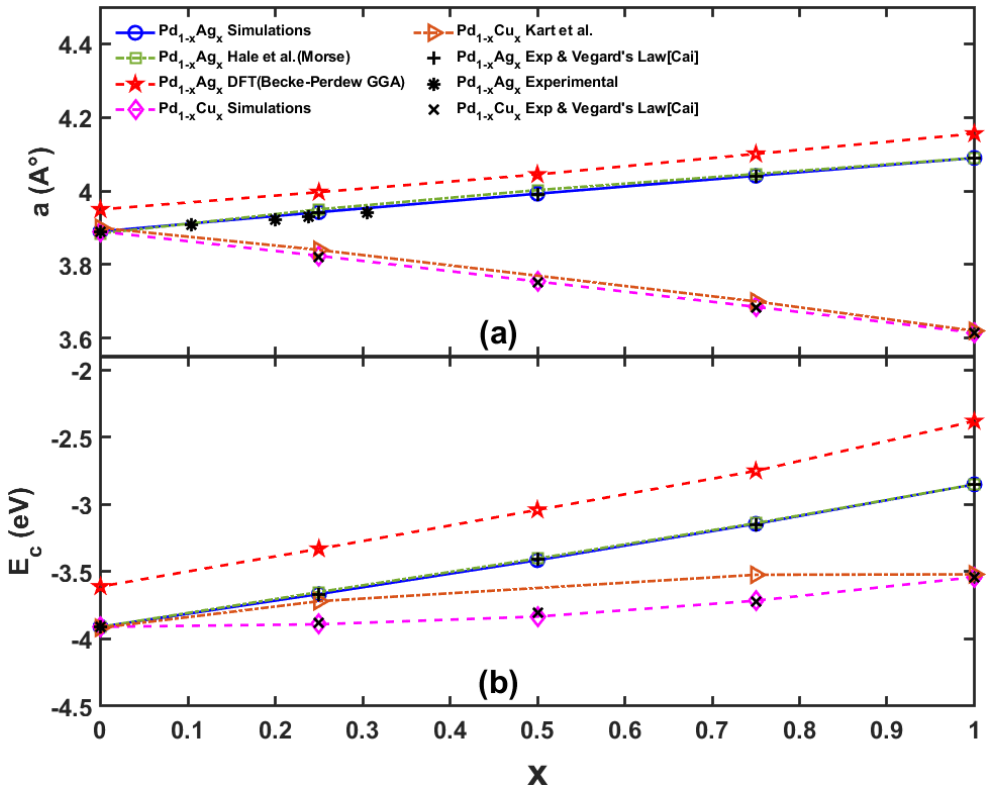

Figure 2: $Pd_{1-x}Ag_x$ and Pd_xCu_{1-x} alloys lattice constants and cohesive energies from MD, experiments, and DFT [31, 62, 59, 58].

In Figure 3(a) and (b) the values for the elastic constants C_{11}, C_{12} from MD simulations for $Pd_{1-x}Ag_x$ and $Pd_{1-x}Cu_x$ structures show consistent trend with the DFT calculations [61] and the results from Hale et al. [5] and Kart and Erbay [31]. The bulk modulus for $Pd_{1-x}Ag_x$ and $Pd_{1-x}Cu_x$ structures obtained from MD simulations match the softening trends predicted by the DFT calculations as well [31,59,61] and match the given experimental data at the edge of the composition range quite closely, as shown in Figure 3(c). It is worth noticing that the Hale et al. [5] EAM potential overestimates C_{11}, C_{12} and bulk modulus for the pure Pd metal, as can be seen at the left edges of Figure 3(a), (b) and (c).

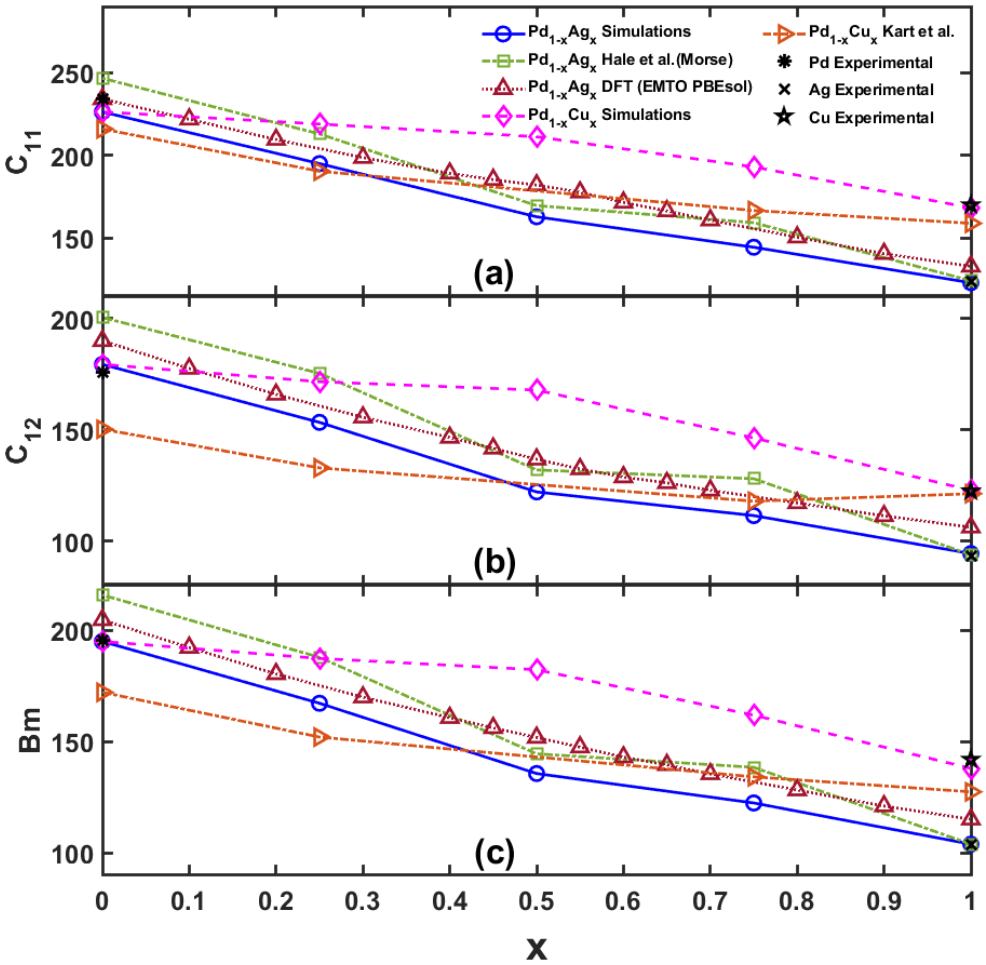

Figure 3: $Pd_{1-x}Ag_x$ and Pd_xCu_{1-x} alloys C_{11}, C_{12} elastic constant and bulk modulus from MD, experiment, and DFT [31, 63].

In Figure 4, the values for the elastic constants C_{44} and C' from MD simulations for $Pd_{1-x}Ag_x$ show that our results are closer to the experimental data at the edges of the composition range than those of Hale et al. [5]. As with the Hale et al. EAM Morse model [5], our $Pd_{1-x}Ag_x$ potential underestimates C_{44} relative to the DFT results and have an overall decreasing trend. For $Pd_{1-x}Cu_x$, our C_{44} and C' values have a slightly increasing trend, with C' for Pd being underestimated, but still more consistent with the experimental data than the results from Kart and Erbay [31].

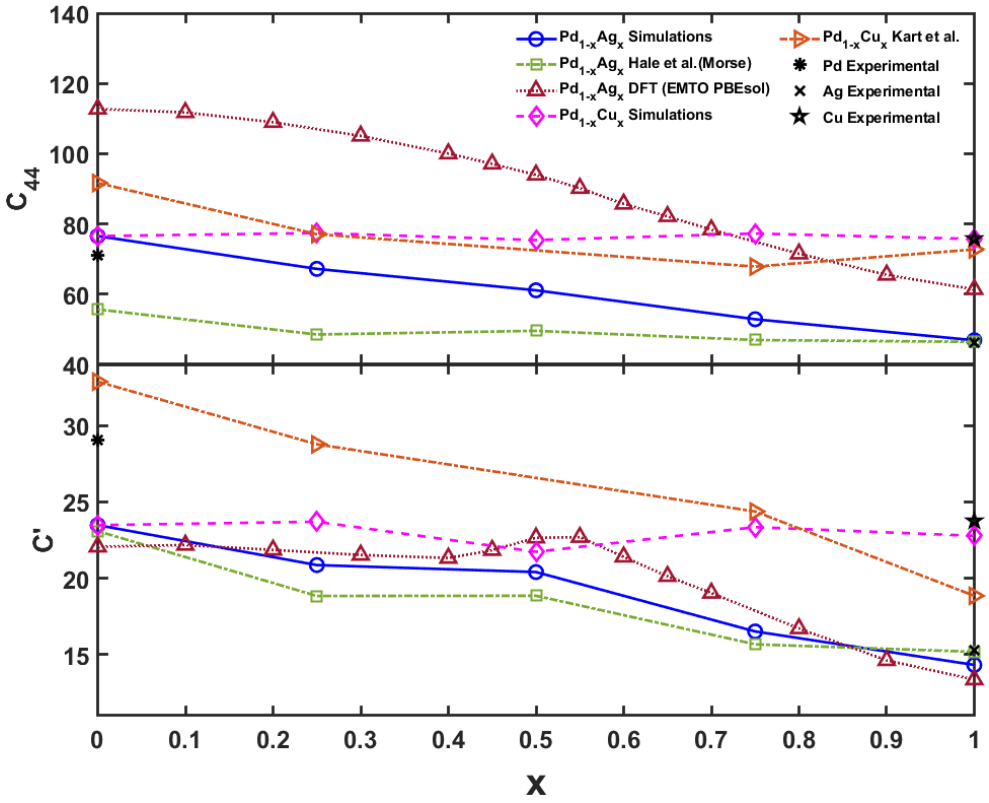

Figure 4: $Pd_{1-x}Ag_x$ and Pd_xCu_{1-x} alloys C_{44} and C' elastic constant from MD, experiment, and DFT [31, 63].

CHAPTER 4

FITTING AND VALIDATION OF PALLADIUM SILVER AND PALLADIUM COPPER HYDRIDES

4.1 DFT Calculations

Since H is almost insoluble in Ag [25, 64], and no experimental fitting data were found for Ag-H and Cu-H systems [46], therefore, the Pd-Ag-H and Pd-Cu-H properties were used as fitting data to fit the φ_{Ag-H} and φ_{Cu-H} pair functions. However, only a limited experimental and ab initio data were available to utilize a full H concentration in the fitting procedure. Hale et al. obtained their fitting data by utilizing DFT calculations for the Pd-Ag-H system [5], but failed to provide the lattice constant values, and only the cohesive energy values were given. Wei et al. performed DFT calculations on Pd-Cu-H phase stability, heat of formations and elastic property based on generalized gradient approximations (GGA) for the range of hydrogen concentration $0 \leq x \leq 0.5$, but they failed to report the exact values for the lattice constants and the cohesive energies [65]. In this thesis, the open source SIESTA software was used to perform ab initio simulations to get full fitting data for the Pd-Ag-H and Pd-Cu-H structures. The SIESTA pseudopotentials were obtained from the Abinit's Fritz-Haber-Institute (FHI) pseudo database [66]. The local density approximation (LDA) method with Ceperley–Alder exchange and correlation form using the norm-conserving Troullier–Martins scheme was utilized in the pseudopotentials. Valence states were described using double zeta-polarized (DZP) basis sets with split-valence scheme for multiple-zeta. The ab initio simulations were conducted with a dense 18×18×18 Monkhorst–Pack grid, a cutoff energy of 100 Ry, a 25 K electronic temperature, and electron spin polarization during the DFT calculations. For our Pd-Ag-H and

Pd-Cu-H structures, the calculations utilized periodic boundary conditions with a unit cell with 3 Pd atoms, 1 Ag or Cu atom, and 1 to 4 H atoms at different locations.

During the DFT simulations, the $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ structures were constructed with five different H concentrations: x = 0, 0.25, 0.50, 0.75, and 1.00. In the Pd-Ag and Pd-Cu fcc lattice, H atoms were located in three different interstitial positions. As shown in Figure 5, these positions included the octahedral (O) positions, in which O1 represent a body center position and O2 an edge center position, and the tetrahedral (TE) positions.

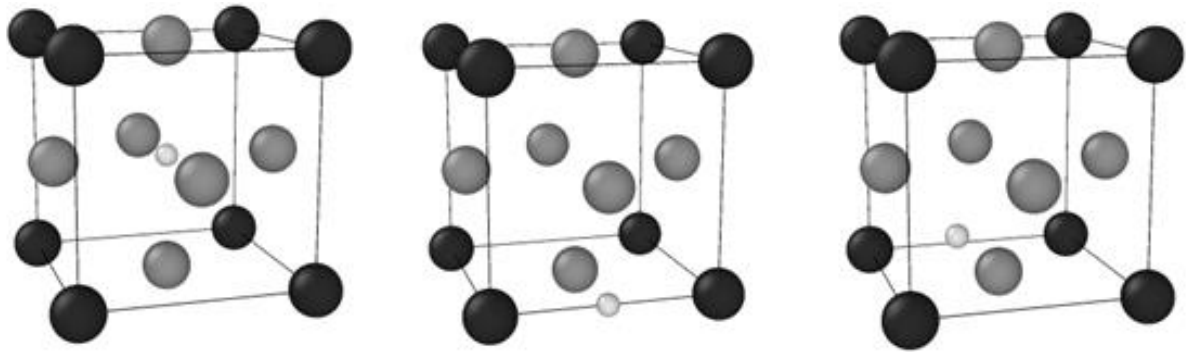

Figure 5: H (light grey) interstitial sites within the Pd (grey) and Ag or Cu (black) lattice.

As with Løvvik and Olsen $Pd_{1-x}Ag_x$ DFT results from Figure 2(a) and (b) [59], the lattice constant and cohesive energy results from our DFT simulations for the Pd-Ag-H and Pd-Cu-H structures were also overestimated in comparison to the available experimentally obtained data for the Pd, Ag, Cu, $PdH_{0.50}$, $PdH_{1.00}$, $Pd_{0.75}Ag_{0.25}$ and $Pd_{0.75}Cu_{0.25}$ structures. The calculated DFT values can be shifted, if multiplied with a selected factor, to make it consistent with the experimental data [67]. Equations 12 to 15 describe the shifting procedure for the cohesive energies for $Pd_{0.75}Ag_{0.25}H_x$ structures, which have been applied in a similar manner to the $Pd_{0.75}Cu_{0.25}H_x$ case by replacing Ag atoms with Cu. The shifting procedure was also applied in a similar fashion to the lattice constants case. The shifting data for $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ structures are given in Table 4 and the shifting factors in Table 5.

$$\left(\text{Cohesive Energy}_{Pd_{0.75}Ag_{0.25}H_x}\right)_{SIESTA\ shifted} = \left(C_{Pd_{0.75}Ag_{0.25}} + x \cdot C_H\right) \cdot \left(\text{Cohesive Energy}_{Pd_{0.75}Ag_{0.25}H_x}\right)_{SIESTA} \quad (12)$$

$$C_{Pd_{0.75}Ag_{0.25}} = \frac{\left(\text{Cohesive Energy}_{Pd_{0.75}Ag_{0.25}}\right)_{Experimental}}{\left(\text{Cohesive Energy}_{Pd_{0.75}Ag_{0.25}}\right)_{SIESTA}} \quad (13)$$

$$C_H = 2 \cdot \left\{\frac{\left(\text{Cohesive Energy}_{PdH_{0.50}}\right)_{Experimental}}{\left(\text{Cohesive Energy}_{PdH_{0.50}}\right)_{SIESTA}} - C_{Pd}\right\} \quad (14)$$

$$C_{Pd} = \frac{\left(\text{Cohesive Energy}_{Pd}\right)_{Experimental}}{\left(\text{Cohesive Energy}_{Pd}\right)_{SIESTA}} \quad (15)$$

Table 4. Experimental values used in shifting ab initio data

Structure	Lattice Constant (A°)	Cohesive Energy (eV)
Pd	**3.89**	**-3.91**
$PdH_{0.50}$	**-**	**-3.4877**
$PdH_{1.00}$	**4.12**	**-**
$Pd_{0.75}Ag_{0.25}$	**3.94**	**-3.65**
$Pd_{0.75}Cu_{0.25}$	**3.821**	**-3.8775**

Table 5. Shifting factors for Pd-Ag-H and Pd-Cu-H ab initio data

Lattice Constant		Cohesive Energy	
Structure	**Factor**	**Structure**	**Factor**
$C_{Pd_{0.75}Ag_{0.25}}$	**0.98546**	$C_{Pd_{0.75}Ag_{0.25}}$	**0.78677**
$C_{Pd_{0.75}Cu_{0.25}}$	**0.98255**	$C_{Pd_{0.75}Cu_{0.25}}$	**0.79091**
C_H	**0.017677**	C_H	**0.021070**
C_{Pd}	**0.98232**	C_{Pd}	**0.79165**

Figure 6 shows the shifted cohesive energy values for the $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ structures with various OC and TE hydrogen interstitial positions. As can be seen from the figure, the shifted cohesive energy values for the $Pd_{0.75}Ag_{0.25}H_x$ structures obtained by Fuller and Hijazi [47] are in close agreement to those obtained by Hale et al. DFT calculations [5] and have similar trend. The $Pd_{0.75}Cu_{0.25}H_x$ shifted data obtained in this work also have a

similar trend to the $Pd_{0.75}Ag_{0.25}H_x$ shifted data, but with lower cohesive energy values as can be seen from the figure. The lower cohesive energies for the $Pd_{0.75}Cu_{0.25}H_x$ structures are expected since Cu has lower cohesive energies than Ag.

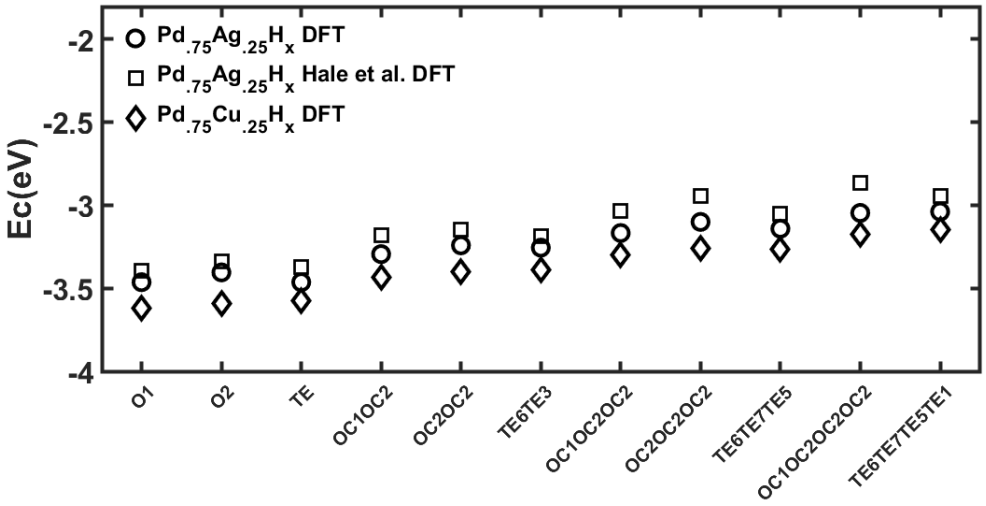

Figure 6: $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ shifted DFT results comparison with Hale et al.

Table 6. Pd-Ag-H and Pd-Cu-H ab initio data, fitting results, and MD results

		SIESTA results		Shifted SIESTA		Fitting Calc	MD Results
Composition	**Structure**	**a (A)**	**E_c (eV)**	**a (A)**	**E_c (eV)**	**E_c (eV)**	**E_c (eV)**
Pd	**FCC**	**3.9600**	**-4.9339**	**3.89**	**-3.9100**	**-3.9100**	**-3.9100**
Ag	**FCC**	**4.1443**	**-3.5480**	**4.09**	**-2.8500**	**-2.8500**	**-2.8500**
Cu	**FCC**	**3.6284**	**-4.5656**	**3.615**	**-3.5400**	**-3.5400**	**-3.5400**
$Pd_{0.75}Ag_{0.25}$	**FCC**	**3.9981**	**-4.6392**	**3.9400**	**-3.6500**	**-3.6660**	**-3.6660**
$Pd_{0.75}Cu_{0.25}$	**FCC**	**3.8889**	**-4.9026**	**3.8210**	**-3.8775**	**-3.8905**	**-3.8905**
$Pd_{0.75}Ag_{0.25}H_{0.25}$	**OC1**	**4.0451**	**-4.3728**	**4.0042**	**-3.4634**	**-3.4415**	**-3.4450**
	OC2	**4.0465**	**-4.2984**	**4.0056**	**-3.4045**	**-3.4359**	
	TE	**4.0427**	**-4.3727**	**4.0019**	**-3.4634**	**-3.3931**	
$Pd_{0.75}Cu_{0.25}H_{0.25}$	**OC1**	**3.9499**	**-4.5465**	**3.8880**	**-3.6198**	**-3.6220**	**-3.6298**
	OC2	**3.9402**	**-4.5106**	**3.8785**	**-3.5912**	**-3.6119**	
	TE	**3.9684**	**-4.4900**	**3.9062**	**-3.5748**	**-3.5590**	
$Pd_{0.75}Ag_{0.25}H_{0.50}$	**OC1 OC2**	**4.0897**	**-4.1322**	**4.0663**	**-3.2946**	**-3.318**	**-3.3116**
	OC2 OC2	**4.1034**	**-4.0657**	**4.0800**	**-3.2416**	**-3.2423**	
	TE TE	**4.1475**	**-4.0833**	**4.1239**	**-3.2556**	**-3.2581**	
$Pd_{0.75}Cu_{0.25}H_{0.50}$	**OC1 OC2**	**3.9985**	**-4.2861**	**3.9429**	**-3.4351**	**-3.4742**	**-3.4841**
	OC2 OC2	**3.9902**	**-4.2422**	**3.9347**	**-3.3999**	**-3.4090**	
	TE TE	**4.0419**	**-4.2297**	**3.9858**	**-3.3899**	**-3.3978**	
$Pd_{0.75}Ag_{0.25}H_{0.75}$	**OC1 OC2 OC2**	**4.1330**	**-3.9469**	**4.1277**	**-3.1677**	**-3.2270**	
	OC2 OC2 OC2	**4.1368**	**-3.8636**	**4.1315**	**-3.1008**	**-3.2160**	**-3.2332**
	TE TE TE	**4.2188**	**-3.9143**	**4.2134**	**-3.1126 to -3.1415**	**-3.0974**	
$Pd_{0.75}Cu_{0.25}H_{0.75}$	**OC1 OC2 OC2**	**4.0360**	**-4.0873**	**3.9871**	**-3.2973**	**-3.3780**	
	OC2 OC2 OC2	**4.0249**	**-4.0433**	**3.9761**	**-3.2617**	**-3.3740**	**-3.4112**
	TE TE TE	**4.1100**	**-4.0466**	**4.0602**	**-3.2315 to -3.2644**	**-3.1549**	
$Pd_{0.75}Ag_{0.25}H_{1.00}$	**OC1 OC2 OC2 OC2**	**4.1717**	**-3.7712**	**4.1848**	**-3.0465**	**-3.2484**	**-3.2417**
	TE TE TE TE	**4.2876**	**-3.7616**	**4.3011**	**-3.0009 to -3.0388**	**-2.8378**	
$Pd_{0.75}Cu_{0.25}H_{1.00}$	**OC1 OC2 OC2 OC2**	**4.0787**	**-3.9131**	**4.0366**	**-3.1773**	**-3.4000**	**-3.3973**
	TE TE TE TE	**4.1832**	**-3.8764**	**4.1399**	**-3.1101 to -3.1475**	**-3.1280**	

4.2 Ag-H and Cu-H Fitting

The shifted values for the cohesive energy and lattice constant obtained from the DFT simulations are listed in Table 6. The shifted cohesive energies for 7 OC structures and 7 TE structures with 4 different H concentrations, were used in fitting the φ_{Ag-H} and φ_{Cu-H} pair potential functions during the fitting process. The φ_{Ag-H} and φ_{Cu-H} pair functions take the same generalized Morse potential mathematical form, as used previously in the Pd-H interaction [52]. Since a third atom type was added to the binary Pd-H structures to create ternary Pd-Ag-H and Pd-Cu-H structures, all Pd-H potentials and property equations utilized in the fitting procedures were expanded by adding a central atom expression as a third type [46]. For the ternary system, the cohesive energy equation has an additional host term and is given by:

$$E_c = \frac{1}{x+y+z}\left[\begin{array}{l} x\left(F_{a,i}(\rho_{a,i}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_a}\varphi_{a-a,ij}(r_{ij}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_b}\varphi_{a-b,ij}(r_{ij}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_c}\varphi_{a-c,ij}(r_{ij})\right) + \\ y\left(F_{b,i}(\rho_{b,i}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_b}\varphi_{b-b,ij}(r_{ij}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_a}\varphi_{b-a,ij}(r_{ij}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_c}\varphi_{b-c,ij}(r_{ij})\right) + \\ z\left(F_{c,i}(\rho_{c,i}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_c}\varphi_{c-c,ij}(r_{ij}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_a}\varphi_{c-a,ij}(r_{ij}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_b}\varphi_{c-b,ij}(r_{ij})\right) \end{array}\right] \quad (16)$$

$$\rho_{a,i} = \rho_{a-a,i} + \rho_{a-b,i} + \rho_{a-c,i} = \sum_{\substack{j=1\\ j\neq i}}^{N_a} f_a(r_{ij}) + \sum_{\substack{j=1\\ j\neq i}}^{N_b} f_b(r_{ij}) + \sum_{\substack{j=1\\ j\neq i}}^{N_c} f_c(r_{ij}) \quad (17)$$

$$\rho_{b,i} = \rho_{b-a,i} + \rho_{b-b,i} + \rho_{b-c,i} = \sum_{\substack{j=1\\ j\neq i}}^{N_a} f_a(r_{ij}) + \sum_{\substack{j=1\\ j\neq i}}^{N_b} f_b(r_{ij}) + \sum_{\substack{j=1\\ j\neq i}}^{N_c} f_c(r_{ij}) \quad (18)$$

$$\rho_{c,i} = \rho_{c-a,i} + \rho_{c-b,i} + \rho_{c-c,i} = \sum_{\substack{j=1\\ j\neq i}}^{N_a} f_a(r_{ij}) + \sum_{\substack{j=1\\ j\neq i}}^{N_b} f_b(r_{ij}) + \sum_{\substack{j=1\\ j\neq i}}^{N_c} f_c(r_{ij}) \quad (19)$$

where *a*, *b*, and *c* are three different type of atoms, and *x*, *y*, and *z* are the concentrations for each type of atom in the structure respectively. A constrained nonlinear optimization MATLAB code was used during the fitting procedure to obtain the fitting parameters. Table 7 lists the parameters for the Cu-H from fitting, and the Pd-H parameters from Hijazi et al. [46], the Ag-H from Fuller and Hijazi [47].

Table 7. Fitting parameters for Pd-H, Cu-H and Ag-H [46, 47]

System	D	α	β	r_0
Pd-H	**0.740938**	**2.373944**	**1.702142**	**1.300000**
Ag-H	**1.476745**	**1.967649**	**1.741865**	**1.850017**
Cu-H	**0.799995**	**1.730082**	**1.400001**	**1.500002**

The previously obtained fitting data for the H-H potential are also included in Table 8 [46]. The two body potential functions used in our Pd-Ag-Cu-H model are plotted in Figure 7 and Figure 8. The calculated cohesive energies for $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ structures from fitting are consistent with the fitting data for most of the H concentrations, but the results start to diverge from the fitting data at high H concentrations as can be seen in Figure 9(b).

Table 8. Fitting parameters for H-H

D_{HH}	α_{HH}	β_{HH}	$r_{o,HH}$	C_H
0.589510	**1.104827**	**0.942490**	**3.474173**	**2.145808**
δ_H	a_H	b_H	c_H	d_H
0.942201	**8.370790**	**62.343273**	**0.000100**	**1.187000**

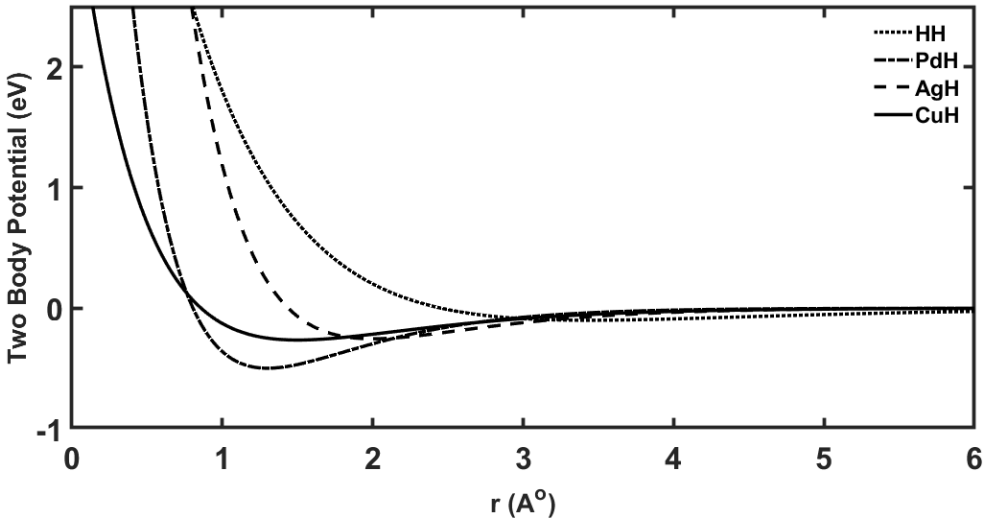

Figure 7: Pair potential functions of the fitted H-H, Pd-H, Ag-H and Cu-H potentials.

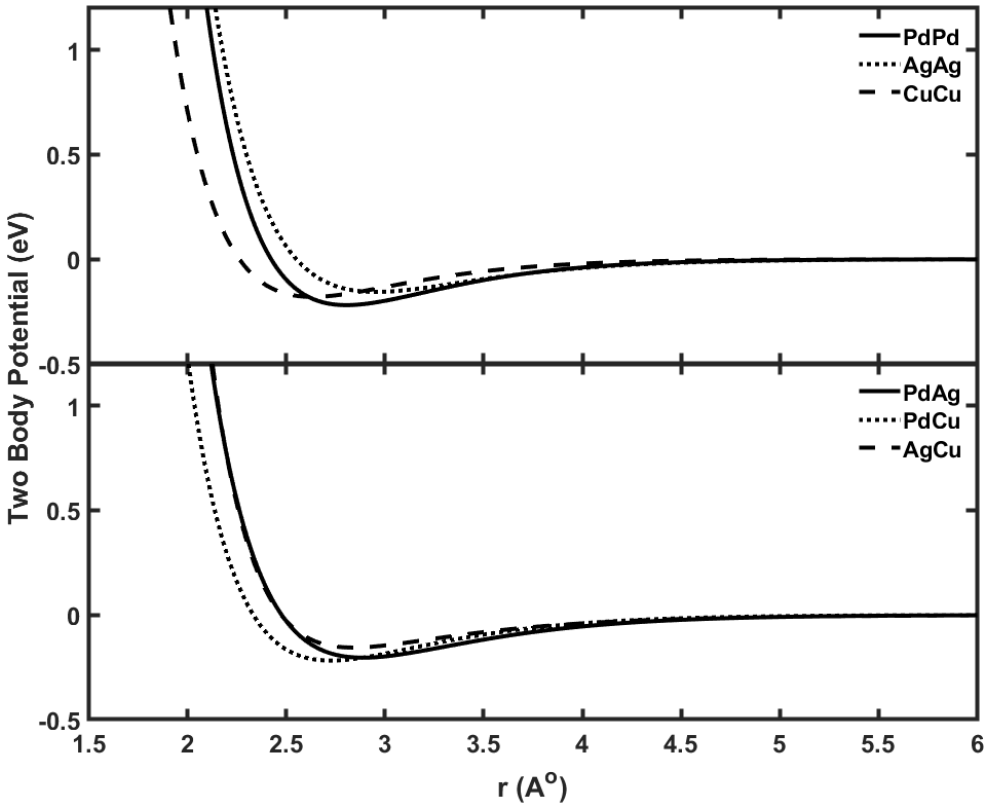

Figure 8: Pair potential functions of the fitted Pd-Pd, Ag-Ag, Cu-Cu, Pd-Ag, Pd-Cu and Ag-Cu potentials.

4.3 Validation

To test the reliability of the Pd-Ag-Cu-H potentials, a tabulated potential file in DYNAMO *setfl* format was generated utilizing the final fitting parameters. Utilizing LAMMPS

and the tabulated potential file, MD annealing simulations with a Nose-Hoover NPT ensemble from 500 K to 1 K in 100 ns with random hydrogen atom positions were performed for the $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ structures. A molecular statics (MS) simulation utilizing the conjugate gradient (cg) minimization method was then applied after each MD simulation. Ten sets of data were generated for each H composition and their average values were taken to ensure accuracy.

4.3.1 Lattice Constants and Cohesive Energy

The stress triggered by variation in the lattice constants in regions with different H concentrations at equilibrium is of important consideration [4]. Therefore, the influence of H concentration on the equilibrium lattice constant was investigated. As can be seen in Figure 9(a), the lattice constant values obtained from MD simulations for $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ structures are almost identical with the DFT results used in the fitting process. The lattice constant plots show an increasing trend with increasing H concentrations, similar with the DFT calculated data and the $Pd_{1.00}H_x$ results from Hijazi et al. [46]. The increasing trend from our plots is also consistent with the results calculated using the Hale et al. EAM potential with the Morse function for $Pd_{0.75}Ag_{0.25}H_x$ [5], and the DFT simulation results for the fcc Pd-Cu-H (O1) from Wei et al. [65].

The cohesive energies for $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$, plotted in Figure 9 (b), were in excellent agreement with the results from the fitting calculations, and in good agreement with our DFT data and follows a similar trend to the plot from our $Pd_{1.00}H_x$ simulations [46]. The simulation results for $Pd_{0.75}Ag_{0.25}H_x$ are in closer agreement to our DFT data than those obtained with the Hale et al. EAM potential with respect to their own DFT fitting data [5].

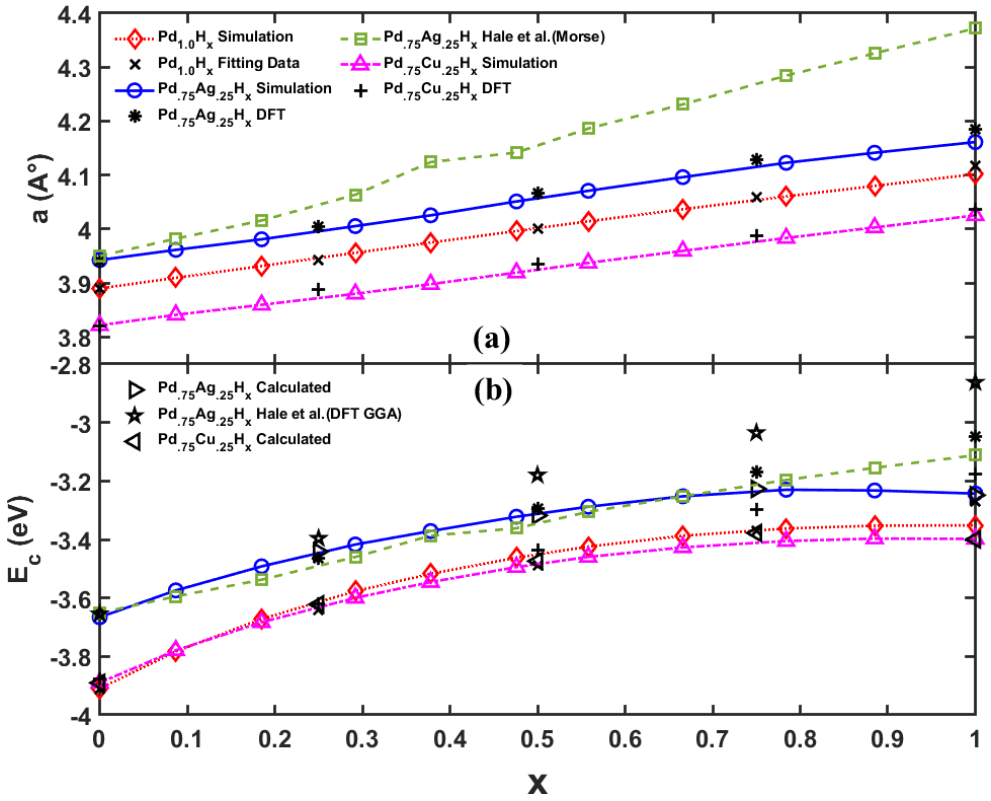

Figure 9: $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ lattice constants and cohesive energies from MD and fitting data.

4.3.2 Bulk Modulus and Elastic Constants

Using the relaxed $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ structures obtained from our MD + MS simulations utilizing a LAMMPS script file, the elastic constants and bulk modulus values were estimated [57]. Figure 10 shows that the elastic constants C_{11} and C_{12} and bulk modulus for $Pd_{0.75}Ag_{0.25}H_x$, $Pd_{0.75}Cu_{0.25}H_x$ and $Pd_{1.00}H_x$ display smooth curves with similar trends. Although, as stated previously, that the Hale et al. EAM potential overestimates the bulk modulus for pure Pd, our overall decreasing trend matches well with the results obtained by Hale et al. EAM Morse potential for the $Pd_{0.75}Ag_{0.25}H_x$ structures and in a good agreement with our previously obtained results for $Pd_{1.00}H_x$ [5,46]. The bulk modulus values from our MD simulations for $Pd_{0.75}Cu_{0.25}H_x$ also have a similar decreasing trend with those obtained by Wei et al. DFT simulations for fcc Pd-Cu-H (O1) [65]. Other researchers have also documented this softening

behavior with increasing H concentrations [68-70]. The results for $Pd_{0.75}Ag_{0.25}H_x$ structures obtained from Hale et al. Hybrid potential yielded an unstable trend but still had an overall similar softening trend [5,60]. As for the $Pd_{0.75}Cu_{0.25}H_x$ compositions, our simulation results for C_{11} and C_{12} also have a similar smooth overall decreasing trend to those obtained by Wei et al. [65].

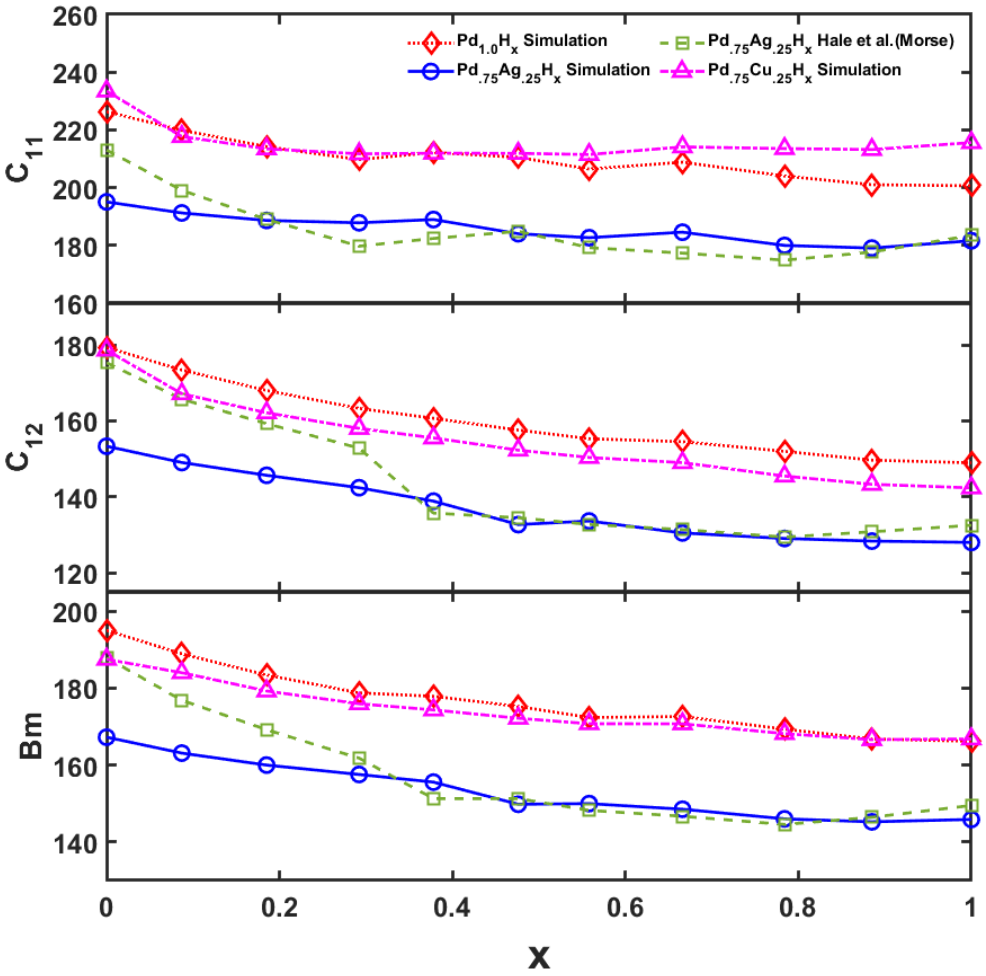

Figure 10: $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ C_{11}, C_{12} elastic constant and bulk modulus from MD.

Figure 11 shows the elastic constants C_{44} and C' for $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ alloys. As previously obtained for $Pd_{1.00}H_x$ [46], the plot values for C_{44} for $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ decrease while the shear elastic constants C' increase with increasing H composition.

In addition, our elastic constants values for the various $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ alloys shown in Figure 10 and Figure 11 satisfy the theory of strain energy for cubic structures [65]. According to strain energy theory, the following formulas can be applied to a mechanically stable cubic: $C_{11} > 0$, $C^2_{11} > C^2_{12}$ and $C_{44} > 0$. From Figure 10 and Figure 11, it can also be seen that the $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ structures have smaller C_{12} and bigger C_{44} than the $Pd_{1.00}H_x$ structures, implying that alloying Pd with Cu or Ag has a significant impact on the elastic constant properties for Pd-Cu-H and Pd-Ag-H phases [65].

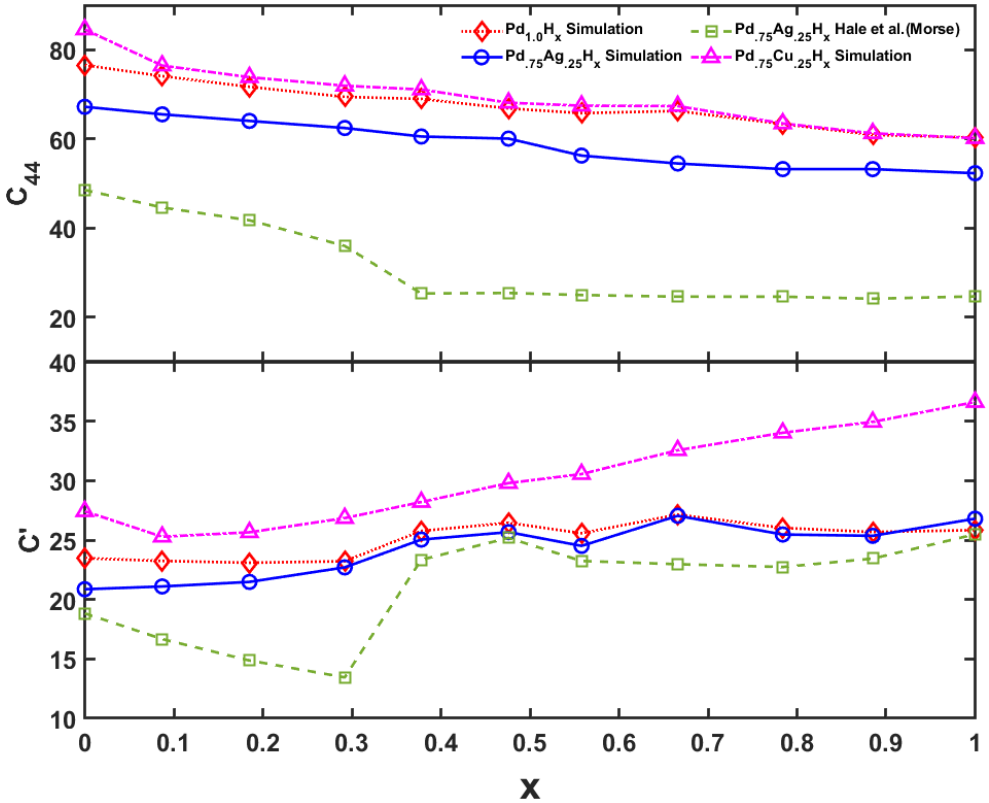

Figure 11: $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ C_{44} elastic constant and *C'* shear elastic constant from MD.

4.4 Additional Compositions

To demonstrate the validity of our EAM potentials beyond the Pd, Ag and Cu concentrations utilized during the fitting process, Figure 12 shows the lattice constants and

cohesive energies for the $Pd_{0.50}Ag_{0.50}H_x$, $Pd_{0.50}Cu_{0.50}H_x$ and $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ hydrides. The $Pd_{50}Cu_{50}$ structure was chosen on purpose because of its relative similarity to the $Pd_{52.5}Cu_{47.5}$ structure which proved to have the highest H permeability by experimental findings [65]. As can be seen from Figure 12, the lattice constant and cohesive energy results for these additional compositions display a similar trend consistent with our previously obtained results for $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ hydrides in Chapter 4, Section 4.3.

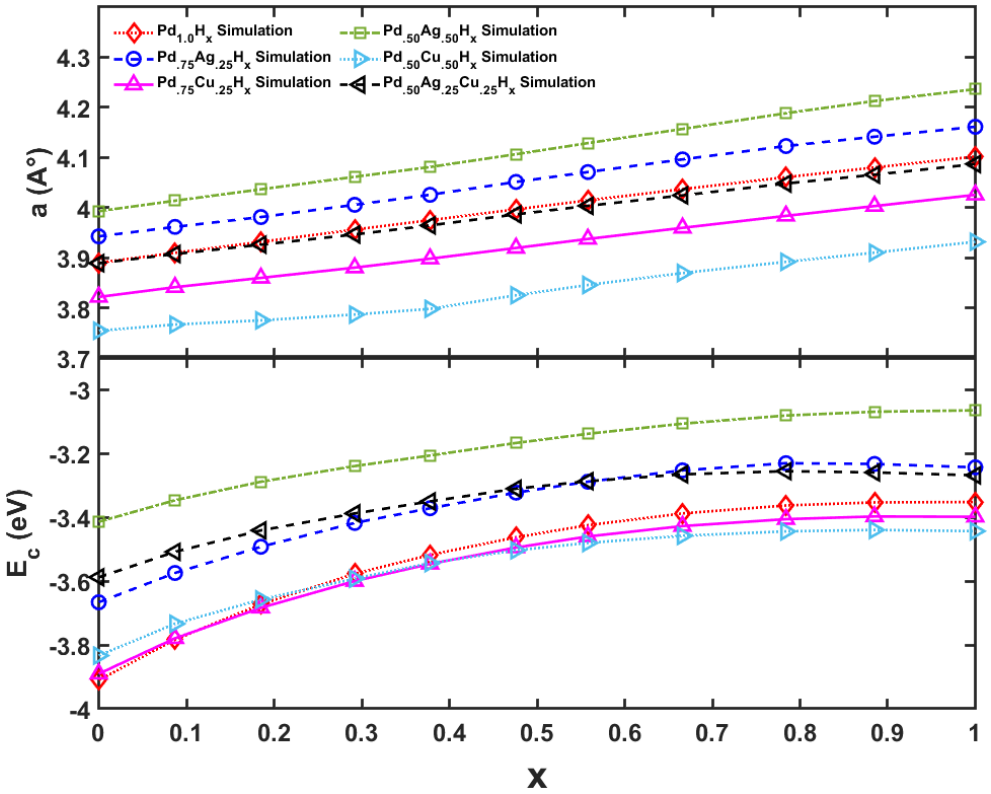

Figure 12: $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$, $Pd_{0.50}Ag_{0.50}H_x$, $Pd_{0.75}Cu_{0.25}H_x$, $Pd_{0.50}Cu_{0.50}H_x$ and $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ lattice constants and cohesive energies from MD.

In Figure 13 (a), (b) and (c) the MD simulation results for the elastic constants and bulk modulus for $Pd_{0.50}Ag_{0.50}H_x$, $Pd_{0.50}Cu_{0.50}H_x$ and $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ are plotted. It can be seen that they also have a consistent trend with our previous results for the $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ hydrides. The plots for bulk modulus for $Pd_{0.50}Ag_{0.50}H_x$, $Pd_{0.50}Cu_{0.50}H_x$ and $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ have a similar softening trend to our previous results for $Pd_{1.00}H_x$,

$Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$. For $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$, the bulk modulus values are higher than those for $Pd_{0.50}Ag_{0.50}H_x$ and closer to $Pd_{0.50}Cu_{0.50}H_x$, indicating that adding Cu has a strengthening impact on the solid solution while adding Ag has a softening effect.

The elastic constant values for C_{11} and C_{12} for the various $Pd_{0.50}Ag_{0.50}H_x$, $Pd_{0.50}Cu_{0.50}H_x$ and $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ structures shown in Figure 13 also satisfy the strain energy theory for cubic structures [65], indicating that these structures also possess mechanical stability.

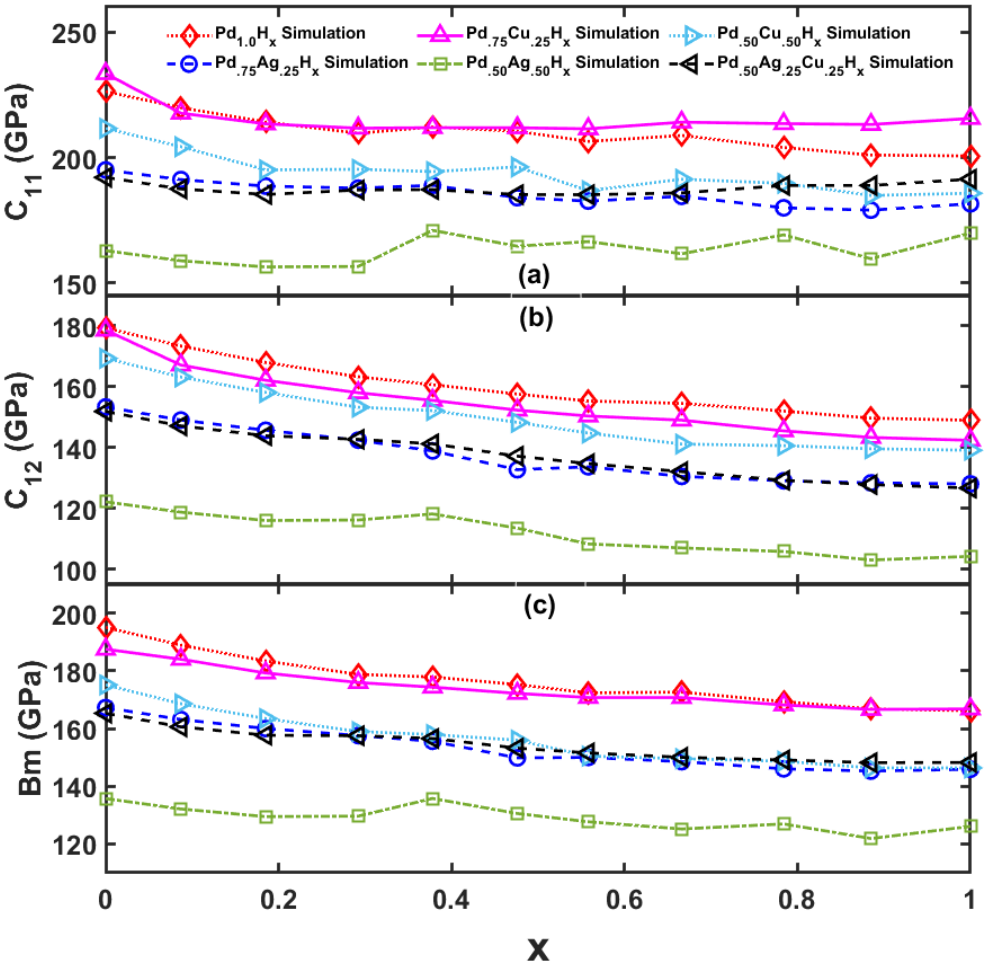

Figure 13: $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$, $Pd_{0.50}Ag_{0.50}H_x$, $Pd_{0.75}Cu_{0.25}H_x$, $Pd_{0.50}Cu_{0.50}H_x$ and $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ C_{11}, C_{12} elastic constant and bulk modulus from MD.

Figure 14 shows the elastic constants C_{44} and C' for the $Pd_{0.50}Ag_{0.50}H_x$, $Pd_{0.50}Cu_{0.50}H_x$ and $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ structures. The plot values for C_{44} decrease while the values for shear elastic

constant *C'* increases with increasing H composition. This behavior is consistent with the results obtained previously for $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ structures.

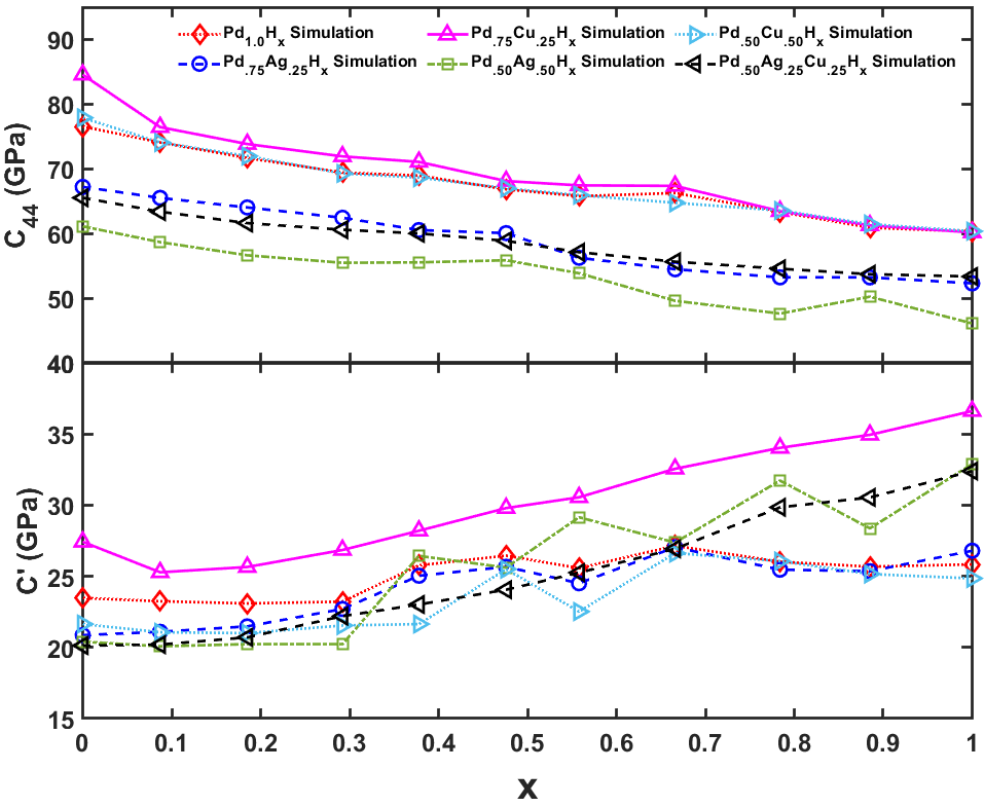

Figure 14: $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$, $Pd_{0.50}Ag_{0.50}H_x$, $Pd_{0.75}Cu_{0.25}H_x$, $Pd_{0.50}Cu_{0.50}H_x$ and $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ C_{44} and *C'* elastic constant from MD.

As a final validation of reliability of the EAM potentials, the young's modulus (E), shear modulus (G) and Poisson's ratio (v) were calculated. It could be seen from Figure 15 that at each H concentration, the descending order of E or G is as follows: $Pd_{0.75}Cu_{0.25}H_x > Pd_{1.00}H_x$. A bigger value of E or G of the Pd-Cu-H phase implies that the addition of Cu should have significant influence of solid-solution strengthening in the Pd lattice, which is in good agreement with similar statements from experiments [71] and Wei et al. [65].

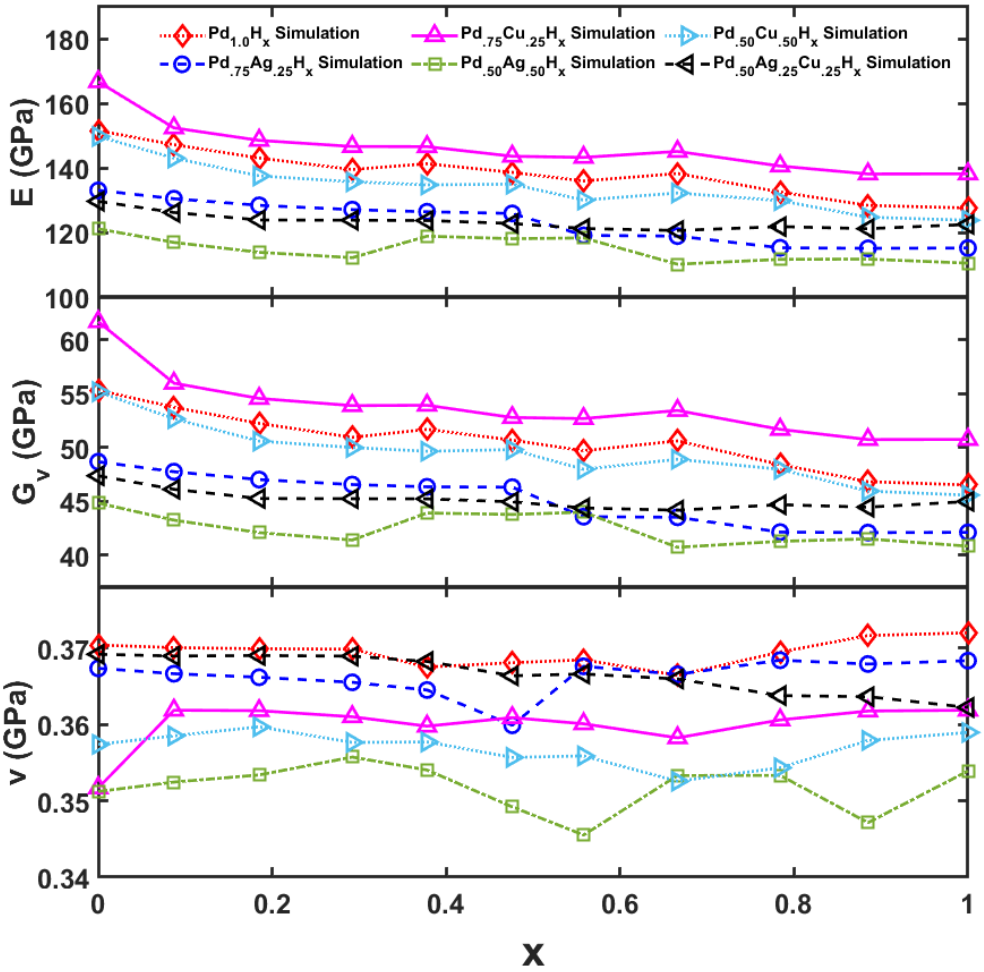

Figure 15: $Pd_{1.00}H_x$, $Pd_{0.75}Ag_{0.25}H_x$, $Pd_{0.50}Ag_{0.50}H_x$, $Pd_{0.75}Cu_{0.25}H_x$, $Pd_{0.50}Cu_{0.50}H_x$ and $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ young's modulus, shear modulus and Poisson's ratio.

4.4.1 Dynamic Stability

In the $Pd_{1.00}H_x$ hydride, H atoms tend to take the OC sites in the Pd fcc lattice [72]. The DFT calculation results show that the OC sites in $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ structures are highly energetically favorable to H atoms; this behavior was also observed and reported by other researchers [5,59]. In order to verify the stability for the $Pd_{1-y}Ag_yH_x$, $Pd_{1-y}Cu_yH_x$ and $Pd_{1-y-z}Ag_yCu_zH_x$ structures using our EAM potentials, structures with TE sites occupied by H atoms were created using LAMMPS, as shown in Figure 16(a). MD simulations were carried out with an NPT ensemble, each TE structure was annealed from 500 K to 1 K for 100 ns, and then followed by cg energy minimization. After each MD simulation, the H atoms moved to lower

energy OC sites, as was reported. The resulting structure for a $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ is shown in Figure 16(b).

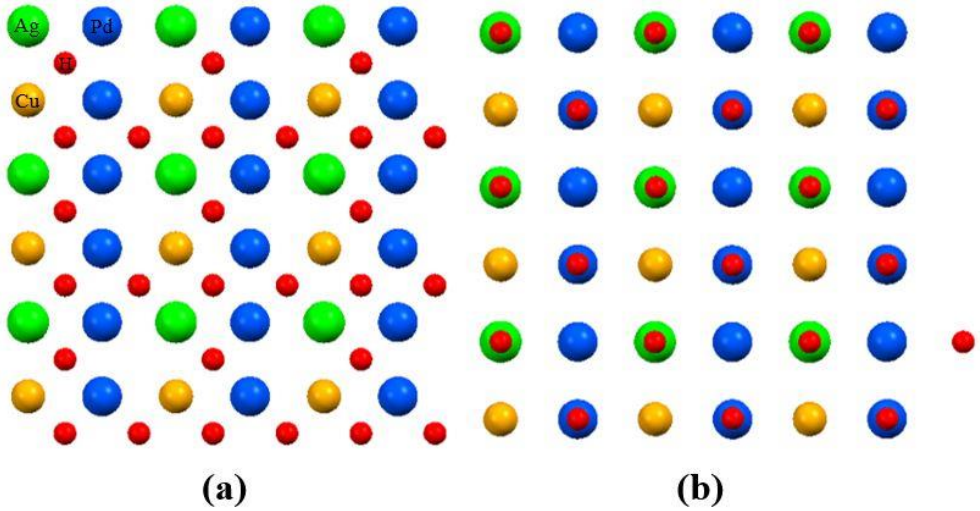

Figure 16: (a) TE structure before simulation. interstitial H (red) within the Pd (blue), Ag (green) and Cu (yellow) lattice. (b) OC structure after simulation.

4.4.2 Miscibility Gap and Gibbs Free Energy of Mixing

For the previously studied $Pd_{1.00}H_x$ hydrides, the miscibility gap was predicted based on the Gibbs free energy of mixing as a function of H concentration [46]. Following the method of Hale et al. [5], the mixing enthalpy term was modified to obtain the Gibbs free energy functions for $Pd_{1-y}Ag_yH_x$, $Pd_{1-y}Cu_yH_x$ and $Pd_{1-y-z}Ag_yCu_zH_x$ hydrides as follows:

$$\Delta G^{mix} = \Delta H^{mix} - \Delta S^{mix} \cdot T \tag{20}$$

$$\Delta H^{mix} = E_{Pd_{1-y}Ag_yH_x} - 2X \cdot E_{Pd_{1-y}Ag_yH_{1.0}} - (1-2X) \cdot E_{Pd_{1-y}Ag_y} \tag{21}$$

where the cohesive energies $E_{Pd_{1-y}Ag_yH_x}$, $E_{Pd_{1-y}Ag_yH_{1.0}}$, and $E_{Pd_{1-y}Ag_y}$ were applied, and X is the mole fraction which $X = x/(1+x)$.

$$\Delta S_t = -k_B \cdot \left[\begin{array}{l} X \cdot \ln\left[X/(1-X) \right] + \\ (1-2 \cdot X) \cdot \ln\left[(1-2X)/(1-X) \right] \end{array} \right] \tag{22}$$

where k_B is Boltzmann's constant.

At 0 K, the Gibbs free energy values in Figure 17, as expected, are all above zero for all structures and various H concentrations, indicating that the average attractive interactions between different atom types are weaker than those between the same atom types. At 300 K, Figure 18 shows that the Gibbs free energy plot for $Pd_{1.00}H_x$ has two minima at x = 0.034 and 0.95, corresponding to the mole fraction of X = 0.033 and 0.49 and represent the α and β phases, respectively. They describe a miscibility gap region in an alloy, where two phases are more stable than a single one. The Hale et al. [5] EAM Morse model predicts the α and β phases to be X = 0.0 and 0.47. Experimentally obtained phase boundaries for $Pd_{1.00}H_x$ at 300 K are x = 0.03 and 0.6, corresponding to mole fraction of X = 0.029 and 0.375. Therefore, our model is in better agreement in predicting the α phase but the β phase is slightly overestimated compared to Hale et al. EAM Morse potential. In Figure 18, our MD results for $Pd_{1-y}Ag_yH_x$ at 300 K show that when Ag concentration increases, the values become more negative relative to the $Pd_{1.00}H_x$ system, indicating more favorable mixing, and the miscibility gap become narrower. No miscibility gap observed when y = 0.5. At 300 K, the experimental values indicate that the α phase and β phase cease to be distinct at y = 0.25–0.30 for $Pd_{1-y}Ag_y$ [63]. This shows that our EAM potentials are able to detect the miscibility gap, and are consistent with the experimental results regarding the miscibility gap overall behavior as Ag concentration increases. For the $Pd_{1-y}Cu_yH_x$ compositions, experimental data at 303 K indicated that by increasing Cu concentration, the α phase and β phase shift to the right and to the left respectively, the miscibility gap narrows, and finally disappears at y = 0.29 [73, 74]. Our values from MD simulations at 300 K in Figure 18 also indicate that adding Cu causes the Gibbs free energy to increase for all H concentrations in comparison to the $Pd_{1.00}H_x$ structures. At y = 0.25, all calculated energies are positive indicating

unfavorable mixing, no miscibility gap observed at $y = 0.5$. This shows that our model predicts the miscibility gap to disappear at high Cu concentration. For $Pd_{0.50}Ag_{0.25}Cu_{0.25}H_x$ compositions, the Gibbs free energy plot has a similar trend with those obtained from the $Pd_{1.00}H_x$ structures, indicating that the addition of copper and silver with equal concentration seems to have an opposite effect on the Gibbs free energy and tend to offset each other.

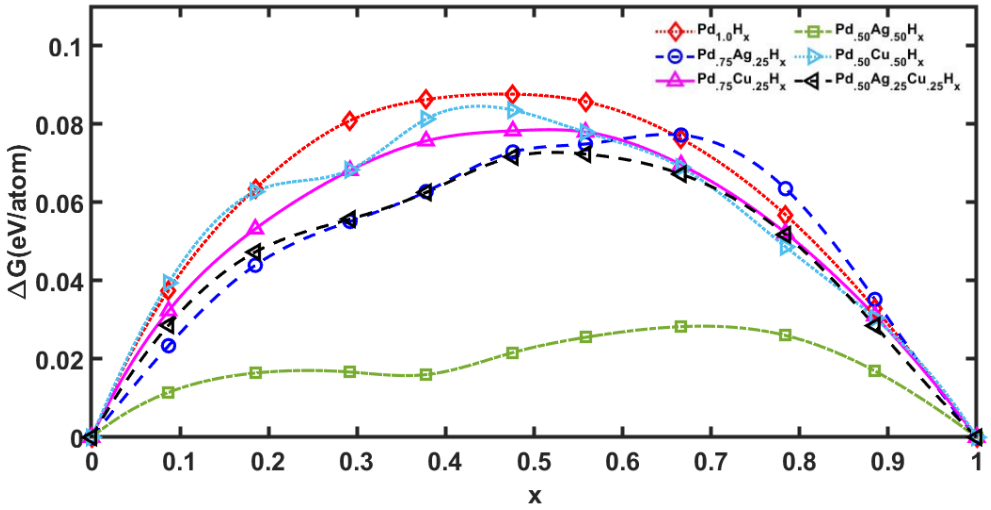

Figure 17: Gibbs free energy plot for different structures at 0 K.

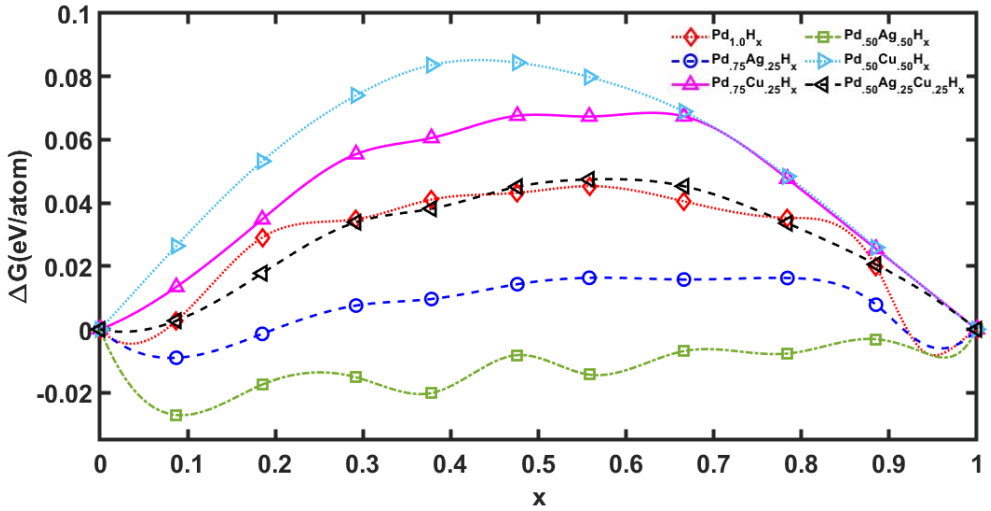

Figure 18: Gibbs free energy plot for different structures at 300 K.

CHAPTER 5

CONCLUSION

In this research, the central atom method was used to fit fully analytical Pd-Ag-Cu-H EAM potentials without utilizing the time-intensive MD simulations during the fitting process. The potentials were efficient in minimizing the objective functions during the fitting calculations, and the number of fitting parameters were reduced compared to previously developed EAM potentials. There were six fitting parameters for each Pd-Pd, Ag-Ag and Cu-Cu EAM potential, two scaling factors calculated by a mixing rule for each Pd-Ag, Pd-Cu and Ag-Cu pair interaction, 10 fitting parameters for H-H, and 4 for each Pd-H, Ag-H and Cu-H EAM potential. Our MD simulation results validated that these EAM potentials can be applied accurately in further simulations.

The experimentally obtained heat of solutions values were used in fitting the Pd-Ag, Pd-Cu and Ag-Cu pair potentials. The Ag-H and Cu-H EAM potentials were fitted to the cohesive energies for 14 $Pd_{0.75}Ag_{0.25}H_x$ and 14 $Pd_{0.75}Cu_{0.25}H_x$ structures, obtained from ab initio SIESTA simulations. The MD simulations utilizing LAMMPS demonstrated that our lattice constant and cohesive energy results for $Pd_{0.75}Ag_{0.25}H_x$ and $Pd_{0.75}Cu_{0.25}H_x$ structures were consistent with the ab initio fitting data for most of the H concentrations. The MD results for the $Pd_{1-y}Ag_yH_x$, $Pd_{1-y}Cu_yH_x$ and $Pd_{1-y-z}Ag_yCu_zH_x$ structures also demonstrated a consistent trend with our previously obtained values for the $Pd_{1.00}H_x$ hydride. The elastic constants trend was as expected, with the bulk modulus decreasing with increasing H concentration. As with $Pd_{1.00}H_x$, dynamic stability testing for the $Pd_{1-y-z}Ag_yCu_zH_x$ quaternary structures also predicted H atoms transferring from higher energy TE sites to lower energy OC sites. Our EAM potentials also captured the existence

of a miscibility gap for the $Pd_{1-y-z}Ag_yCu_zH_x$ and predicted it to narrow and disappear when Ag and Cu concentration increases as was predicted by the experimental findings.

REFERENCES

[1] G. Nelin, "A Neutron Diffraction Study of Palladium Hydride," *physica status solidi (b)*, vol. 45, no. 2, pp. 527–536, 1971.

[2] R. Caputo and A. Alavi, "Where do the H atoms reside in PdH_X systems?" *Molecular Physics*, vol. 101, no. 11, pp. 1781–1787, Jun. 2003.

[3] F. D. Manchester, A. San-Martin, and J. M. Pitre, "The H-Pd (hydrogen-palladium) System," *JPE*, vol. 15, no. 1, pp. 62–83, Feb. 1994.

[4] X. W. Zhou, J. A. Zimmerman, B. M. Wong, and J. J. Hoyt, "An embedded-atom method interatomic potential for Pd–H alloys," *Journal of Materials Research*, vol. 23, no. 3, pp. 704–718, Mar. 2008.

[5] L. M. Hale, B. M. Wong, J. A. Zimmerman, and X. W. Zhou, "Atomistic potentials for palladium–silver hydrides," *Modelling Simul. Mater. Sci. Eng.*, vol. 21, no. 4, p. 045005, Apr. 2013.

[6] D. K. Hsu and R. G. Leisure, "Elastic constants of palladium and β phase palladium hydride between 4 and 300 K," *Phys. Rev. B*, vol. 20, no. 4, pp. 1339–1344, Aug. 1979.

[7] W. Dong, V. Ledentu, Ph. Sautet, A. Eichler, and J. Hafner, "Hydrogen adsorption on palladium: a comparative theoretical study of different surfaces," *Surface Science*, vol. 411, no. 1, pp. 123–136, Aug. 1998.

[8] M. V. Goltsova, Yu. A. Artemenko, and V. I. Zaitsev, "Kinetics and morphology of the reverse β→α hydride transformation in thermodynamically open Pd–H system," *Journal of Alloys and Compounds*, vol. 293–295, pp. 379–384, Dec. 1999.

[9] G. Alefeld and J. Völkl, Eds., *Hydrogen in Metals II: Application-Oriented Properties*. Berlin Heidelberg: Springer-Verlag, 1978.

[10] E. L. Muetterties, "Transition metal--hydrogen interaction," *Transition Metal Hydrides*, 1971.

[11] B. Siegel, G.G. Libowitz, W.M. Mueller, and J.P. Blackledge. *Metal Hydrides.* Academic Press, New York, 1968.

[12] Y. Fukai, *The Metal-Hydrogen System: Basic Bulk Properties*. Berlin Heidelberg: Springer-Verlag, 1993.

[13] R. Lässer, *Tritium and Helium-3 in Metals*. Berlin Heidelberg: Springer-Verlag, vol. 9, 1989.

[14] G. Alefeld and J. Völkl, Eds., *Hydrogen in Metals I: Basic Properties*. Berlin Heidelberg: Springer-Verlag, vol.1, 1978.

[15] R. Povel, K. Feucht, W. Gelse, and G. Withalm, "Hydrogen Fuel for Motorcars," *Interdisciplinary Science Reviews*, vol. 14, no. 4, pp. 365–373, Dec. 1989.

[16] A. G. Knapton, "Palladium alloys for hydrogen diffusion membranes." *Platinum Met*, vol. 21, no. 2, pp. 44–50, 1977.

[17] B. D. Adams and A. Chen, "The role of palladium in a hydrogen economy," *Materials Today*, vol. 14, no. 6, pp. 282–289, Jun. 2011.

[18] G. Jimenez, E. Dillon, J. Dahlmeyer, T. Garrison, T. Garrison, S. Darkey, K. Wald, J. Kubik, D. Paciulli, M. Talukder, J. Nott, M. Ferrer, J. Prinke, P. Villaneuva, F. Massicotte, K. Rebeiz, S. Nesbit and A. Craft, "A Comparative Assessment of Hydrogen Embrittlement: Palladium and Palladium-Silver (25 Weight% Silver) Subjected to Hydrogen

Absorption/Desorption Cycling," *Advances in Chemical Engineering and Science*, vol. 06, no. 03, Art. no. 03, 2016.

[19] E. Foletto, J. Silveira, and S. L. Jahn, "Preparation of palladium-silver alloy membranes for hydrogen permeation," *Latin American applied research*, vol. 38, pp. 79-84, Mar. 2008.

[20] C. H. Chen and Y. H. Ma, "The effect of H2S on the performance of Pd and Pd/Au composite membrane," *Journal of Membrane Science*, vol. 362, no. 1, pp. 535–544, Oct. 2010.

[21] C. P. O'Brien, A. J. Gellman, B. D. Morreale, and J. B. Miller, "The hydrogen permeability of Pd4S," *Journal of Membrane Science*, vol. 371, no. 1, pp. 263–267, Apr. 2011.

[22] J. Gabitto and C. Tsouris, "Modeling Sulfur Poisoning of Palladium Membranes Used for Hydrogen Separation," *International Journal of Chemical Engineering*, Mar. 03, 2019.

[23] T. H. Nguyen, S. Mori, and M. Suzuki, "Hydrogen permeance and the effect of H2O and CO on the permeability of Pd0.75Ag0.25 membranes under gas-driven permeation and plasma-driven permeation," *Chemical Engineering Journal*, vol. 155, no. 1, pp. 55–61, Dec. 2009.

[24] H. Amandusson, L. -G. Ekedahl, and H. Dannetun, "Hydrogen permeation through surface modified Pd and PdAg membranes," *Journal of Membrane Science*, vol. 193, no. 1, pp. 35–47, Oct. 2001.

[25] L. Semidey-Flecha and D. S. Sholl, "Combining density functional theory and cluster expansion methods to predict H2 permeance through Pd-based binary alloy membranes," *J. Chem. Phys.*, vol. 128, no. 14, p. 144701, Apr. 2008.

[26] L. Qin and C. Jiang, “First-principles based modeling of hydrogen permeation through Pd–Cu alloys,” *International Journal of Hydrogen Energy*, vol. 37, no. 17, pp. 12760–12764, Sep. 2012.

[27] A. Weiss, S. Ramaprabhu, and N. Rajalakshmi, “Hydrogen Solubility and Thermodynamics of Hydrogen Absorption in Palladium-rich Binary Pd1−xZx and Ternary Pd1−x−yZxZ′y Solid Solution Alloys,” *Zeitschrift für Physikalische Chemie*, vol. 199, no. 2, pp. 165–212, Feb. 1997.

[28] F. A. Lewis, “The Palladium Hydrogen.” *System, Academic Press, London, New York*, 1967.

[29] S. Adhikari and S. Fernando, “Hydrogen Membrane Separation Techniques,” *Ind. Eng. Chem. Res.*, vol. 45, no. 3, pp. 875–881, Feb. 2006.

[30] M. H. Martin, J. Galipaud, A. Tranchot, L. Roué, and D. Guay, “Measurements of hydrogen solubility in CuxPd100−x thin films,” *Electrochimica Acta*, vol. 90, pp. 615–622, Feb. 2013.

[31] S. Ö. Kart and A. Erbay, “Molecular dynamics study of Cu-Pd ordered alloys,” *Journal of Achievements in Materials and Manufacturing Engineering*, vol. 31, no. 1, p. 7, 2008.

[32] D. L. Mckinley, “Method for hydrogen separation and purification,” US3439474A, Apr. 22, 1969.

[33] F. Illas, N. López, J. M. Ricart, A. Clotet, J. C. Conesa, and M. Fernández-García, “Interaction of CO and NO with PdCu(111) Surfaces,” *J. Phys. Chem. B*, vol. 102, no. 41, pp. 8017–8023, Oct. 1998.

[34] A. Rochefort, M. Abon, P. Delichère, and J. C. Bertolini, “Alloying effect on the adsorption properties of Pd50Cu50{111} single crystal surface,” *Surface Science*, vol. 294, no. 1, pp. 43–52, Sep. 1993.

[35] C. P. O’Brien and I. C. Lee, “The interaction of CO with PdCu hydrogen separation membranes: An operando infrared spectroscopy study,” *Catalysis Today*, vol. 336, pp. 216–222, Oct. 2019.

[36] L. Yuan, A. Goldbach, and H. Xu, “Segregation and H2 transport rate control in body-centered cubic PdCu membranes,” *The Journal of Physical Chemistry B*, vol. 111, no. 37, pp. 10952–10958, 2007.

[37] B. H. Guerreiro, M. H. Martin, L. Roué, and D. Guay, “Hydrogen permeability of PdCuAu membranes prepared from mechanically-alloyed powders,” *Journal of Membrane Science*, vol. 509, pp. 68–82, Jul. 2016.

[38] L. C. Liu, J. W. Wang, Y. H. He, and H. R. Gong, “Solubility, diffusivity, and permeability of hydrogen at PdCu phases,” *Journal of Membrane Science*, vol. 542, pp. 24–30, Nov. 2017.

[39] F. Roa, M. J. Block, and J. D. Way, “The influence of alloy composition on the H2 flux of composite Pd Cu membranes,” *Desalination*, vol. 147, no. 1, pp. 411–416, Sep. 2002.

[40] A. M. Tarditi, F. Braun, and L. M. Cornaglia, “Novel PdAgCu ternary alloy: Hydrogen permeation and surface properties,” *Applied Surface Science*, vol. 257, no. 15, pp. 6626–6635, May 2011.

[41] L. Semidey-Flecha, C. Ling, and D. S. Sholl, “Detailed first-principles models of hydrogen permeation through PdCu-based ternary alloys,” *Journal of Membrane Science*, vol. 362, no. 1, pp. 384–392, Oct. 2010.

[42] C. Ling, L. Semidey-Flecha, and D. S. Sholl, “First-principles screening of PdCuAg ternary alloys as H2 purification membranes,” *Journal of Membrane Science*, vol. 371, no. 1, pp. 189–196, Apr. 2011.

[43] L. Yuan, A. Goldbach, and H. Xu, “Real-time monitoring of metal deposition and segregation phenomena during preparation of PdCu membranes,” *Journal of Membrane Science*, vol. 322, no. 1, pp. 39–45, Sep. 2008.

[44] L. Zhao, A. Goldbach, and H. Xu, “Tailoring palladium alloy membranes for hydrogen separation from sulfur contaminated gas streams,” *Journal of Membrane Science*, vol. 507, pp. 55–62, Jun. 2016.

[45] M. Foiles and J. J. Hoyt, “Computer Simulation of Bubble Growth in Metals Due to He,” SAND2001-0661, 780304, Mar. 2001

[46] I. Hijazi, Y. Zhang, and R. Fuller, “A simple embedded atom potential for Pd-H alloys,” *Molecular Simulation*, vol. 44, no. 17, pp. 1371–1379, Nov. 2018.

[47] R. Fuller and I. Hijazi, “Simple Pd-Ag-H EAM Potentials for Hydrogen Storage Applications,” presented at the ASME 2019 Pressure Vessels & Piping Conference, Nov. 2019.

[48] M. S. Daw and M. I. Baskes, “Semiempirical, Quantum Mechanical Calculation of Hydrogen Embrittlement in Metals,” *Phys. Rev. Lett.*, vol. 50, no. 17, pp. 1285–1288, Apr. 1983.

[49] I. Hijazi and Y. H. Park, “Consistent Analytic Embedded Atom Potential for Face-Centered Cubic Metals and Alloys,” *J. Mater. Sci. Technol*, vol. 25, pp. 835-846, Nov. 2009.

[50] Y. H. Park and I. A. Hijazi, “Simple analytic embedded atom potential for FCC materials,” *International Journal of Microstructure and Materials Properties*, vol. 6, no. 5, pp. 378–396, Jan. 2011.

[51] I. A. Hijazi and Y. H. Park, “Structure of pure metallic nanoclusters: Monte Carlo simulation and ab initio study,” *Eur. Phys. J. D*, vol. 59, no. 2, pp. 215–221, Aug. 2010.

[52] Y. H. Park and I. A. Hijazi, “Critical size of transitional copper clusters for ground state structure determination: empirical and ab initio study,” *Molecular Simulation*, vol. 38, no. 3, pp. 241–247, Mar. 2012.

[53] I. A. Hijazi, and Y. H. Park. “Structural, Electronic and Magnetic Properties of Pure Metallic and Bimetallic Nanoclusters: Empirical and Density Functional Studies.” *Molecular Simulation.* vol 38, no. 6, pp. 505, 2013.

[54] J. H. Rose, J. R. Smith, F. Guinea, and J. Ferrante, “Universal features of the equation of state of metals,” *Phys. Rev. B*, vol. 29, no. 6, pp. 2963–2969, Mar. 1984.

[55] S. M. Foiles, M. I. Baskes, and M. S. Daw, “Embedded-atom-method functions for the fcc metals Cu, Ag, Au, Ni, Pd, Pt, and their alloys,” *Phys. Rev. B*, vol. 33, no. 12, pp. 7983–7991, Jun. 1986.

[56] R. A. Johnson, “Alloy models with the embedded-atom method,” *Phys. Rev. B*, vol. 39, no. 17, pp. 12554–12559, Jun. 1989.

[57] A. Thompson, “in.elastic.” Sandia National Laboratory. Available from: https://github.com/lammps/lammps/tree/master/examples/ELASTIC

[58] J. Cai, and Y. Y. Ye. “Simple analytical embedded-atom-potential model including a long-range force for fcc metals and their alloys." Physical Review B 54.12 (1996) 8398.

[59] O. M. Løvvik and R. A. Olsen, “Density functional calculations on hydrogen in palladium–silver alloys,” *Journal of Alloys and Compounds*, vol. 330–332, pp. 332–337, Jan. 2002.

[60] L. M. Hale, B. M. Wong, J.A. Zimmerman, and X.W. Zhou. Interatomic Potentials Repository Project. Available from: https://www.ctcms.nist.gov/potentials/system/Ag-H-Pd/

[61] E. K. Delczeg-Czirjak, L. Delczeg, M. Ropo, K. Kokko, M. P. J. Punkkinen, B. Johansson, and L. Vitos, “Ab initio study of the elastic anomalies in Pd-Ag alloys,” *Phys. Rev. B*, vol. 79, no. 8, p. 085107, Feb. 2009.

[62] S. Uemiya, T. Matsuda, and E. Kikuchi, “Hydrogen permeable palladium-silver alloy membrane supported on porous ceramics,” *Journal of Membrane Science*, vol. 56, no. 3, pp. 315–325, Mar. 1991.

[63] R. Burch, “On the role of silver atoms in the absorption of hydrogen by palladium-silver alloys,” *Solid State Communications*, vol. 7, no. 18, pp. 1313–1317, Sep. 1969.

[64] I. Karakaya, and W. T. Thompson. “The Ag−Pd (Silver-Palladium) system.” *Bulletin of alloy phase diagrams.* vol. 9, no. 3, pp. 237-243, 1988.

[65] C. Wei, F. T. Kong, and H. R. Gong, “Phase stability and elastic property of PdH and PdCuH phases,” *International Journal of Hydrogen Energy*, vol. 38, no. 36, pp. 16485–16494, Dec. 2013.

[66] “ABINIT.” Accessed Dec. 31, 2020. [Online]. Available: https://www.abinit.org/downloads/psp-links/pseudopotentials.

[67] R. Fuller, “Improved embedded atom method potentials for metal hydride systems,” *Theses, Dissertations and Capstones*, Jan. 2018, [Online]. Available: https://mds.marshall.edu/etd/1124.

[68] N. V. Ilawe, J. A. Zimmerman, and B. M. Wong, “Breaking Badly: DFT-D2 Gives Sizeable Errors for Tensile Strengths in Palladium-Hydride Solids,” *J. Chem. Theory Comput.*, vol. 11, no. 11, pp. 5426–5435, Nov. 2015.

[69] R. B. Schwarz, H. T. Bach, U. Harms, and D. Tuggle, “Elastic properties of Pd–hydrogen, Pd–deuterium, and Pd–tritium single crystals,” *Acta Materialia*, vol. 53, no. 3, pp. 569–580, Feb. 2005.

[70] W. Zhong, Y. Cai, and D. Tomanek. “Computer simulation of hydrogen embrittlement in metals.” *Nature.* vol. 362, no. 6419 (1993) 435-437.

[71] A. Kulprathipanja, G. O. Alptekin, J. L. Falconer, and J. D. Way, “Pd and Pd–Cu membranes: inhibition of H2 permeation by H2S,” *Journal of Membrane Science*, vol. 254, no. 1, pp. 49–62, Jun. 2005.

[72] V. E. Antonov, “Phase transformations, crystal and magnetic structures of high-pressure hydrides of d-metals,” *Journal of Alloys and Compounds*, vol. 330–332, pp. 110–116, Jan. 2002.

[73] T. B. Flanagan, S. Luo, and J. D. Clewley, “Calorimetric enthalpies for the reaction of H2 with Pd–Cu alloys at 303 K,” *Journal of Alloys and Compounds,* vol. 356–357, pp. 13–16, Aug. 2003.

[74] W. Huang, S. M. Opalka, D. Wang, and T. B. Flanagan, “Thermodynamic modelling of the Cu–Pd–H system,” *Calphad,* vol. 31, no. 3, pp. 315–329, Sep. 2007.

APPENDIX A:

INSTITUTIONAL REVIEW BOARD LETTER.

MARSHALL UNIVERSITY.
www.marshall.edu

Office of Research Integrity

September 29, 2020

Chaonan Zhang
432 Stoneheath Dr.
Barboursville, WV 25504

Dear Chaonan:

This letter is in response to the submitted thesis abstract entitled "**QUATERNARY HYDRIDES $Pd_{1-y-z}Ag_yCu_zH_x$ EMBEDDED ATOM METHOD POTENTIALS FOR HYDROGEN ENERGY APPLICATIONS.**" After assessing the abstract, it has been deemed not to be human subject research and therefore exempt from oversight of the Marshall University Institutional Review Board (IRB). The Code of Federal Regulations (45CFR46) has set forth the criteria utilized in making this determination. Since the information in this study does not involve human subjects as defined in the above referenced instruction, it is not considered human subject research. If there are any changes to the abstract you provided then you would need to resubmit that information to the Office of Research Integrity for review and a determination.

I appreciate your willingness to submit the abstract for determination. Please feel free to contact the Office of Research Integrity if you have any questions regarding future protocols that may require IRB review.

Sincerely,

Bruce F. Day, ThD, CIP
Director

APPENDIX B:

CALCULATIONS FOR INTERSTITIAL SOLID SOLUTION

$$r_{ij} = \sqrt{r_i^2 + r_j^2 + r_k^2} = \sqrt{l_1^2 + l_2^2 + l_3^2}\, a = M_{ij} a \tag{1}$$

$$\frac{\partial r_{ij}}{\partial a} = M_{ij} = \frac{r_{ij}}{a} \tag{2}$$

$$\frac{\partial^2 r_{ij}}{\partial a^2} = 0 \tag{3}$$

$$\frac{\partial r_{ij}}{\partial r_i} = \frac{r_i}{\sqrt{r_i^2 + r_j^2 + r_k^2}} = \frac{r_i}{r_{ij}} \tag{4}$$

$$\frac{\partial^2 r_{ij}}{\partial r_i^2} = \frac{r_{ij} - r_i^2 / r_{ij}}{r_{ij}^2} = \frac{1}{r_{ij}} - \frac{r_i^2}{r_{ij}^3} \tag{5}$$

$$\frac{\partial^2 r_{ij}}{\partial r_i \partial r_j} = \frac{0 - r_i r_j / r_{ij}}{r_{ij}^2} = -\frac{r_i r_j}{r_{ij}^3} \tag{6}$$

$$\frac{\partial^2 r_{ij}}{\partial r_i \partial r_j} = \delta_{ij} \frac{1}{r_{ij}} - \frac{r_i r_j}{r_{ij}^3}, \quad \delta_{ij} = \begin{cases} 1 & , i = j \\ 0 & , i \neq j \end{cases} \tag{7}$$

$$\frac{\partial E_c}{\partial a} = \frac{\partial E_c}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial a} = \frac{\partial E_c}{\partial r_{ij}} \frac{r_{ij}}{a} \tag{8}$$

$$\frac{\partial^2 E_c}{\partial a^2} = \frac{\partial}{\partial a}\left(\frac{\partial E}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial a}\right) = \frac{\partial}{\partial a}\left(\frac{\partial E}{\partial r_{ij}}\right)\frac{\partial r_{ij}}{\partial a} + \frac{\partial}{\partial a}\left(\frac{\partial r_{ij}}{\partial a}\right)\frac{\partial E}{\partial r_{ij}}$$

$$= \frac{\partial}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial a}\left(\frac{\partial E}{\partial r_{ij}}\right)\frac{\partial r_{ij}}{\partial a} + \frac{\partial r_{ij}^2}{\partial a^2}\frac{\partial E}{\partial r_{ij}}$$

$$= \frac{\partial^2 E_c}{\partial r_{ij}^2}\left(\frac{\partial r_{ij}}{\partial a}\right)^2 + 0 \cdot \frac{\partial E}{\partial r_{ij}}$$

$$= \frac{\partial^2 E_c}{\partial r_{ij}^2}\left(\frac{\partial r_{ij}}{\partial a}\right)^2$$

$$= \frac{\partial^2 E_c}{\partial r_{ij}^{\ 2}}\left(\frac{r_{ij}}{a}\right)^2 \tag{9}$$

$$\rho_{H,i} = \rho_{H-H,i} + \rho_{H-I,i} \tag{10a}$$

$$\rho_{I,i} = \rho_{I-H,i} + \rho_{I-I,i} \tag{10b}$$

$$\frac{\partial \rho_{H,i}}{\partial r_{ij}} = \frac{\partial \rho_{H\text{-}H,i}}{\partial r_{ij}} + \frac{\partial \rho_{H-I,i}}{\partial r_{ij}} \tag{11a}$$

$$\frac{\partial \rho_{I,i}}{\partial r_{ij}} = \frac{\partial \rho_{I\text{-}I,i}}{\partial r_{ij}} + \frac{\partial \rho_{I-H,i}}{\partial r_{ij}} \tag{11b}$$

$$\frac{\partial^2 \rho_{H,i}}{\partial r_{ij}^{\ 2}} = \frac{\partial^2 \rho_{H\text{-}H,i}}{\partial r_{ij}^{\ 2}} + \frac{\partial^2 \rho_{H-I,i}}{\partial r_{ij}^{\ 2}} \tag{12a}$$

$$\frac{\partial^2 \rho_{I,i}}{\partial r_{ij}^{\ 2}} = \frac{\partial^2 \rho_{I\text{-}I,i}}{\partial r_{ij}^{\ 2}} + \frac{\partial^2 \rho_{I-H,i}}{\partial r_{ij}^{\ 2}} \tag{12b}$$

$$\varphi_{H,i} = \varphi_{H-H,i} + \varphi_{H-I,i} \tag{13a}$$

$$\varphi_{I,i} = \varphi_{I-H,i} + \varphi_{I-I,i} \tag{13b}$$

$$\frac{\partial \varphi_{H,i}}{\partial r_{ij}} = \frac{\partial \varphi_{H\text{-}H,i}}{\partial r_{ij}} + \frac{\partial \varphi_{H-I,i}}{\partial r_{ij}} \tag{14a}$$

$$\frac{\partial \phi_{I,i}}{\partial r_{ij}} = \frac{\partial \phi_{I-I,i}}{\partial r_{ij}} + \frac{\partial \phi_{I-H,i}}{\partial r_{ij}} \tag{14b}$$

$$\frac{\partial^2 \varphi_{H,i}}{\partial r_{ij}^{\ 2}} = \frac{\partial^2 \varphi_{H\text{-}H,i}}{\partial r_{ij}^{\ 2}} + \frac{\partial^2 \varphi_{H-I,i}}{\partial r_{ij}^{\ 2}} \tag{15a}$$

$$\frac{\partial^2 \varphi_{I,i}}{\partial r_{ij}^{\ 2}} = \frac{\partial^2 \varphi_{I\text{-}I,i}}{\partial r_{ij}^{\ 2}} + \frac{\partial^2 \varphi_{I-H,i}}{\partial r_{ij}^{\ 2}} \tag{15b}$$

$$\frac{\partial F_H}{\partial r_{ij}} = \frac{\partial F_H}{\partial \rho_{H,i}} \frac{\partial \rho_{H,i}}{\partial r_{ij}} \tag{16a}$$

$$\frac{\partial F_I}{\partial r_{ij}} = \frac{\partial F_I}{\partial \rho_{I,i}} \frac{\partial \rho_{I,i}}{\partial r_{ij}} \tag{16b}$$

$$\frac{\partial^2 F_H}{\partial r_{ij}^{\ 2}} = \frac{\partial}{\partial r_{ij}}\left(\frac{\partial F_H}{\partial \rho_{H,i}} \frac{\partial \rho_{H,i}}{\partial r_{ij}}\right) = \frac{\partial}{\partial r_{ij}}\left(\frac{\partial F_H}{\partial \rho_{H,i}}\right)\frac{\partial \rho_{H,i}}{\partial r_{ij}} + \frac{\partial F_H}{\partial \rho_{H,i}} \frac{\partial^2 \rho_{H,i}}{\partial r_{ij}^{\ 2}} = \frac{\partial^2 F_H}{\partial \rho_{H,i}^2}\left(\frac{\partial \rho_{H,i}}{\partial r_{ij}}\right)^2 + \frac{\partial F_H}{\partial \rho_{H,i}} \frac{\partial^2 \rho_{H,i}}{\partial r_{ij}^{\ 2}} \tag{17a}$$

$$\frac{\partial^2 F_I}{\partial r_{ij}^{\ 2}} = \frac{\partial^2 F_I}{\partial \rho_{I,i}^2}\left(\frac{\partial \rho_{I,i}}{\partial r_{ij}}\right)^2 + \frac{\partial F_I}{\partial \rho_{I,i}} \frac{\partial^2 \rho_{I,i}}{\partial r_{ij}^{\ 2}} \tag{17b}$$

$$\frac{\partial E_H}{\partial r_{ij}} = \frac{\partial F_H}{\partial r_{ij}} + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^{N_H} \frac{\partial \varphi_{H-H,ij}}{\partial r_{ij}} + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^{N_I} \frac{\partial \varphi_{H-I,ij}}{\partial r_{ij}} \tag{18a}$$

$$\frac{\partial E_I}{\partial r_{ij}} = \frac{\partial F_I}{\partial r_{ij}} + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^{N_I} \frac{\partial \varphi_{I-I,ij}}{\partial r_{ij}} + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^{N_H} \frac{\partial \varphi_{I-H,ij}}{\partial r_{ij}} \tag{18b}$$

$$\frac{\partial^2 E_H}{\partial r_{ij}^2} = \frac{\partial^2 F_H}{\partial r_{ij}^2} + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^{N_H} \frac{\partial^2 \varphi_{H-H,ij}}{\partial r_{ij}^2} + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^{N_I} \frac{\partial^2 \varphi_{H-I,ij}}{\partial r_{ij}^2} \tag{19a}$$

$$\frac{\partial^2 E_I}{\partial r_{ij}^2} = \frac{\partial^2 F_I}{\partial r_{ij}^2} + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^{N_I} \frac{\partial^2 \varphi_{I-I,ij}}{\partial r_{ij}^2} + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^{N_H} \frac{\partial^2 \varphi_{I-H,ij}}{\partial r_{ij}^2} \tag{19b}$$

$$\frac{\partial E_H}{\partial r_i} = \frac{\partial E_H}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial r_i} = \frac{\partial E_H}{\partial r_{ij}} \frac{r_i}{r_{ij}} \tag{20a}$$

$$\frac{\partial E_I}{\partial r_i} = \frac{\partial E_I}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial r_i} = \frac{\partial E_I}{\partial r_{ij}} \frac{r_i}{r_{ij}} \tag{20b}$$

$$\frac{\partial E_H}{\partial r_{\mathrm{j}}} = \frac{\partial E_H}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial r_j} = \frac{\partial E_H}{\partial r_{ij}} \frac{r_j}{r_{ij}} \tag{21a}$$

$$\frac{\partial E_I}{\partial r_{\mathrm{j}}} = \frac{\partial E_I}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial r_j} = \frac{\partial E_I}{\partial r_{ij}} \frac{r_j}{r_{ij}} \tag{21b}$$

$$\frac{\partial^2 E_H}{\partial r_i \partial r_j} = \frac{\partial}{\partial r_j}\left(\frac{\partial E_H}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial r_i}\right) = \frac{\partial}{\partial r_j}\left(\frac{\partial E_H}{\partial r_{ij}}\right)\frac{\partial r_{ij}}{\partial r_i} + \frac{\partial E_H}{\partial r_{ij}} \frac{\partial}{\partial r_j}\left(\frac{\partial r_{ij}}{\partial r_i}\right)$$

$$= \frac{\partial}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial r_j}\left(\frac{\partial E_H}{\partial r_{ij}}\right)\frac{\partial r_{ij}}{\partial r_i} + \frac{\partial E_H}{\partial r_{ij}} \frac{\partial}{\partial r_j}\left(\frac{\partial r_{ij}}{\partial r_i}\right)$$

$$= \left(\frac{\partial^2 E_H}{\partial r_{ij}^2}\right)\frac{\partial r_{ij}}{\partial r_i} \frac{\partial r_{ij}}{\partial r_j} + \frac{\partial E_H}{\partial r_{ij}}\left(\frac{\partial^2 r_{ij}}{\partial r_i \partial r_j}\right)$$

$$= \left(\frac{\partial^2 E_H}{\partial r_{ij}^2}\right)\frac{r_i r_j}{r_{ij}^2} + \frac{\partial E_H}{\partial r_{ij}}\left(\delta_{ij}\frac{1}{r_{ij}} - \frac{r_i r_j}{r_{ij}^3}\right) \tag{22a}$$

$$\frac{\partial^2 E_I}{\partial r_i \partial r_j} = \frac{\partial}{\partial r_j}\left(\frac{\partial E_I}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial r_i}\right) = \frac{\partial}{\partial r_j}\left(\frac{\partial E_I}{\partial r_{ij}}\right)\frac{\partial r_{ij}}{\partial r_i} + \frac{\partial E_I}{\partial r_{ij}} \frac{\partial}{\partial r_j}\left(\frac{\partial r_{ij}}{\partial r_i}\right)$$

$$= \frac{\partial}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial r_j}\left(\frac{\partial E_I}{\partial r_{ij}}\right)\frac{\partial r_{ij}}{\partial r_i} + \frac{\partial E_I}{\partial r_{ij}} \frac{\partial}{\partial r_j}\left(\frac{\partial r_{ij}}{\partial r_i}\right)$$

$$= \left(\frac{\partial^2 E_I}{\partial r_{ij}^2}\right)\frac{\partial r_{ij}}{\partial r_i} \frac{\partial r_{ij}}{\partial r_j} + \frac{\partial E_I}{\partial r_{ij}}\left(\frac{\partial^2 r_{ij}}{\partial r_i \partial r_j}\right)$$

$$= \left(\frac{\partial^2 E_I}{\partial r_{ij}^2}\right)\frac{r_i r_j}{r_{ij}^2} + \frac{\partial E_I}{\partial r_{ij}}\left(\delta_{ij}\frac{1}{r_{ij}} - \frac{r_i r_j}{r_{ij}^3}\right) \tag{22b}$$

MIXING RULE PAIR POTENTIAL DERIVATES

$$\phi_{ab}(r) = \tfrac{1}{2}\left[\tfrac{f_b(r)}{f_a(r)}\phi_{aa}(r) + \tfrac{f_a(r)}{f_b(r)}\phi_{bb}(r)\right] = \tfrac{1}{2}\left[f_b f_a{}^{-1}\phi_{aa} + f_a f_b{}^{-1}\phi_{bb}\right] \tag{23}$$

$$\frac{\partial^2\phi_{ab}}{\partial r^2} = \frac{1}{2}\frac{\partial}{\partial r}\begin{Bmatrix}\left[\left(-f_b\dot{f}_a f_a^{-2} + \dot{f}_b f_a^{-1}\right)\phi_{aa} + \left(f_b f_a^{-1}\right)\dot{\phi}_{aa}\right] + \\ \left[\left(-f_a\dot{f}_b f_b^{-2} + \dot{f}_a f_b^{-1}\right)\phi_{bb} + \left(f_a f_b^{-1}\right)\dot{\phi}_{bb}\right]\end{Bmatrix} \tag{24}$$

$$\frac{\partial^2\phi_{ab}}{\partial r^2} = \frac{1}{2}\begin{Bmatrix}\left\{-1\left[\dot{f}_b\dot{f}_a f_a^{-2} + f_b\ddot{f}_a f_a^{-2} - 2f_b\dot{f}_a^2 f_a^{-3}\right] + \left[\ddot{f}_b f_a^{-1} - 1\dot{f}_b f_a^{-2}\dot{f}_a\right]\right\}\phi_{aa} \\ +\left(-f_b\dot{f}_a f_a^{-2} + \dot{f}_b f_a^{-1}\right)\dot{\phi}_{aa} + \dot{f}_b f_a^{-1}\dot{\phi}_{aa} - 1f_b f_a^{-2}\dot{f}_a\dot{\phi}_{aa} + f_b f_a^{-1}\ddot{\phi}_{aa} \\ + \left\{-1\left[\dot{f}_a\dot{f}_b f_b^{-2} + f_a\ddot{f}_b f_b^{-2} - 2f_a f_b^2 f_b^{-3}\right] + \left[\ddot{f}_a f_b^{-1} - 1\dot{f}_a f_b^{-2}\dot{f}_b\right]\right\}\phi_{bb} \\ +\left(-f_a\dot{f}_b f_b^{-2} + \dot{f}_a f_b^{-1}\right)\dot{\phi}_{bb} + \dot{f}_a f_b^{-1}\dot{\phi}_{bb} - 1f_a f_b^{-2}\dot{f}_b\dot{\phi}_{bb} + f_a f_b^{-1}\ddot{\phi}_{bb}\end{Bmatrix} \tag{25}$$

TOTAL ENERGY

$$E_{total} = \sum_{i=1}^{N_H} F_{H,i}(\rho_{H,i}) + \frac{1}{2}\sum_{i=1}^{N_H}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\varphi_{H-H,ij}(r_{ij}) + \frac{1}{2}\sum_{i=1}^{N_H}\sum_{\substack{j=1\\ j\neq i}}^{N_I}\varphi_{H-I,ij}(r_{ij}) + \sum_{i=1}^{N_I} F_{I,i}(\rho_{H,i}) + \frac{1}{2}\sum_{i=1}^{N_I}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\varphi_{I-H,ij}(r_{ij}) + \frac{1}{2}\sum_{i=1}^{N_I}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\varphi_{I-H,ij}(r_{ij}) \tag{26}$$

COHESIVE ENERGY

$$E_c = \frac{1}{N_H + N_I}\left[N_H E_H + N_I E_I\right] \tag{27}$$

$$E_c = \frac{1}{N_H + N_I}\begin{bmatrix} N_H\left(F_{H,i}(\rho_{H,i}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\varphi_{H-H,ij}(r_{ij}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_I}\varphi_{H-I,ij}(r_{ij})\right) + \\ N_I\left(F_{I,i}(\rho_{H,i}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\varphi_{I-H,ij}(r_{ij}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\varphi_{I-H,ij}(r_{ij})\right)\end{bmatrix} \tag{28}$$

$$= \frac{1}{\mathrm{x}+y}\begin{bmatrix} \mathrm{x}\left(F_{H,i}(\rho_{H,i}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\varphi_{H-H,ij}(r_{ij}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_I}\varphi_{H-I,ij}(r_{ij})\right) + \\ y\left(F_{I,i}(\rho_{I,i}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\varphi_{I-H,ij}(r_{ij}) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_I}\varphi_{I-I,ij}(r_{ij})\right)\end{bmatrix} \tag{29}$$

or with three types: a, b, and c

$$E_c = \frac{1}{\mathrm{x}+y+z}\left[\begin{array}{l} \mathrm{x}\left(F_{a,i}(\rho_{a,i})+\frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_a}\varphi_{a-a,ij}(r_{ij})+\frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_b}\varphi_{a-b,ij}(r_{ij})+\frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_c}\varphi_{a-c,ij}(r_{ij})\right)+ \\ y\left(F_{b,i}(\rho_{b,i})+\frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_b}\varphi_{b-b,ij}(r_{ij})+\frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_a}\varphi_{b-a,ij}(r_{ij})+\frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_c}\varphi_{b-c,ij}(r_{ij})\right)+ \\ z\left(F_{c,i}(\rho_{c,i})+\frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_c}\varphi_{c-c,ij}(r_{ij})+\frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_a}\varphi_{c-a,ij}(r_{ij})+\frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_b}\varphi_{c-b,ij}(r_{ij})\right) \end{array}\right] \tag{30}$$

$$\rho_{\mathrm{a},i} = \rho_{a-a,i}+\rho_{a-b,i}+\rho_{a-c,i} = \sum_{\substack{j=1\\ j\neq i}}^{N_a} f_a(r_{ij})+\sum_{\substack{j=1\\ j\neq i}}^{N_b} f_b(r_{ij})+\sum_{\substack{j=1\\ j\neq i}}^{N_c} f_c(r_{ij}) \tag{31}$$

$$\rho_{\mathrm{b},i} = \rho_{b-a,i}+\rho_{b-b,i}+\rho_{b-c,i} = \sum_{\substack{j=1\\ j\neq i}}^{N_a} f_a(r_{ij})+\sum_{\substack{j=1\\ j\neq i}}^{N_b} f_b(r_{ij})+\sum_{\substack{j=1\\ j\neq i}}^{N_c} f_c(r_{ij}) \tag{32}$$

$$\rho_{\mathrm{c},i} = \rho_{c-a,i}+\rho_{c-b,i}+\rho_{c-c,i} = \sum_{\substack{j=1\\ j\neq i}}^{N_a} f_a(r_{ij})+\sum_{\substack{j=1\\ j\neq i}}^{N_b} f_b(r_{ij})+\sum_{\substack{j=1\\ j\neq i}}^{N_c} f_c(r_{ij}) \tag{33}$$

BULK MODULUS

$$B = \frac{\frac{\partial P}{\partial V}}{V_o} = V_o\frac{\partial P}{\partial V} = V_o\frac{\partial^2 E}{\partial V^2} \tag{34}$$

$$\frac{\partial E}{\partial V} = \frac{\partial}{\partial a}\frac{\partial a}{\partial V}E = \frac{\partial E}{\partial a}\frac{\partial a}{\partial V} \tag{35}$$

$$\frac{\partial^2 E}{\partial V^2} = \frac{\partial}{\partial V}\left(\frac{\partial E}{\partial a}\frac{\partial a}{\partial V}\right) = \frac{\partial}{\partial V}\left(\frac{\partial E}{\partial a}\right)\frac{\partial a}{\partial V}+\frac{\partial}{\partial V}\left(\frac{\partial a}{\partial V}\right)\frac{\partial E}{\partial a}$$

$$= \frac{\partial}{\partial V}\left(\frac{\partial E}{\partial a}\right)\frac{\partial a}{\partial V}+\frac{\partial}{\partial V}\left(\frac{\partial a}{\partial V}\right)\frac{\partial E}{\partial a}$$

$$= \frac{\partial}{\partial a}\frac{\partial a}{\partial V}\left(\frac{\partial E}{\partial a}\right)\frac{\partial a}{\partial V} + \frac{\partial}{\partial V}\left(\frac{\partial a}{\partial V}\right) * 0$$

$$= \frac{\partial^2 E}{\partial a^2}\left(\frac{\partial a}{\partial V}\right)^2 \tag{36}$$

$$B = V_o\frac{\partial^2 E}{\partial V^2} = a^3\frac{\partial^2 E}{\partial a^2}\left(\frac{\partial a}{\partial V}\right)^2 = a^3\frac{\partial^2 E}{\partial a^2}\left(\frac{1}{3a^2}\right)^2 = \frac{a^2}{9V_o}\frac{\partial^2 E}{\partial a^2} \tag{37}$$

$$= a^3\frac{\partial^2 E}{\partial a^2}\left(\frac{1}{3a^2}\right)^2 = \frac{a^3}{9a^4}\frac{\partial^2 E}{\partial a^2} = \frac{a^3}{9Va}\frac{\partial^2 E}{\partial a^2} = \frac{a^2}{9V}\frac{\partial^2 E}{\partial a^2}$$

$$= \frac{a^2}{9V}\frac{\partial^2 E}{\partial a^2} = \frac{a^2}{9V}\frac{\partial^2 E_c}{\partial r_{ij}^{\,2}}\left(\frac{r_{ij}}{a}\right)^2$$

$$= \frac{a^2}{9V}\left(\frac{\partial^2 F}{\partial r_{ij}^{\,2}} + \frac{1}{2}\sum_{j\neq i}\frac{\partial^2\phi}{\partial r_{ij}^{\,2}}\right)\left(\frac{r_{ij}}{a}\right)^2 \tag{38}$$

$$= \frac{a^2}{9V}\left(\frac{\partial^2 F}{\partial\rho^2}\left(\frac{\partial\rho}{\partial r_{ij}}\right)^2 + \frac{\partial F}{\partial\rho}\frac{\partial^2\rho}{\partial r_{ij}^{\,2}} + \frac{1}{2}\sum_{j\neq i}\frac{\partial^2\phi}{\partial r_{ij}^{\,2}}\right)\left(\frac{r_{ij}}{a}\right)^2$$

$$= \frac{a^2}{9V}\left[F''(\rho)\left[\frac{\partial\rho}{\partial r_{ij}}\frac{r_{ij}}{a}\right]^2 + F'(\rho)\left[\frac{\partial^2\rho}{\partial r_{ij}^{\,2}}\frac{r_{ij}^2}{a^2}\right] + \frac{1}{2}\sum_{i\neq j,j}\phi''_{ij}(r_{ij})\frac{r_{ij}^2}{a^2}\right] \tag{39}$$

SOLID SOLUTION BULK MODULUS

Substituting Equations (15a), (15b), (17a), (17b) into (31) we get

$$B_I = \frac{a^2}{9V_o}\left[\frac{\partial^2 F_I}{\partial r_{ij}^{\,2}} + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\frac{\partial^2\phi_{I-H,ij}}{\partial r_{ij}^2} + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_I}\frac{\partial^2\phi_{I-I,ij}}{\partial r_{ij}^2}\right]\left(\frac{r_{ij}}{a}\right)^2 \tag{40a}$$

$$B_H = \frac{a^2}{9V_o}\left[\frac{\partial^2 F_H}{\partial r_{ij}^{\,2}} + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\frac{\partial^2\phi_{H-H,ij}}{\partial r_{ij}^2} + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_I}\frac{\partial^2\phi_{H-I,ij}}{\partial r_{ij}^2}\right]\left(\frac{r_{ij}}{a}\right)^2 \tag{40b}$$

$$B_T = B_H + B_I$$

$$= \frac{a^2}{9V_o}\left[\frac{\partial^2 F_H}{\partial r_{ij}^{\,2}} + \frac{\partial^2 F_I}{\partial r_{ij}^{\,2}} + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\frac{\partial^2 \varphi_{H-H,ij}}{\partial r_{ij}^2} + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_I}\frac{\partial^2 \varphi_{H-I,ij}}{\partial r_{ij}^2} + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\frac{\partial^2 \varphi_{I-H,ij}}{\partial r_{ij}^2} + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_I}\frac{\partial^2 \varphi_{I-I,ij}}{\partial r_{ij}^2}\right]\left(\frac{r_{ij}}{a}\right)^2 \tag{41}$$

$$= \frac{a^2}{9V_o}\left[\frac{\partial^2 F_H}{\partial r_{ij}^{\,2}} + \frac{\partial^2 F_I}{\partial r_{ij}^{\,2}} + \frac{1}{2}\left(\sum_{\substack{j=1\\ j\neq i}}^{N_H}\frac{\partial^2 \varphi_{H-H,ij}}{\partial r_{ij}^2} + \sum_{\substack{j=1\\ j\neq i}}^{N_I}\frac{\partial^2 \varphi_{H-I,ij}}{\partial r_{ij}^2} + \sum_{\substack{j=1\\ j\neq i}}^{N_H}\frac{\partial^2 \varphi_{I-H,ij}}{\partial r_{ij}^2} + \sum_{\substack{j=1\\ j\neq i}}^{N_I}\frac{\partial^2 \varphi_{I-I,ij}}{\partial r_{ij}^2}\right)\right]\left(\frac{r_{ij}}{a}\right)^2$$

STRESS

$$\sigma_{H,ij} = \frac{1}{\Omega_o}\left[\frac{\partial E}{\partial r_i} r_j\right] \tag{42}$$

SOLID SOLUTION STRESS

Substituting Equation (20a) into (36), we get

$$\sigma_{H,ij} = \frac{1}{\Omega_o}\left[\frac{\partial E_H}{\partial r_{ij}}\frac{r_i r_j}{r_{ij}}\right] \tag{43}$$

$$\sigma_{H,ij} = \frac{1}{\Omega_o}\left[\frac{\partial F_H}{\partial \rho_{H,i}}\frac{\partial \rho_{H,i}}{\partial r_{ij}} + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\frac{\partial \varphi_{H-H,ij}}{\partial r_{ij}} + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_I}\frac{\partial \varphi_{H-I,ij}}{\partial r_{ij}}\right]\frac{r_i r_j}{r_{ij}} \tag{44}$$

$$\sigma_{H,ij} = \frac{1}{\Omega_o}\left[\frac{\partial F_H}{\partial \rho_{H,i}}\left(\frac{\partial \rho_{H\text{-}H,i}}{\partial r_{ij}} + \frac{\partial \rho_{H-I,i}}{\partial r_{ij}}\right) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\frac{\partial \varphi_{H-H,ij}}{\partial r_{ij}} + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_I}\frac{\partial \varphi_{H-I,ij}}{\partial r_{ij}}\right]\frac{r_i r_j}{r_{ij}} \tag{45}$$

$$\sigma_{I,ij} = \frac{1}{\Omega_o}\left[\frac{\partial F_I}{\partial \rho_{I,i}}\left(\frac{\partial \rho_{I-I,i}}{\partial r_{ij}} + \frac{\partial \rho_{I-H,i}}{\partial r_{ij}}\right) + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_I}\frac{\partial \phi_{I-I,ij}}{\partial r_{ij}} + \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{N_H}\frac{\partial \phi_{I-H,ij}}{\partial r_{ij}}\right]\frac{r_i r_j}{r_{ij}} \tag{46}$$

$$\sigma_{T,ij} = \frac{1}{\Omega_o}\left[\frac{\partial F_H}{\partial \rho_{H,i}}\left(\frac{\partial \rho_{H\text{-}H,i}}{\partial r_{ij}} + \frac{\partial \rho_{H-I,i}}{\partial r_{ij}}\right) + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^{N_H}\frac{\partial \varphi_{H-H,ij}}{\partial r_{ij}} + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^{N_I}\frac{\partial \varphi_{H-I,ij}}{\partial r_{ij}}\right]\frac{r_i r_j}{r_{ij}} + \\ = \frac{1}{\Omega_o}\left[\frac{\partial F_I}{\partial \rho_{I,i}}\left(\frac{\partial \rho_{I-I,\mathrm{i}}}{\partial r_{ij}} + \frac{\partial \rho_{I-H,i}}{\partial r_{ij}}\right) + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^{N_I}\frac{\partial \varphi_{I-I,ij}}{\partial r_{ij}} + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^{N_H}\frac{\partial \varphi_{I-H,ij}}{\partial r_{ij}}\right]\frac{r_i r_j}{r_{ij}} \qquad (47)$$

ELASTIC CONSTANTS

$$C_{\mathrm{ijkl}} = \frac{1}{\Omega_e}\frac{\partial^2 E}{\partial r_i \partial r_j} r_k r_l$$

(48)

Substituting Equations (18a) and (19a) into (42) and Equations (18b) and (19b) into (42), we get

$$C_{\mathrm{ijkl,H}} = \frac{1}{\Omega_e}\left[\left(\frac{r_i r_j}{r_{ij}^2}\right)\frac{\partial^2 E_H}{\partial r_{ij}^2} + \left(\delta_{ik}\frac{1}{r_{ij}} - \frac{r_i r_j}{r_{ij}^3}\right)\frac{\partial E_H}{\partial r_{ij}}\right] r_k r_l \qquad (49a)$$

$$C_{\mathrm{ijkl,I}} = \frac{1}{\Omega_e}\left[\left(\frac{r_i r_j}{r_{ij}^2}\right)\frac{\partial^2 E_I}{\partial r_{ij}^2} + \left(\delta_{ik}\frac{1}{r_{ij}} - \frac{r_i r_j}{r_{ij}^3}\right)\frac{\partial E_I}{\partial r_{ij}}\right] r_k r_l \qquad (49b)$$

Substituting Equations (20) and (22) into (48) and Equations (21) and (23) into (49), we get

$$C_{\mathrm{ijkl,H}} = \frac{1}{\Omega_e}\left[\frac{\partial^2 F_H}{\partial \rho_{u,i}^2}\left(V_{\mathrm{ij}}^{H-H} + V_{\mathrm{ij}}^{H-I}\right)\left(V_{\mathrm{kl}}^{H-H} + V_{\mathrm{kl}}^{H-I}\right) + \frac{\partial F_H}{\partial \rho_{H,i}}\left(W_{\mathrm{ijkl}}^{H-H} + W_{\mathrm{ijkl}}^{H-I}\right) + B_{\mathrm{ijkl}}^{H-H} + B_{\mathrm{ijkl}}^{H-I}\right] \qquad (50a)$$

$$C_{\mathrm{ijkl,I}} = \frac{1}{\Omega_e}\left[\frac{\partial^2 F_I}{\partial \rho_{u,i}^2}\left(V_{\mathrm{ij}}^{I-I} + V_{\mathrm{ij}}^{I-H}\right)\left(V_{\mathrm{kl}}^{I-I} + V_{\mathrm{kl}}^{I-H}\right) + \frac{\partial F_H}{\partial \rho_{H,i}}\left(W_{\mathrm{ijkl}}^{I-I} + W_{\mathrm{ijkl}}^{I-H}\right) + B_{\mathrm{ijkl}}^{I-I} + B_{\mathrm{ijkl}}^{I-H}\right] \qquad (50b)$$

where

$$W_{\mathrm{ijkl}}^{H} = \left(\sum_{\substack{j=1 \\ j\neq i}}^{N_H}\frac{\partial^2 \rho_{H,i}}{\partial r_{ij}^{\,2}} - \sum_{\substack{j=1 \\ j\neq i}}^{N_H}\frac{1}{\mathrm{r}_{ij}}\frac{\partial \rho_{H,i}}{\partial r_{ij}}\right)\frac{r_i r_j r_k r_l}{r_{ij}^2} + \delta_{ik}\frac{r_k r_l}{r_{ij}}\left(\frac{\partial \rho_{H,i}}{\partial r_{ij}}\right) \qquad (51a)$$

$$W^{I}_{\text{ijkl}} = \left(\sum_{\substack{j=1 \\ j\neq i}}^{N_I} \frac{\partial^2 \rho_{I,i}}{\partial r_{ij}^{\,2}} - \sum_{\substack{j=1 \\ j\neq i}}^{N_I} \frac{1}{r_{ij}} \frac{\partial \rho_{I,i}}{\partial r_{ij}} \right) \frac{r_i r_j r_k r_l}{r_{ij}^2} + \delta_{ik} \frac{r_k r_l}{r_{ij}} \left(\frac{\partial \rho_{I,i}}{\partial r_{ij}} \right) \tag{51b}$$

$$B^{H}_{\text{ijkl}} = \frac{1}{2} \left[\sum_{\substack{j=1 \\ j\neq i}}^{N_H} \frac{\partial^2 \varphi_{H,ij}}{\partial r_{ij}^2} - \frac{1}{r_{ij}} \sum_{\substack{j=1 \\ j\neq i}}^{N_H} \frac{\partial \varphi_{H,ij}}{\partial r_{ij}} \right] \frac{r_i r_j r_k r_l}{r_{ij}^2} + \frac{1}{2} \delta_{ik} \left(\sum_{\substack{j=1 \\ j\neq i}}^{N_H} \frac{\partial \varphi_{H,ij}}{\partial r_{ij}} \right) \frac{r_k r_l}{r_{ij}} \tag{52a}$$

$$B^{I}_{\text{ijkl}} = \frac{1}{2} \left[\sum_{\substack{j=1 \\ j\neq i}}^{N_I} \frac{\partial^2 \varphi_{I-I,ij}}{\partial r_{ij}^2} - \frac{1}{r_{ij}} \sum_{\substack{j=1 \\ j\neq i}}^{N_I} \frac{\partial \varphi_{I-I,ij}}{\partial r_{ij}} \right] \frac{r_i r_j r_k r_l}{r_{ij}^2} + \frac{1}{2} \delta_{ik} \left(\sum_{\substack{j=1 \\ j\neq i}}^{N_I} \frac{\partial \varphi_{I-I,ij}}{\partial r_{ij}} \right) \frac{r_k r_l}{r_{ij}} \tag{52b}$$

$$V^{H-H}{}_{ij} = \left(\frac{\partial \rho_{H\text{-}Hj}}{\partial r_{ij}} \right) \frac{r_i r_j}{r_{ij}}, \qquad V^{I-I}{}_{ij} = \left(\frac{\partial \rho_{I\text{-}Ij}}{\partial r_{ij}} \right) \frac{r_i r_j}{r_{ij}} \tag{53a}$$

$$V^{H-H}{}_{kl} = \left(\frac{\partial \rho_{H\text{-}Hj}}{\partial r_{ij}} \right) \frac{r_k r_l}{r_{ij}}, \qquad V^{I-I}{}_{kl} = \left(\frac{\partial \rho_{I\text{-}Ij}}{\partial r_{ij}} \right) \frac{r_k r_l}{r_{ij}} \tag{53b}$$

VACANCY FORMATION ENERGY

$$E^1_{UV} = -\frac{1}{2} \sum_m \phi(\mathrm{r}_m) - \sum_m F(\bar{\rho}) + \sum_m F(\bar{\rho} - f(r_m)) \tag{54}$$

$$E^1_{V} = -\frac{1}{2} \sum_m \phi(\mathrm{r}_m) - \sum_m F(\bar{\rho}) + \sum_m F(\bar{\rho} - f(r_m)) + E_{\text{relax}} \tag{55}$$

$$E^2_{UV} = -2E^1_V + \phi(r_e) - \sum_m \phi(\mathrm{r}_m) - \sum_m F(\bar{\rho}) + \sum_{m>2} F(\bar{\rho} - 2f(r_m)) \tag{56}$$

GIBBS FREE ENERGY MIXING

$$\Delta \mathrm{G}^{\text{mix}} = \Delta \mathrm{H}^{\text{mix}} - \Delta \mathrm{S}^{\text{mix}} \cdot \mathrm{T} \tag{57}$$

$$\left[\mathrm{N}_{\text{Pd}} \cdot x \right] PdH + \left[N_{Pd} \cdot (1-x) \right] Pd \rightarrow \left[N_{Pd} \right] PdH_x \tag{58}$$

$$\Delta E = N_{\text{Pd}} (1+x) \cdot E_{PdH_x} - 2 \cdot N_{Pd} \cdot x \cdot E_{PdH} - N_{Pd} \cdot (1-x) E_{Pd} \tag{59}$$

$$\Delta E = \left[(1+x) \cdot E_{PdH_x} \text{-} 2 \cdot \text{x} \cdot E_{PdH} - (1-x) E_{Pd} \right] / (1-x) \tag{60}$$

Replacing x with mol fraction $X=x/(1+x)$, obtain the heat of mixing

$$\Delta \text{H}^{\text{mix}} = E_{PdHx} - 2X \cdot E_{PdH} - (1-2X) \cdot E_{Pd} \tag{61}$$

$$\Delta S_t = -N_{Pd} \cdot k_B \cdot \left[x \cdot \ln(x) + (1-x) \cdot \ln(x) \right] \tag{62}$$

$$\Delta S = -k_B \cdot \left[x \cdot \ln(x) + (1-x) \cdot \ln(1\text{-}x) \right] / (1+x) \tag{63}$$

$$\Delta S_t = -k_B \cdot \left[X \cdot \ln\left[X / (1-X) \right] + (1-2 \cdot X) \cdot \ln\left[(1-2X) / (1-X) \right] \right] \tag{64}$$

APPENDIX C:

CODE USED TO CALCULATE DATA IN THIS THESIS

SIESTA CODE

$Pd_{0.75}Cu_{0.25}H_{1.00}$.fdf

```
# $Id: Fe.fdf,v 1.1 1999/04/20 12:52:43 emilio Exp $
# ---------------------------------------------------------------------
# FDF for bcc iron
# GGA, Ferromagnetic.
# Scalar-relativistic pseudopotential with non-linear partial-core
  correction
# E. Artacho, April 1999
# ---------------------------------------------------------------------

SystemName      fcc PdCuH LDA   # Descriptive name of the system
SystemLabel        PdCuH        # Short name for naming files

# Output options

WriteCoorStep
WriteMullikenPop       1

# Species and atoms

NumberOfSpecies        3
NumberOfAtoms          8
%block ChemicalSpeciesLabel
 1  46  Pd
 2  29  Cu
 3   1   H
%endblock ChemicalSpeciesLabel

# Basis

PAO.EnergyShift       50 meV
PAO.BasisSize         DZP

LatticeConstant        4.22 Ang

%block LatticeVectors
 1.00000   0.00000  0.00000
 0.00000   1.00000  0.00000
 0.00000   0.00000  1.000000
%endblock LatticeVectors

#KgridCutoff          15. Ang

%block kgrid_Monkhorst_Pack
```

```
 18  0  0  0.5
  0 18  0  0.5
  0  0 18  0.5
%endblock kgrid_Monkhorst_Pack

#%block BandLines
#  1  0.00000   0.000000  0.000000  \Gamma
# 40  2.00000   0.000000  0.000000  H
# 28  1.00000   1.000000  0.000000  N
# 28  0.00000   0.000000  0.000000  \Gamma
# 34  1.00000   1.000000  1.000000  P
#%endblock BandLines

xc.functional         LDA           # Exchange-correlation functional
xc.authors            ca            # Exchange-correlation version

SpinPolarized         true          # Logical parameters are: yes or
no

#MeshCutoff           150. Ry       # Mesh cutoff. real space mesh

# SCF options
MaxSCFIterations      200           # Maximum number of SCF iter
DM.MixingWeight       0.1           # New DM amount for next SCF cycle
DM.Tolerance          1.d-3         # Tolerance in maximum difference
                                    # between input and output DM
DM.UseSaveDM          true          # to use continuation files
DM.NumberPulay         3

SolutionMethod        diagon        # OrderN or Diagon
ElectronicTemperature  25 meV       # Temp. for Fermi smearing

# MD options

MD.TypeOfRun           cg             # Type of dynamics:
MD.NumCGsteps          100            # Number of CG steps for
MD.VariableCell        true           #   coordinate optimization
%block GeometryConstraints
rigid 1 2 3 4 5 6 7 8
cellangle alpha beta gamma
%endblock GeometryConstraints
MD.MaxCGDispl          0.2 Ang        # Maximum atomic displacement
                                      #   in one CG step (Bohr)
MD.MaxForceTol         0.0005 eV/Ang  # Tolerance in the maximum
                                      #   atomic force (Ry/Bohr)

# Atomic coordinates
AtomicCoordinatesFormat     Fractional
%block AtomicCoordinatesAndAtomicSpecies
 0.0   0.0   0.0   2  # Ag
 0.5   0.5   0.0   1  # Pd
 0.0   0.5   0.5   1  # Pd
```

```
 0.5   0.0   0.5   1  # Pd
 0.5   0.5   0.5   3  # H
 0.5   0.0   0.0   3  # H
 0.0   0.5   0.0   3  # H
 0.0   0.0   0.5   3  # H
%endblock AtomicCoordinatesAndAtomicSpecies
```

LAMMPS CODE

ELASTIC CONSTANTS CALCULATION [49]

By running one simulation in LAMMPS, the lattice constants, cohesive energies, and elastic constants can be obtained. The in.elastic file uses additional files in the same directory for certain functions like designating the structure, EAM potential, etc.

in.elastic_PdAgCuH0666

```
# Compute elastic constant tensor for a crystal
#
# Written by Aidan Thompson (Sandia, athomps@sandia.gov)
#
#  This script uses the following three include files.
#
#   init.mod     (must be modified for different crystal structures)
#                Define units, deformation parameters and initial
#           configuration of the atoms and simulation cell.
#
#
#   potential.mod    (must be modified for different pair styles)
#               Define pair style and other attributes
#               not stored in restart file
#
#
#   displace.mod    (displace.mod should not need to be modified)
#              Perform positive and negative box displacements
#              in direction ${dir} and size ${up}.
#              It uses the resultant changes
#              in stress to compute one
#              row of the elastic stiffness tensor
#
#              Inputs variables:
#                   dir = the Voigt deformation component
#                         (1,2,3,4,5,6)
#              Global constants:
#                   up = the deformation magnitude (strain units)
#                   cfac = conversion from LAMMPS pressure units to
```

```
#                   output units for elastic constants
#
#
#  To run this on a different system, it should only be necessary to
#  modify the files init.mod and potential.mod. In order to calculate
#  the elastic constants correctly, care must be taken to specify
#  the correct units in init.mod (units, cfac and cunits). It is also
#  important to verify that the minimization of energy w.r.t atom
#  positions in the deformed cell is fully converged.
#  One indication of this is that the elastic constants are
insensitive
#  to the choice of the variable ${up} in init.mod. Another is to
check
#  the final max and two-norm forces reported in the log file. If you
know
#  that minimization is not required, you can set maxiter = 0.0 in
#  init.mod.
#
#  There are two alternate versions of displace.mod provided.
#  They are displace_restart.mod and displace_reverse.mod.
#  The former resets the box using a restart file while
#  the latter reverses the deformation. Copy whichever
#  one you like best to displace.mod.
#

include init.mod-PdAgCuH-0.666
include potential.mod
include NPT.mod
# Compute initial state
fix 3 all box/relax iso 0.0
minimize ${etol} ${ftol} ${maxiter} ${maxeval}

variable tmp equal pxx
variable pxx0 equal ${tmp}
variable tmp equal pyy
variable pyy0 equal ${tmp}
variable tmp equal pzz
variable pzz0 equal ${tmp}
variable tmp equal pyz
variable pyz0 equal ${tmp}
variable tmp equal pxz
variable pxz0 equal ${tmp}
variable tmp equal pxy
variable pxy0 equal ${tmp}

variable tmp equal lx
variable lx0 equal ${tmp}
variable tmp equal ly
variable ly0 equal ${tmp}
variable tmp equal lz
variable lz0 equal ${tmp}
```

```
# These formulas define the derivatives w.r.t. strain components
# Constants uses $, variables use v_
variable d1 equal -(v_pxx1-${pxx0})/(v_delta/v_len0)*${cfac}
variable d2 equal -(v_pyy1-${pyy0})/(v_delta/v_len0)*${cfac}
variable d3 equal -(v_pzz1-${pzz0})/(v_delta/v_len0)*${cfac}
variable d4 equal -(v_pyz1-${pyz0})/(v_delta/v_len0)*${cfac}
variable d5 equal -(v_pxz1-${pxz0})/(v_delta/v_len0)*${cfac}
variable d6 equal -(v_pxy1-${pxy0})/(v_delta/v_len0)*${cfac}

# Write restart
unfix 3
write_restart restart.equil

# uxx Perturbation

variable dir equal 1
include displace.mod

# uyy Perturbation

variable dir equal 2
include displace.mod

# uzz Perturbation

variable dir equal 3
include displace.mod

# uyz Perturbation

variable dir equal 4
include displace.mod

# uxz Perturbation

variable dir equal 5
include displace.mod

# uxy Perturbation

variable dir equal 6
include displace.mod

# Output final values

variable C11all equal ${C11}
variable C22all equal ${C22}
variable C33all equal ${C33}

variable C12all equal 0.5*(${C12}+${C21})
variable C13all equal 0.5*(${C13}+${C31})
variable C23all equal 0.5*(${C23}+${C32})
```

```
variable C44all equal ${C44}
variable C55all equal ${C55}
variable C66all equal ${C66}

variable C14all equal 0.5*(${C14}+${C41})
variable C15all equal 0.5*(${C15}+${C51})
variable C16all equal 0.5*(${C16}+${C61})

variable C24all equal 0.5*(${C24}+${C42})
variable C25all equal 0.5*(${C25}+${C52})
variable C26all equal 0.5*(${C26}+${C62})

variable C34all equal 0.5*(${C34}+${C43})
variable C35all equal 0.5*(${C35}+${C53})
variable C36all equal 0.5*(${C36}+${C63})

variable C45all equal 0.5*(${C45}+${C54})
variable C46all equal 0.5*(${C46}+${C64})
variable C56all equal 0.5*(${C56}+${C65})
variable Bm     equal (1/3)*(${C11all}+2*(${C12all}))
variable Cpr    equal 0.5*(${C11all}-(${C12all}))

# For Stillinger-Weber silicon, the analytical results
# are known to be (E. R. Cowley, 1988):
#               C11 = 151.4 GPa
#               C12 = 76.4 GPa
#               C44 = 56.4 GPa

print "Elastic Constant C11all = ${C11all} ${cunits}"
print "Elastic Constant C22all = ${C22all} ${cunits}"
print "Elastic Constant C33all = ${C33all} ${cunits}"

print "Elastic Constant C12all = ${C12all} ${cunits}"
print "Elastic Constant C13all = ${C13all} ${cunits}"
print "Elastic Constant C23all = ${C23all} ${cunits}"

print "Elastic Constant C44all = ${C44all} ${cunits}"
print "Elastic Constant C55all = ${C55all} ${cunits}"
print "Elastic Constant C66all = ${C66all} ${cunits}"

print "Elastic Constant C14all = ${C14all} ${cunits}"
print "Elastic Constant C15all = ${C15all} ${cunits}"
print "Elastic Constant C16all = ${C16all} ${cunits}"

print "Elastic Constant C24all = ${C24all} ${cunits}"
print "Elastic Constant C25all = ${C25all} ${cunits}"
print "Elastic Constant C26all = ${C26all} ${cunits}"

print "Elastic Constant C34all = ${C34all} ${cunits}"
print "Elastic Constant C35all = ${C35all} ${cunits}"
print "Elastic Constant C36all = ${C36all} ${cunits}"
```

```
print "Elastic Constant C45all = ${C45all} ${cunits}"
print "Elastic Constant C46all = ${C46all} ${cunits}"
print "Elastic Constant C56all = ${C56all} ${cunits}"

variable natoms equal "count(all)"
variable xlength equal "lx/4"
variable ecoh equal "pe/v_natoms"

print ""
print "Number of atoms = ${natoms};"
print "Lattice constant in x (Angstoms) = ${xlength};"
print "Cohesive energy (eV) = ${ecoh};"
print "Elastic Constant C11all = ${C11all} ${cunits}"
print "Elastic Constant C12all = ${C12all} ${cunits}"
print "Elastic Constant C44all = ${C44all} ${cunits}"
print "Elastic Constant Cprall = ${Cpr}    ${cunits}"
print "Bulk Modulus BMall      = ${Bm}     ${cunits}"


print "${xlength}     ${ecoh}    ${C11all}  ${C12all}  ${C44all}

      ${Cpr}     ${Bm}" file MD_results_PdAgCuH0666.dat
```

init.mod-PdAgCuH-0.666

```
# NOTE: This script can be modified for different atomic structures,
# units, etc. See in.elastic for more info.
#

# Define the finite deformation size. Try several values of this
# variable to verify that results do not depend on it.
variable up equal 1.0e-6

# Uncomment one of these blocks, depending on what units
# you are using in LAMMPS and for output

# metal units, elastic constants in eV/A^3
#units          metal
#variable cfac equal 6.2414e-7
#variable cunits string eV/A^3

# metal units, elastic constants in GPa
units      metal
variable cfac equal 1.0e-4
variable cunits string GPa

# real units, elastic constants in GPa
#units          real
```

```
#variable cfac equal 1.01325e-4
#variable cunits string GPa

# Define minimization parameters
variable etol equal 0.0
variable ftol equal 1.0e-10
variable maxiter equal 10000
variable maxeval equal 1000000
variable dmax equal 1.0e-6

# generate the box and atom positions using a diamond lattice
variable a equal 3.89
boundary   p p p
lattice         custom $a a1 1.0 0.0 0.0 a2 0.0 1.0 0.0 a3 0.0 0.0 1.0
basis 0.0 0.0 0.0 basis 0.5 0.5 0.0 basis 0.5 0.0 0.5 basis 0.0 0.5
0.5 &
basis 0.5 0.0 0.0 basis 0.0 0.5 0.0 basis 0.0 0.0 0.5 basis 0.5 0.5
0.5
#lattice        fcc $a
region          box prism 0 4.0 0 4.0 0 4.0 0 0 0
create_box 4 box
create_atoms    1 region box basis 1 2 basis 2 3 basis 5 4 basis 6 4
basis 7 4 basis 8 4

group           group1 id 5:509:8   # PdH0.784
group           group2 id 7:175:8

delete_atoms    group  group1
delete_atoms   group  group2

# Need to set mass to something, just to satisfy LAMMPS
mass 1 1.0e-20
```

potential.mod

```
# NOTE: This script can be modified for different pair styles
# See in.elastic for more info.

#include  init.mod

# Choose potential
pair_style     eam/alloy
pair_coeff     * * PdAgCuH.eam.alloy Pd Ag Cu H

# Setup neighbor style
neighbor 1.0 nsq
neigh_modify once no every 1 delay 0 check yes

# Setup minimization style
```

```
min_style       cg
min_modify      dmax ${dmax} line quadratic

dump        1  all xyz 1000 PdAgCuH.xyz
dump        2  all custom 1000 PdAgCuH_Custom.xyz type id x y z
dump_modify 2  sort  1

# Setup output
thermo          1000

#  pxx,pyy,pzz,pxy,pxz,pyz = 6 components of pressure tensor

thermo_style custom step temp pe press pxx pyy pzz pxy pxz pyz lx ly
lz vol
thermo_modify norm no
```

NPT.mod

```
variable                t equal 1000
variable                p equal 0
velocity                all create $t 4928459
#fix                    1 all npt temp $t $t 1 iso $p $p 100

fix                     1 all npt temp $t 1.0 10 iso 0.0 0.0 100
fix                     2 all temp/rescale 1 $t 1.0 0.01 1.0

run                     200000
unfix                   1
unfix                   2
```

MATLAB PROGRAM FOR FITTING THE MODEL PARAMETERS

The fitting procedure was run from the MATLAB file Fit.m The objective function used in optimization was in a separate file objfunc.m Another file, Check_Fit.m, was used to generate plots and review the quality of the fit with the parameters obtained.

Fit.m

```
function EAM_Parameters_Fitting

    warning off
    % Clear command window
    clc
```

```
global a_PdCuH025_o1;
global a_PdCuH025_o2;
global a_PdCuH025_T;
global a_PdCuH050_O1_O2;
global a_PdCuH050_O2_O2;
global a_PdCuH050_T;
global a_PdCuH075_O1_O2_O2;
global a_PdCuH075_O2_O2_O2;
global a_PdCuH075_T;
global a_PdCuH100_O1_O2_O2_O2;
global a_PdCuH100_T;

global Ec_exp_PdCuH000;
global Ec_exp_PdCuH025_O1;
global Ec_exp_PdCuH025_O2;
global Ec_exp_PdCuH025_T;
global Ec_exp_PdCuH050_O1_O2;
global Ec_exp_PdCuH050_O2_O2;
global Ec_exp_PdCuH050_T;
global Ec_exp_PdCuH075_O1_O2_O2;
global Ec_exp_PdCuH075_O2_O2_O2;
global Ec_exp_PdCuH075_T;
global Ec_exp_PdCuH100_O1_O2_O2_O2;
global Ec_exp_PdCuH100_T;

global a_PdCuH025_TE1
global a_PdCuH050_TE3_TE4
global a_PdCuH050_TE1_TE8
global a_PdCuH075_TE1_TE2_TE3
global a_PdCuH075_TE1_TE2_TE8
global a_PdCuH075_TE5_TE6_TE7
global a_PdCuH100_TE1_TE2_TE3_TE4
global a_PdCuH100_TE1_TE2_TE7_TE8
global a_PdCuH100_TE1_TE5_TE6_TE7

global Ec_exp_PdCuH025_TE1
global Ec_exp_PdCuH050_TE3_TE4
global Ec_exp_PdCuH050_TE1_TE8
global Ec_exp_PdCuH075_TE1_TE2_TE3
global Ec_exp_PdCuH075_TE1_TE2_TE8
global Ec_exp_PdCuH075_TE5_TE6_TE7
global Ec_exp_PdCuH100_TE1_TE2_TE3_TE4
global Ec_exp_PdCuH100_TE1_TE2_TE7_TE8
global Ec_exp_PdCuH100_TE1_TE5_TE6_TE7


a_PdCuH025_o1 = 3.8880;
```

```
a_PdCuH025_o2 = 3.8785;
a_PdCuH025_T = 3.9062;
a_PdCuH050_O1_O2 = 3.9429;
a_PdCuH050_O2_O2 = 3.9347;
a_PdCuH050_T = 3.9858;
a_PdCuH075_O1_O2_O2 = 3.9871;
a_PdCuH075_O2_O2_O2 = 3.9761;
a_PdCuH075_T = 4.0602;
a_PdCuH100_O1_O2_O2_O2 = 4.0366;
a_PdCuH100_T = 4.1399;

Ec_exp_PdCuH000 = -3.8775;
Ec_exp_PdCuH025_O1 = -3.6198;
Ec_exp_PdCuH025_O2 = -3.5912;
Ec_exp_PdCuH025_T = -3.5748;
Ec_exp_PdCuH050_O1_O2 = -3.4351;
Ec_exp_PdCuH050_O2_O2 = -3.3999;
Ec_exp_PdCuH050_T = -3.3899;
Ec_exp_PdCuH075_O1_O2_O2 = -3.2973;
Ec_exp_PdCuH075_O2_O2_O2 = -3.2617;
Ec_exp_PdCuH075_T = -3.2644;
Ec_exp_PdCuH100_O1_O2_O2_O2 = -3.1773;
Ec_exp_PdCuH100_T = -3.1475;

a_PdCuH025_TE1 = 3.9062;
a_PdCuH050_TE3_TE4 = 3.9857;
a_PdCuH050_TE1_TE8 = 4.0036;
a_PdCuH075_TE1_TE2_TE3 = 4.0417;
a_PdCuH075_TE1_TE2_TE8 = 4.0677;
a_PdCuH075_TE5_TE6_TE7 = 4.2134;
a_PdCuH100_TE1_TE2_TE3_TE4 = 4.1226;
a_PdCuH100_TE1_TE2_TE7_TE8 = 4.1330;
a_PdCuH100_TE1_TE5_TE6_TE7 = 4.3011;

Ec_exp_PdCuH025_TE1  = -3.5748;
Ec_exp_PdCuH050_TE3_TE4 = -3.3899;
Ec_exp_PdCuH050_TE1_TE8 = -3.3828;
Ec_exp_PdCuH075_TE1_TE2_TE3 = -3.2315;
Ec_exp_PdCuH075_TE1_TE2_TE8 = -3.2529;
Ec_exp_PdCuH075_TE5_TE6_TE7 = -3.1415;
Ec_exp_PdCuH100_TE1_TE2_TE3_TE4 = -3.1105;
Ec_exp_PdCuH100_TE1_TE2_TE7_TE8 = -3.1101;
Ec_exp_PdCuH100_TE1_TE5_TE6_TE7 = -3.0388;

global ri_PdCuH000;
global index_PdCuH000_Pd
global index_PdCuH000_Cu
```

```
global ri_PdCuH025_o1;
global index_PdCuH025_o1_Pd
global index_PdCuH025_o1_Cu
global index_PdCuH025_o1_H

global ri_PdCuH025_o2;
global index_PdCuH025_o2_Pd
global index_PdCuH025_o2_Cu
global index_PdCuH025_o2_H

global ri_PdCuH025_T_25_25_25;
global index_PdCuH025_T_Pd_25_25_25
global index_PdCuH025_T_Cu_25_25_25
global index_PdCuH025_T_H_25_25_25

global ri_PdCuH025_T_75_75_75;
global index_PdCuH025_T_Pd_75_75_75
global index_PdCuH025_T_Cu_75_75_75
global index_PdCuH025_T_H_75_75_75

global ri_PdCuH025_T_25_75_75;
global index_PdCuH025_T_Pd_25_75_75
global index_PdCuH025_T_Cu_25_75_75
global index_PdCuH025_T_H_25_75_75

global ri_PdCuH050_O1_O2;
global index_PdCuH050_O1_O2_Pd
global index_PdCuH050_O1_O2_Cu
global index_PdCuH050_O1_O2_H_O1
global index_PdCuH050_O1_O2_H_O2

global ri_PdCuH050_O2_O2;
global index_PdCuH050_O2_O2_Pd
global index_PdCuH050_O2_O2_Cu
global index_PdCuH050_O2_O2_H

global ri_PdCuH075_O1_O2_O2;
global index_PdCuH075_O1_O2_O2_Pd
global index_PdCuH075_O1_O2_O2_Cu
global index_PdCuH075_O1_O2_O2_H_O1
global index_PdCuH075_O1_O2_O2_H_O2

global ri_PdCuH075_O2_O2_O2;
global index_PdCuH075_O2_O2_O2_Pd
global index_PdCuH075_O2_O2_O2_Cu
global index_PdCuH075_O2_O2_O2_H
```

```
global ri_PdCuH075_T;
global index_PdCuH075_T_Pd
global index_PdCuH075_T_Cu
global index_PdCuH075_T_H

global ri_PdCuH100_O1_O2_O2_O2;
global index_PdCuH100_O1_O2_O2_O2_Pd
global index_PdCuH100_O1_O2_O2_O2_Cu
global index_PdCuH100_O1_O2_O2_O2_H_O1
global index_PdCuH100_O1_O2_O2_O2_H_O2

global ri_PdCuH100_T;
global index_PdCuH100_T_Pd
global index_PdCuH100_T_Cu
global index_PdCuH100_T_H

global rc_1
global rc_2
global rc_3

global NP
global NP2
global NP3

global x_Pd
global x_Cu

global S_Cu
global S_Pd

global x_PdH

% Pd and Cu fitting parameters
x_Pd = [2.054085    0.216817    8.414105    7.221224
0.999999    3.316887];
x_Cu = [2.5045   0.175425   8.713725    6.906629   0.560027
3.648665];

% % %     re fit scaling only mixing rule potential-Pd-Cu
S_Pd = 1.7097;
S_Cu = 1.3419;

x_PdH = [0.589510   1.104827    0.942490    2.145808
0.942201...
    0.740938    2.373944    1.702142    8.370790
62.343273...
```

```
    0.000100    1.187000    1.300000    3.474173];

NP = 0;
NP2 = 0;
NP3 = 0;

rc_1 = 5.35;
rc_2 = 5.35;
rc_3 = 5.35;


global ri_PdH100_OC;
global index_PdH100_OC_1;
load PdH100_OC_SORTED.data;
ri_PdH100_OC = PdH100_OC_SORTED(:,:);
ra_OC = [1 0 0 0];
index_PdH100_OC_1 =
find(ismember(ri_PdH100_OC,ra_OC,'rows'));


% Position Code For Tetrahedral H Atoms
% TE1   0.025   0.025   0.025
% TE2   0.075   0.025   0.025
% TE3   0.025   0.075   0.025
% TE4   0.025   0.025   0.075
% TE5   0.075   0.075   0.025
% TE6   0.025   0.075   0.075
% TE7   0.075   0.025   0.075
% TE8   0.075   0.075   0.075


load PdCuH000_sorted_6x6x6.data;
load PdCuH025_O1_sorted_6x6x6.dat;
load PdCuH025_O2_sorted_6x6x6.dat;
load PdCuH025_T_sorted_6x6x6_25_25_25.dat;
load PdCuH025_T_sorted_6x6x6_75_75_75.dat;
load PdCuH025_T_sorted_6x6x6_25_75_75.dat;
load PdCuH050_O1_O2_sorted_6x6x6.dat;
load PdCuH050_O2_O2_sorted_6x6x6.dat;
load PdCuH050_T_sorted_6x6x6_252525_752525.dat;
load PdCuH050_T_sorted_6x6x6_252525_757575.dat;
load PdCuH075_O1_O2_O2_sorted_6x6x6.dat;
load PdCuH075_O2_O2_O2_sorted_6x6x6.dat;
load PdCuH075_T_sorted_6x6x6.dat;
load PdCuH100_O1_O2_O2_O2_sorted_6x6x6.dat;
load PdCuH100_T_sorted_6x6x6.dat;
```

```
% Atomic positions for structure
ri_PdCuH000 = PdCuH000_sorted_6x6x6(:,:);
ri_PdCuH025_o1 = PdCuH025_O1_sorted_6x6x6(:,:);
ri_PdCuH025_o2 = PdCuH025_O2_sorted_6x6x6(:,:);
ri_PdCuH025_T_25_25_25 =
PdCuH025_T_sorted_6x6x6_25_25_25(:,:);
ri_PdCuH025_T_75_75_75 =
PdCuH025_T_sorted_6x6x6_75_75_75(:,:);
ri_PdCuH025_T_25_75_75 =
PdCuH025_T_sorted_6x6x6_25_75_75(:,:);
ri_PdCuH050_O1_O2 = PdCuH050_O1_O2_sorted_6x6x6(:,:);
ri_PdCuH050_O2_O2 = PdCuH050_O2_O2_sorted_6x6x6(:,:);
ri_PdCuH075_O1_O2_O2 = PdCuH075_O1_O2_O2_sorted_6x6x6(:,:);
ri_PdCuH075_O2_O2_O2 = PdCuH075_O2_O2_O2_sorted_6x6x6(:,:);
ri_PdCuH075_T = PdCuH075_T_sorted_6x6x6(:,:);
ri_PdCuH100_O1_O2_O2_O2 =
PdCuH100_O1_O2_O2_O2_sorted_6x6x6(:,:);
ri_PdCuH100_T = PdCuH100_T_sorted_6x6x6(:,:);


% Indices for OC central atoms
ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
index_PdCuH000_Pd = find(ismember(ri_PdCuH000,ra_1,'rows'));
index_PdCuH000_Cu = find(ismember(ri_PdCuH000,ra_2,'rows'));

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 1 1 1];   % body center H
index_PdCuH025_o1_Pd =
find(ismember(ri_PdCuH025_o1,ra_1,'rows'));
index_PdCuH025_o1_Cu =
find(ismember(ri_PdCuH025_o1,ra_2,'rows'));
index_PdCuH025_o1_H =
find(ismember(ri_PdCuH025_o1,ra_3,'rows'));

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 1 0 0];   % edge H
index_PdCuH025_o2_Pd =
find(ismember(ri_PdCuH025_o2,ra_1,'rows'));
index_PdCuH025_o2_Cu =
find(ismember(ri_PdCuH025_o2,ra_2,'rows'));
index_PdCuH025_o2_H =
find(ismember(ri_PdCuH025_o2,ra_3,'rows'));

ra_1 = [1 1 1 0];   % face center Pd
```

```
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 .5 .5 .5];   % TE H
index_PdCuH025_T_Pd_25_25_25 =
find(ismember(ri_PdCuH025_T_25_25_25,ra_1,'rows'));
index_PdCuH025_T_Cu_25_25_25 =
find(ismember(ri_PdCuH025_T_25_25_25,ra_2,'rows'));
index_PdCuH025_T_H_25_25_25 =
find(ismember(ri_PdCuH025_T_25_25_25,ra_3,'rows'));

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 1.5 1.5 1.5];   % TE H
index_PdCuH025_T_Pd_75_75_75 =
find(ismember(ri_PdCuH025_T_75_75_75,ra_1,'rows'));
index_PdCuH025_T_Cu_75_75_75 =
find(ismember(ri_PdCuH025_T_75_75_75,ra_2,'rows'));
index_PdCuH025_T_H_75_75_75 =
find(ismember(ri_PdCuH025_T_75_75_75,ra_3,'rows'));

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 .5 1.5 1.5];   % TE H
index_PdCuH025_T_Pd_25_75_75 =
find(ismember(ri_PdCuH025_T_25_75_75,ra_1,'rows'));
index_PdCuH025_T_Cu_25_75_75 =
find(ismember(ri_PdCuH025_T_25_75_75,ra_2,'rows'));
index_PdCuH025_T_H_25_75_75 =
find(ismember(ri_PdCuH025_T_25_75_75,ra_3,'rows'));

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 1 1 1];   % body center H
ra_4 = [3 1 0 0];   % edge H
index_PdCuH050_O1_O2_Pd =
find(ismember(ri_PdCuH050_O1_O2,ra_1,'rows'));
index_PdCuH050_O1_O2_Cu =
find(ismember(ri_PdCuH050_O1_O2,ra_2,'rows'));
index_PdCuH050_O1_O2_H_O1 =
find(ismember(ri_PdCuH050_O1_O2,ra_3,'rows'));
index_PdCuH050_O1_O2_H_O2 =
find(ismember(ri_PdCuH050_O1_O2,ra_4,'rows'));

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 1 0 0];   % edge H
index_PdCuH050_O2_O2_Pd =
find(ismember(ri_PdCuH050_O2_O2,ra_1,'rows'));
```

```
index_PdCuH050_O2_O2_Cu =
find(ismember(ri_PdCuH050_O2_O2,ra_2,'rows'));
index_PdCuH050_O2_O2_H =
find(ismember(ri_PdCuH050_O2_O2,ra_3,'rows'));

ri_PdCuH050_TE1_TE2 =
PdCuH050_T_sorted_6x6x6_252525_752525(:,:);

global ri_PdCuH050_TE1_TE2;
global index_PdCuH050_TE1_TE2_Pd
global index_PdCuH050_TE1_TE2_Cu
global index_PdCuH050_TE1_TE2_H_TE1
global index_PdCuH050_TE1_TE2_H_TE2

ra_1 = [1 1 1 0];    % face center Pd
ra_2 = [2 0 0 0];    % corner Cu
ra_3 = [3 .5 .5 .5];    % TE H
ra_4 = [3 1.5 .5 .5];    % TE H
index_PdCuH050_TE1_TE2_Pd =
find(ismember(ri_PdCuH050_TE1_TE2,ra_1,'rows'));
index_PdCuH050_TE1_TE2_Cu =
find(ismember(ri_PdCuH050_TE1_TE2,ra_2,'rows'));
index_PdCuH050_TE1_TE2_H_TE1 =
find(ismember(ri_PdCuH050_TE1_TE2,ra_3,'rows'));
index_PdCuH050_TE1_TE2_H_TE2 =
find(ismember(ri_PdCuH050_TE1_TE2,ra_4,'rows'));

ri_PdCuH050_TE1_TE8 =
PdCuH050_T_sorted_6x6x6_252525_757575(:,:);

global ri_PdCuH050_TE1_TE8;
global index_PdCuH050_TE1_TE8_Pd
global index_PdCuH050_TE1_TE8_Cu
global index_PdCuH050_TE1_TE8_H_TE1
global index_PdCuH050_TE1_TE8_H_TE8

ra_1 = [1 1 1 0];    % face center Pd
ra_2 = [2 0 0 0];    % corner Cu
ra_3 = [3 .5 .5 .5];    % TE H
ra_4 = [3 1.5 1.5 1.5];    % TE H
index_PdCuH050_TE1_TE8_Pd =
find(ismember(ri_PdCuH050_TE1_TE8,ra_1,'rows'));
index_PdCuH050_TE1_TE8_Cu =
find(ismember(ri_PdCuH050_TE1_TE8,ra_2,'rows'));
index_PdCuH050_TE1_TE8_H_TE1 =
find(ismember(ri_PdCuH050_TE1_TE8,ra_3,'rows'));
```

```
index_PdCuH050_TE1_TE8_H_TE8 =
find(ismember(ri_PdCuH050_TE1_TE8,ra_4,'rows'));

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 1 1 1];   % body center H
ra_4 = [3 1 0 0];   % edge H
index_PdCuH075_O1_O2_O2_Pd =
find(ismember(ri_PdCuH075_O1_O2_O2,ra_1,'rows'));
index_PdCuH075_O1_O2_O2_Cu =
find(ismember(ri_PdCuH075_O1_O2_O2,ra_2,'rows'));
index_PdCuH075_O1_O2_O2_H_O1 =
find(ismember(ri_PdCuH075_O1_O2_O2,ra_3,'rows'));
index_PdCuH075_O1_O2_O2_H_O2 =
find(ismember(ri_PdCuH075_O1_O2_O2,ra_4,'rows'));

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 1 0 0];   % edge H
% ra_3 = [3 0 0 1];   % edge H
index_PdCuH075_O2_O2_O2_Pd =
find(ismember(ri_PdCuH075_O2_O2_O2,ra_1,'rows'));
index_PdCuH075_O2_O2_O2_Cu =
find(ismember(ri_PdCuH075_O2_O2_O2,ra_2,'rows'));
index_PdCuH075_O2_O2_O2_H =
find(ismember(ri_PdCuH075_O2_O2_O2,ra_3,'rows'));

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 .5 .5 .5];   % TE H
% ra_4 = [3 1.5 .5 .5];   % TE H
index_PdCuH075_T_Pd =
find(ismember(ri_PdCuH075_T,ra_1,'rows'));
index_PdCuH075_T_Cu =
find(ismember(ri_PdCuH075_T,ra_2,'rows'));
index_PdCuH075_T_H =
find(ismember(ri_PdCuH075_T,ra_3,'rows'));

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 1 1 1];   % body center H
ra_4 = [3 1 0 0];   % edge H
index_PdCuH100_O1_O2_O2_O2_Pd =
find(ismember(ri_PdCuH100_O1_O2_O2_O2,ra_1,'rows'));
index_PdCuH100_O1_O2_O2_O2_Cu =
find(ismember(ri_PdCuH100_O1_O2_O2_O2,ra_2,'rows'));
```

```
index_PdCuH100_O1_O2_O2_O2_H_O1 =
find(ismember(ri_PdCuH100_O1_O2_O2_O2,ra_3,'rows'));
index_PdCuH100_O1_O2_O2_O2_H_O2 =
find(ismember(ri_PdCuH100_O1_O2_O2_O2,ra_4,'rows'));

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 .5 .5 .5];   % TE H
index_PdCuH100_T_Pd =
find(ismember(ri_PdCuH100_T,ra_1,'rows'));
index_PdCuH100_T_Cu =
find(ismember(ri_PdCuH100_T,ra_2,'rows'));
index_PdCuH100_T_H =
find(ismember(ri_PdCuH100_T,ra_3,'rows'));

load PdCuH050_T_sorted_6x6x6_TE3_TE4.dat;
load PdCuH050_T_sorted_6x6x6_TE3_TE6.dat;
load PdCuH075_T_sorted_6x6x6_TE1_TE2_TE3.dat;
load PdCuH075_T_sorted_6x6x6_TE1_TE2_TE8.dat;
load PdCuH100_T_sorted_6x6x6_TE1_TE2_TE3_TE4.dat;
load PdCuH100_T_sorted_6x6x6_TE1_TE2_TE7_TE8.dat;

global ri_PdCuH050_TE3_TE4
global index_PdCuH050_TE3_TE4_Pd
global index_PdCuH050_TE3_TE4_Cu
global index_PdCuH050_TE3_TE4_H_TE3
global index_PdCuH050_TE3_TE4_H_TE4

ri_PdCuH050_TE3_TE4 = PdCuH050_T_sorted_6x6x6_TE3_TE4(:,:);

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 .5 1.5 .5];   % TE H
ra_4 = [3 .5 .5 1.5];   % TE H
index_PdCuH050_TE3_TE4_Pd =
find(ismember(ri_PdCuH050_TE3_TE4,ra_1,'rows'));
index_PdCuH050_TE3_TE4_Cu =
find(ismember(ri_PdCuH050_TE3_TE4,ra_2,'rows'));
index_PdCuH050_TE3_TE4_H_TE3 =
find(ismember(ri_PdCuH050_TE3_TE4,ra_3,'rows'));
index_PdCuH050_TE3_TE4_H_TE4 =
find(ismember(ri_PdCuH050_TE3_TE4,ra_4,'rows'));

global ri_PdCuH050_TE3_TE6
global index_PdCuH050_TE3_TE6_Pd
global index_PdCuH050_TE3_TE6_Cu
global index_PdCuH050_TE3_TE6_H_TE3
```

```
global index_PdCuH050_TE3_TE6_H_TE6

ri_PdCuH050_TE3_TE6 = PdCuH050_T_sorted_6x6x6_TE3_TE6(:,:);

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 .5 1.5 .5];   % TE H
ra_4 = [3 .5 1.5 1.5];   % TE H
index_PdCuH050_TE3_TE6_Pd =
find(ismember(ri_PdCuH050_TE3_TE6,ra_1,'rows'));
index_PdCuH050_TE3_TE6_Cu =
find(ismember(ri_PdCuH050_TE3_TE6,ra_2,'rows'));
index_PdCuH050_TE3_TE6_H_TE3 =
find(ismember(ri_PdCuH050_TE3_TE6,ra_3,'rows'));
index_PdCuH050_TE3_TE6_H_TE6 =
find(ismember(ri_PdCuH050_TE3_TE6,ra_4,'rows'));

global ri_PdCuH075_TE1_TE2_TE3
global index_PdCuH075_TE1_TE2_TE3_Pd
global index_PdCuH075_TE1_TE2_TE3_Cu
global index_PdCuH075_TE1_TE2_TE3_H_TE1
global index_PdCuH075_TE1_TE2_TE3_H_TE2
global index_PdCuH075_TE1_TE2_TE3_H_TE3

ri_PdCuH075_TE1_TE2_TE3 =
PdCuH075_T_sorted_6x6x6_TE1_TE2_TE3(:,:);

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 .5 .5 .5];   % TE H
ra_4 = [3 1.5 .5 .5];   % TE H
ra_5 = [3 .5 1.5 .5];   % TE H
index_PdCuH075_TE1_TE2_TE3_Pd =
find(ismember(ri_PdCuH075_TE1_TE2_TE3,ra_1,'rows'));
index_PdCuH075_TE1_TE2_TE3_Cu =
find(ismember(ri_PdCuH075_TE1_TE2_TE3,ra_2,'rows'));
index_PdCuH075_TE1_TE2_TE3_H_TE1 =
find(ismember(ri_PdCuH075_TE1_TE2_TE3,ra_3,'rows'));
index_PdCuH075_TE1_TE2_TE3_H_TE2 =
find(ismember(ri_PdCuH075_TE1_TE2_TE3,ra_4,'rows'));
index_PdCuH075_TE1_TE2_TE3_H_TE3 =
find(ismember(ri_PdCuH075_TE1_TE2_TE3,ra_5,'rows'));

global ri_PdCuH075_TE1_TE2_TE8
global index_PdCuH075_TE1_TE2_TE8_Pd
global index_PdCuH075_TE1_TE2_TE8_Cu
global index_PdCuH075_TE1_TE2_TE8_H_TE1
```

```
global index_PdCuH075_TE1_TE2_TE8_H_TE2
global index_PdCuH075_TE1_TE2_TE8_H_TE8

ri_PdCuH075_TE1_TE2_TE8 =
PdCuH075_T_sorted_6x6x6_TE1_TE2_TE8(:,:);

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 .5 .5 .5];   % TE H
ra_4 = [3 1.5 .5 .5];   % TE H
ra_5 = [3 1.5 1.5 1.5];   % TE H
index_PdCuH075_TE1_TE2_TE8_Pd =
find(ismember(ri_PdCuH075_TE1_TE2_TE8,ra_1,'rows'));
index_PdCuH075_TE1_TE2_TE8_Cu =
find(ismember(ri_PdCuH075_TE1_TE2_TE8,ra_2,'rows'));
index_PdCuH075_TE1_TE2_TE8_H_TE1 =
find(ismember(ri_PdCuH075_TE1_TE2_TE8,ra_3,'rows'));
index_PdCuH075_TE1_TE2_TE8_H_TE2 =
find(ismember(ri_PdCuH075_TE1_TE2_TE8,ra_4,'rows'));
index_PdCuH075_TE1_TE2_TE8_H_TE8 =
find(ismember(ri_PdCuH075_TE1_TE2_TE8,ra_5,'rows'));

global ri_PdCuH100_TE1_TE2_TE3_TE4
global index_PdCuH100_TE1_TE2_TE3_TE4_Pd
global index_PdCuH100_TE1_TE2_TE3_TE4_Cu
global index_PdCuH100_TE1_TE2_TE3_TE4_H_TE1
global index_PdCuH100_TE1_TE2_TE3_TE4_H_TE2
global index_PdCuH100_TE1_TE2_TE3_TE4_H_TE3
global index_PdCuH100_TE1_TE2_TE3_TE4_H_TE4

ri_PdCuH100_TE1_TE2_TE3_TE4 =
PdCuH100_T_sorted_6x6x6_TE1_TE2_TE3_TE4(:,:);

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 .5 .5 .5];   % TE H
ra_4 = [3 1.5 .5 .5];   % TE H
ra_5 = [3 .5 1.5 .5];   % TE H
ra_6 = [3 .5 .5 1.5];   % TE H
index_PdCuH100_TE1_TE2_TE3_TE4_Pd =
find(ismember(ri_PdCuH100_TE1_TE2_TE3_TE4,ra_1,'rows'));
index_PdCuH100_TE1_TE2_TE3_TE4_Cu =
find(ismember(ri_PdCuH100_TE1_TE2_TE3_TE4,ra_2,'rows'));
index_PdCuH100_TE1_TE2_TE3_TE4_H_TE1 =
find(ismember(ri_PdCuH100_TE1_TE2_TE3_TE4,ra_3,'rows'));
index_PdCuH100_TE1_TE2_TE3_TE4_H_TE2 =
find(ismember(ri_PdCuH100_TE1_TE2_TE3_TE4,ra_4,'rows'));
```

```
index_PdCuH100_TE1_TE2_TE3_TE4_H_TE3 =
find(ismember(ri_PdCuH100_TE1_TE2_TE3_TE4,ra_5,'rows'));
index_PdCuH100_TE1_TE2_TE3_TE4_H_TE4 =
find(ismember(ri_PdCuH100_TE1_TE2_TE3_TE4,ra_6,'rows'));

global ri_PdCuH100_TE1_TE2_TE7_TE8
global index_PdCuH100_TE1_TE2_TE7_TE8_Pd
global index_PdCuH100_TE1_TE2_TE7_TE8_Cu
global index_PdCuH100_TE1_TE2_TE7_TE8_H_TE1
global index_PdCuH100_TE1_TE2_TE7_TE8_H_TE2
global index_PdCuH100_TE1_TE2_TE7_TE8_H_TE7
global index_PdCuH100_TE1_TE2_TE7_TE8_H_TE8

ri_PdCuH100_TE1_TE2_TE7_TE8 =
PdCuH100_T_sorted_6x6x6_TE1_TE2_TE7_TE8(:,:);

ra_1 = [1 1 1 0];   % face center Pd
ra_2 = [2 0 0 0];   % corner Cu
ra_3 = [3 .5 .5 .5];   % TE H
ra_4 = [3 1.5 .5 .5];   % TE H
ra_5 = [3 1.5 .5 1.5];   % TE H
ra_6 = [3 1.5 1.5 1.5];   % TE H
index_PdCuH100_TE1_TE2_TE7_TE8_Pd =
find(ismember(ri_PdCuH100_TE1_TE2_TE7_TE8,ra_1,'rows'));
index_PdCuH100_TE1_TE2_TE7_TE8_Cu =
find(ismember(ri_PdCuH100_TE1_TE2_TE7_TE8,ra_2,'rows'));
index_PdCuH100_TE1_TE2_TE7_TE8_H_TE1 =
find(ismember(ri_PdCuH100_TE1_TE2_TE7_TE8,ra_3,'rows'));
index_PdCuH100_TE1_TE2_TE7_TE8_H_TE2 =
find(ismember(ri_PdCuH100_TE1_TE2_TE7_TE8,ra_4,'rows'));
index_PdCuH100_TE1_TE2_TE7_TE8_H_TE7 =
find(ismember(ri_PdCuH100_TE1_TE2_TE7_TE8,ra_5,'rows'));
index_PdCuH100_TE1_TE2_TE7_TE8_H_TE8 =
find(ismember(ri_PdCuH100_TE1_TE2_TE7_TE8,ra_6,'rows'));

% Used in Thesis
Xl_CuH = [0.78          2.1         1.7                 1.9];
X0_CuH = [0.755217     2.16562     1.76185
2.3539];
Xu_CuH = [0.8          2.2         1.8                 2];

tic

options.MaxFunEvals = 70000;
options.TolFun = 1.00e-7;
options.TolX = 1.00e-7;
```

```
    % Perform Constrained Optimization
    [x,fval] =
    fmincon(@objfun_18_07_09,X0_CuH,[],[],[],[],Xl_CuH,Xu_CuH,[],
    options)

    % stop timer
    toc

    Check_Fit_18_07_09(x,false,false)
    objfun_18_07_09(x);
    format long
    x
end
```

objfun.m

```
function f = objfun(x,weights)

    warning off
    clc
    format short

    global ri_PdH100_OC;
    global index_PdH100_OC_1;

    global ri_PdCuH000;
    global index_PdCuH000_Pd
    global index_PdCuH000_Cu

    global a_PdCuH025_o1;
    global a_PdCuH025_o2;
    global a_PdCuH025_T;
    global a_PdCuH050_O1_O2;
    global a_PdCuH050_O2_O2;
    global a_PdCuH050_T;
    global a_PdCuH075_O1_O2_O2;
    global a_PdCuH075_O2_O2_O2;
    global a_PdCuH075_T;
    global a_PdCuH100_O1_O2_O2_O2;
    global a_PdCuH100_T;

    global Ec_exp_PdCuH000;
    global Ec_exp_PdCuH025_O1;
    global Ec_exp_PdCuH025_O2;
    global Ec_exp_PdCuH025_T;
```

```
global Ec_exp_PdCuH050_O1_O2;
global Ec_exp_PdCuH050_O2_O2;
global Ec_exp_PdCuH050_T;
global Ec_exp_PdCuH075_O1_O2_O2;
global Ec_exp_PdCuH075_O2_O2_O2;
global Ec_exp_PdCuH075_T;
global Ec_exp_PdCuH100_O1_O2_O2_O2;
global Ec_exp_PdCuH100_T;

global a_PdCuH025_TE1
global a_PdCuH050_TE3_TE4
global a_PdCuH050_TE1_TE8
global a_PdCuH075_TE1_TE2_TE3
global a_PdCuH075_TE1_TE2_TE8
global a_PdCuH075_TE5_TE6_TE7
global a_PdCuH100_TE1_TE2_TE3_TE4
global a_PdCuH100_TE1_TE2_TE7_TE8
global a_PdCuH100_TE1_TE5_TE6_TE7

global Ec_exp_PdCuH025_TE1
global Ec_exp_PdCuH050_TE3_TE4
global Ec_exp_PdCuH050_TE1_TE8
global Ec_exp_PdCuH075_TE1_TE2_TE3
global Ec_exp_PdCuH075_TE1_TE2_TE8
global Ec_exp_PdCuH075_TE5_TE6_TE7
global Ec_exp_PdCuH100_TE1_TE2_TE3_TE4
global Ec_exp_PdCuH100_TE1_TE2_TE7_TE8
global Ec_exp_PdCuH100_TE1_TE5_TE6_TE7

global ri_PdCuH025_o1;
global index_PdCuH025_o1_Pd
global index_PdCuH025_o1_Cu
global index_PdCuH025_o1_H

global ri_PdCuH025_o2;
global index_PdCuH025_o2_Pd
global index_PdCuH025_o2_Cu
global index_PdCuH025_o2_H

global ri_PdCuH025_T_25_25_25;
global index_PdCuH025_T_Pd_25_25_25
global index_PdCuH025_T_Cu_25_25_25
global index_PdCuH025_T_H_25_25_25

global ri_PdCuH050_O1_O2;
global index_PdCuH050_O1_O2_Pd
global index_PdCuH050_O1_O2_Cu
```

```
global index_PdCuH050_O1_O2_H_O1
global index_PdCuH050_O1_O2_H_O2

global ri_PdCuH050_O2_O2;
global index_PdCuH050_O2_O2_Pd
global index_PdCuH050_O2_O2_Cu
global index_PdCuH050_O2_O2_H

global ri_PdCuH050_T_752525;
global index_PdCuH050_T_Pd_752525
global index_PdCuH050_T_Cu_752525
global index_PdCuH050_T_H_752525

global ri_PdCuH050_T_757575;
global index_PdCuH050_T_Pd_757575
global index_PdCuH050_T_Cu_757575
global index_PdCuH050_T_H_757575

global ri_PdCuH075_O1_O2_O2;
global index_PdCuH075_O1_O2_O2_Pd
global index_PdCuH075_O1_O2_O2_Cu
global index_PdCuH075_O1_O2_O2_H_O1
global index_PdCuH075_O1_O2_O2_H_O2

global ri_PdCuH075_O2_O2_O2;
global index_PdCuH075_O2_O2_O2_Pd
global index_PdCuH075_O2_O2_O2_Cu
global index_PdCuH075_O2_O2_O2_H

global ri_PdCuH075_T;
global index_PdCuH075_T_Pd
global index_PdCuH075_T_Cu
global index_PdCuH075_T_H

global ri_PdCuH100_O1_O2_O2_O2;
global index_PdCuH100_O1_O2_O2_O2_Pd
global index_PdCuH100_O1_O2_O2_O2_Cu
global index_PdCuH100_O1_O2_O2_O2_H_O1
global index_PdCuH100_O1_O2_O2_O2_H_O2

global ri_PdCuH100_T;
global index_PdCuH100_T_Pd
global index_PdCuH100_T_Cu
global index_PdCuH100_T_H

global NP
global NP2
```

```
global NP3

global x_Pd
global x_Cu

global S_Cu
global S_Pd

global x_PdH

global rc_1
global rc_2
global rc_3

% Pd Experimental data

[aFcce_Pd,EcFcce_Pd,C11e_Pd,C12e_Pd,C44e_Pd,Bme_Pd,Eve_Pd,aBc
ce_Pd,...
EcBcce_Pd] = parameters('Pd');

re = aFcce_Pd/sqrt(2);
Fe = EcFcce_Pd-Eve_Pd;
N  = 4;
V  = aFcce_Pd^3/N;
fe = EcFcce_Pd/V;

fe_S = S_Pd*EcFcce_Pd/V;

% Pd Fitting Parameters
Xi   = x_Pd(1);
phie = x_Pd(2);
S    = x_Pd(3);
B    = x_Pd(4);
n    = x_Pd(5);
rhoe = x_Pd(6);

%  Rho PdPd fitting parameters
X0_PdPd = [fe,Xi,re];
X0_PdPd_S = [fe_S,Xi,re];
%  F PdPd fitting parameters
X1_PdPd = [Fe,rhoe,n];
%  Phi PdPd fitting parametters
X2_PdPd = [phie,S,B,re];

% Cu Experimental data
```

```
[aFcce_Cu,EcFcce_Cu,C11e_Cu,C12e_Cu,C44e_Cu,Bme_Cu,Eve_Cu,aBc
ce_Cu,...
EcBcce_Cu] = parameters('Cu');

re = aFcce_Cu/sqrt(2);
Fe = EcFcce_Cu-Eve_Cu;
N  = 4;
V  = aFcce_Cu^3/N;
fe = EcFcce_Cu/V;

fe_S = S_Cu*EcFcce_Cu/V;


% Cu Fitting Parameters
Xi   = x_Cu(1);
phie = x_Cu(2);
S    = x_Cu(3);
B    = x_Cu(4);
n    = x_Cu(5);
rhoe = x_Cu(6);

%  Rho PdPd fitting parameters
X0_CuCu = [fe,Xi,re];
X0_CuCu_S = [fe_S,Xi,re];
%  F PdPd fitting parameters
X1_CuCu = [Fe,rhoe,n];
%  Phi PdPd fitting parametters
X2_CuCu = [phie,S,B,re];

X5_PdCu = [X0_PdPd_S X0_CuCu_S X2_PdPd X2_CuCu];

% Phi_HH
DHH =  x_PdH(1);
aHH =  x_PdH(2);
bHH =  x_PdH(3);
% f_HH
CH = x_PdH(4);
DH = x_PdH(5);
% Phi_PdH
DPdH = x_PdH(6);
aPdH = x_PdH(7);
bPdH = x_PdH(8);
% F_H
aH = x_PdH(9);
bH = x_PdH(10);
cH = x_PdH(11);
```

```
dH = x_PdH(12);
%
r0PdH = x_PdH(13);
r0HH =  x_PdH(14);

% f_HH
X0_HH  = [CH,DH];

% rho0H calculation
rho0H = 4.903820;

% F HH fitting parameters
X1_HH  = [aH,bH,cH,dH,rho0H];
% Phi HH fitting parametters
X2_HH  = [DHH,aHH,bHH,r0HH];
%
X22_HH = [X2_HH,X0_HH,X1_HH];
% Phi PdH fitting parametters
X4_PdH = [DPdH,aPdH,bPdH,r0PdH];

% Phi_CuH
DCuH = x(1);
aCuH = x(2);
bCuH = x(3);
r0CuH = x(4);

% Phi CuH fitting parametters
X4_CuH = [DCuH,aCuH,bCuH,r0CuH];

index_Pd = index_PdCuH000_Pd;
index_Cu = index_PdCuH000_Cu;
a = 3.821;

ri    = ri_PdCuH000;

XX = 1;
YY = 1;
ZZ = 1;

xx = 0.75;
yy = 0.25;

% Pd Central
Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
    @f_PdPd,X0_PdPd,...
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
```

```
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1);

    Ec_Pd075Cu025H000   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12 )/(xx
    + yy);

%*****************************************************************
% For Pd075Cu025H025_O1
%*****************************************************************
    ri = ri_PdCuH025_o1;
    index_Pd = index_PdCuH025_o1_Pd;
    index_Cu = index_PdCuH025_o1_Cu;
    index_H  = index_PdCuH025_o1_H;
    a = a_PdCuH025_o1;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
```

```
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H025_O1   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12 +
    zz * Ec_H_12)/(xx + yy + zz);

%*****************************************************************
% For Pd075Cu025H025_O2
%*****************************************************************
    ri = ri_PdCuH025_o2;
    index_Pd = index_PdCuH025_o2_Pd;
    index_Cu = index_PdCuH025_o2_Cu;
    index_H  = index_PdCuH025_o2_H;
    a = a_PdCuH025_o2;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
```

```
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H025_O2   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12 +
    zz * Ec_H_12)/(xx + yy + zz);

%*****************************************************************
% For Pd075Cu025H025_TE1
%*****************************************************************
    ri = ri_PdCuH025_T_25_25_25;
    index_Pd = index_PdCuH025_T_Pd_25_25_25;
    index_Cu = index_PdCuH025_T_Cu_25_25_25;
    index_H  = index_PdCuH025_T_H_25_25_25;

    a = a_PdCuH025_TE1;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
```

```
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H025_TE1   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12 +
    zz * Ec_H_12)/(xx + yy + zz);

%*****************************************************************
% For Pd075Cu025H050_O1_O2
%*****************************************************************
    ri = ri_PdCuH050_O1_O2;
    index_Pd = index_PdCuH050_O1_O2_Pd;
    index_Cu = index_PdCuH050_O1_O2_Cu;
    %     index_H  = index_PdCuH050_O1_O2_H_O1;
    index_H  = index_PdCuH050_O1_O2_H_O2;
    a = a_PdCuH050_O1_O2;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.50;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
```

```
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H050_O1_O2   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12
    + zz * Ec_H_12)/(xx + yy + zz);

%*****************************************************************
% For Pd075Cu025H050_O1_O2
%*****************************************************************
    ri = ri_PdCuH050_O1_O2;
    index_Pd = index_PdCuH050_O1_O2_Pd;
    index_Cu = index_PdCuH050_O1_O2_Cu;
    index_H_O1  = index_PdCuH050_O1_O2_H_O1;
    index_H_O2  = index_PdCuH050_O1_O2_H_O2;
    a = a_PdCuH050_O1_O2;


    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;
    zzz = 0.25;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
```

```
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_12 =    Ec3(ri,a,index_H_O1,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_22 =    Ec3(ri,a,index_H_O2,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);


    Ec_Pd075Cu025H050_O1_O2   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12
    + zz * Ec_H_12 + zzz * Ec_H_22)/(xx + yy + zz + zzz);

%*************************************************************
% For Pd075Cu025H025_O2_O2
%*************************************************************
    ri = ri_PdCuH050_O2_O2;
    index_Pd = index_PdCuH050_O2_O2_Pd;
    index_Cu = index_PdCuH050_O2_O2_Cu;
    index_H  = index_PdCuH050_O2_O2_H;
```

```
    a = a_PdCuH050_O2_O2;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.50;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H050_O2_O2   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12
    + zz * Ec_H_12)/(xx + yy + zz);

%*****************************************************************
% For Pd075Cu025H050_TE1_TE2
%*****************************************************************
    global ri_PdCuH050_TE1_TE2;
    global index_PdCuH050_TE1_TE2_Pd
```

```
global index_PdCuH050_TE1_TE2_Cu
global index_PdCuH050_TE1_TE2_H_TE1
global index_PdCuH050_TE1_TE2_H_TE2

ri = ri_PdCuH050_TE1_TE2;
index_Pd = index_PdCuH050_TE1_TE2_Pd;
index_Cu = index_PdCuH050_TE1_TE2_Cu;
index_H_TE1  = index_PdCuH050_TE1_TE2_H_TE1;
index_H_TE2  = index_PdCuH050_TE1_TE2_H_TE2;

a = a_PdCuH050_T;

XX = 1;
YY = 1;
ZZ = 1;

xx = 0.75;
yy = 0.25;
zz = 0.50;
zzz = 0.25;

% Pd Central
Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
    @f_PdPd,X0_PdPd,...
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_PdH,rc_1);
% Cu Central
Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
    @f_PdPd,X0_CuCu,...
    @F_Pd,X1_CuCu,...
    @phi_PdPd,X2_CuCu,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_CuH,rc_1);

% H Central
Ec_H_12 =    Ec3(ri,a,index_H_TE1,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
```

```
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_22 =    Ec3(ri,a,index_H_TE2,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H050_TE1_TE2   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zzz * Ec_H_12 + zzz * Ec_H_22)/(xx + yy + zzz +
    zzz);

%*****************************************************************
% For Pd075Cu025H050_TE1_TE8
%*****************************************************************
    global ri_PdCuH050_TE1_TE8;
    global index_PdCuH050_TE1_TE8_Pd
    global index_PdCuH050_TE1_TE8_Cu
    global index_PdCuH050_TE1_TE8_H_TE1
    global index_PdCuH050_TE1_TE8_H_TE8

    ri = ri_PdCuH050_TE1_TE8;
    index_Pd = index_PdCuH050_TE1_TE8_Pd;
    index_Cu = index_PdCuH050_TE1_TE8_Cu;
    index_H_TE1  = index_PdCuH050_TE1_TE8_H_TE1;
    index_H_TE8  = index_PdCuH050_TE1_TE8_H_TE8;

    a = a_PdCuH050_T;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.50;
    zzz = 0.25;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
```

```
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_12 =    Ec3(ri,a,index_H_TE1,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_22 =    Ec3(ri,a,index_H_TE8,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H050_TE1_TE8   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zzz * Ec_H_12+ zzz * Ec_H_22)/(xx + yy + zzz +
    zzz);

%*****************************************************************
% For Pd075Cu025H050_TE3_TE4
%*****************************************************************
    global ri_PdCuH050_TE3_TE4
    global index_PdCuH050_TE3_TE4_Pd
    global index_PdCuH050_TE3_TE4_Cu
    global index_PdCuH050_TE3_TE4_H_TE3
    global index_PdCuH050_TE3_TE4_H_TE4
```

```
ri = ri_PdCuH050_TE3_TE4;
index_Pd = index_PdCuH050_TE3_TE4_Pd;
index_Cu = index_PdCuH050_TE3_TE4_Cu;
index_H_TE3  = index_PdCuH050_TE3_TE4_H_TE3;
index_H_TE4  = index_PdCuH050_TE3_TE4_H_TE4;

a = a_PdCuH050_TE1_TE8;

XX = 1;
YY = 1;
ZZ = 1;

xx = 0.75;
yy = 0.25;
zz = 0.25;
zzz = 0.25;

% Pd Central
Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
    @f_PdPd,X0_PdPd,...
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_PdH,rc_1);
% Cu Central
Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
    @f_PdPd,X0_CuCu,...
    @F_Pd,X1_CuCu,...
    @phi_PdPd,X2_CuCu,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_CuH,rc_1);

% H Central
Ec_H_12 =    Ec3(ri,a,index_H_TE3,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);
```

```
    % H Central
    Ec_H_22 =    Ec3(ri,a,index_H_TE4,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H050_TE3_TE4   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zz * Ec_H_12 + zzz * Ec_H_22)/(xx + yy + zz +
    zzz);

%*************************************************************
% For Pd075Cu025H075_O1_O2_O2
%*************************************************************
    ri = ri_PdCuH075_O1_O2_O2;
    index_Pd = index_PdCuH075_O1_O2_O2_Pd;
    index_Cu = index_PdCuH075_O1_O2_O2_Cu;
    index_H_O1  = index_PdCuH075_O1_O2_O2_H_O1;
    index_H_O2  = index_PdCuH075_O1_O2_O2_H_O2;

    a = a_PdCuH075_O1_O2_O2;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;
    zzz = 0.50;
    zzzz = 0.75;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
```

```
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_12 =    Ec3(ri,a,index_H_O1,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_22 =    Ec3(ri,a,index_H_O2,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H075_O1_O2_O2   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zz * Ec_H_12 + zzz * Ec_H_22)/(xx + yy + zz +
    zzz);

%*****************************************************************
% For Pd075Cu025H075_O2_O2_O2
%*****************************************************************
    ri = ri_PdCuH075_O2_O2_O2;
    index_Pd = index_PdCuH075_O2_O2_O2_Pd;
    index_Cu = index_PdCuH075_O2_O2_O2_Cu;
    index_H  = index_PdCuH075_O2_O2_O2_H;

    a = a_PdCuH075_O2_O2_O2;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
```

```
    zz = 0.75;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H075_O2_O2_O2   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zz * Ec_H_12)/(xx + yy + zz);

%*************************************************************
% For Pd075Cu025H075_TE1_TE2_TE3
%*************************************************************
    global ri_PdCuH075_TE1_TE2_TE3
    global index_PdCuH075_TE1_TE2_TE3_Pd
    global index_PdCuH075_TE1_TE2_TE3_Cu
    global index_PdCuH075_TE1_TE2_TE3_H_TE1
    global index_PdCuH075_TE1_TE2_TE3_H_TE2
    global index_PdCuH075_TE1_TE2_TE3_H_TE3

    ri = ri_PdCuH075_TE1_TE2_TE3;
    index_Pd = index_PdCuH075_TE1_TE2_TE3_Pd;
    index_Cu = index_PdCuH075_TE1_TE2_TE3_Cu;
```

```
index_H_TE1  = index_PdCuH075_TE1_TE2_TE3_H_TE1;
index_H_TE2  = index_PdCuH075_TE1_TE2_TE3_H_TE2;
index_H_TE3  = index_PdCuH075_TE1_TE2_TE3_H_TE3;

a = a_PdCuH075_TE1_TE2_TE3;

XX = 1;
YY = 1;
ZZ = 1;

xx = 0.75;
yy = 0.25;
zz = 0.25;
zzz = 0.25;
zzzz = 0.25;

% Pd Central
Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
    @f_PdPd,X0_PdPd,...
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_PdH,rc_1);
% Cu Central
Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
    @f_PdPd,X0_CuCu,...
    @F_Pd,X1_CuCu,...
    @phi_PdPd,X2_CuCu,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_12 =    Ec3(ri,a,index_H_TE1,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_22 =    Ec3(ri,a,index_H_TE2,3,ZZ,...
```

```
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_32 =    Ec3(ri,a,index_H_TE3,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H075_TE1_TE2_TE3   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zz * Ec_H_12 + zzz * Ec_H_22 + zzzz * Ec_H_32)/(xx
    + yy + zz + zzz + zzzz);

%*****************************************************************
% For Pd075Cu025H075_TE1_TE2_TE8
%*****************************************************************
    global ri_PdCuH075_TE1_TE2_TE8
    global index_PdCuH075_TE1_TE2_TE8_Pd
    global index_PdCuH075_TE1_TE2_TE8_Cu
    global index_PdCuH075_TE1_TE2_TE8_H_TE1
    global index_PdCuH075_TE1_TE2_TE8_H_TE2
    global index_PdCuH075_TE1_TE2_TE8_H_TE8

    ri = ri_PdCuH075_TE1_TE2_TE8;
    index_Pd = index_PdCuH075_TE1_TE2_TE8_Pd;
    index_Cu = index_PdCuH075_TE1_TE2_TE8_Cu;
    index_H_TE1  = index_PdCuH075_TE1_TE2_TE8_H_TE1;
    index_H_TE2  = index_PdCuH075_TE1_TE2_TE8_H_TE2;
    index_H_TE8  = index_PdCuH075_TE1_TE2_TE8_H_TE8;

    a = a_PdCuH075_TE1_TE2_TE8;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
```

```
zz = 0.25;
zzz = 0.25;
zzzz = 0.25;

% Pd Central
Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
    @f_PdPd,X0_PdPd,...
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_PdH,rc_1);
% Cu Central
Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
    @f_PdPd,X0_CuCu,...
    @F_Pd,X1_CuCu,...
    @phi_PdPd,X2_CuCu,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_12 =    Ec3(ri,a,index_H_TE1,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_22 =    Ec3(ri,a,index_H_TE2,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_32 =    Ec3(ri,a,index_H_TE8,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
```

```
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H075_TE1_TE2_TE8   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zz * Ec_H_12 + zzz * Ec_H_22 + zzzz * Ec_H_32)/(xx
    + yy + zz + zzz + zzzz);
%************************************************************
% For Pd075Cu025H100_O1_O2_O2_O2
%************************************************************
    ri = ri_PdCuH100_O1_O2_O2_O2;
    index_Pd = index_PdCuH100_O1_O2_O2_O2_Pd;
    index_Cu = index_PdCuH100_O1_O2_O2_O2_Cu;
    index_H_O1  = index_PdCuH100_O1_O2_O2_O2_H_O1;
    index_H_O2  = index_PdCuH100_O1_O2_O2_O2_H_O2;

    a = a_PdCuH100_O1_O2_O2_O2;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;
    zzz = 0.75;
    zzzz = 1.00;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
```

```
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_12 =    Ec3(ri,a,index_H_O1,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_22 =    Ec3(ri,a,index_H_O2,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H100_O1_O2_O2_O2   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zz * Ec_H_12 + zzz * Ec_H_22)/(xx + yy + zz +
    zzz);

%*****************************************************************
% For Pd075Cu025H100_TE1_TE2_TE3_TE4
%*****************************************************************
    global ri_PdCuH100_TE1_TE2_TE3_TE4
    global index_PdCuH100_TE1_TE2_TE3_TE4_Pd
    global index_PdCuH100_TE1_TE2_TE3_TE4_Cu
    global index_PdCuH100_TE1_TE2_TE3_TE4_H_TE1
    global index_PdCuH100_TE1_TE2_TE3_TE4_H_TE2
    global index_PdCuH100_TE1_TE2_TE3_TE4_H_TE3
    global index_PdCuH100_TE1_TE2_TE3_TE4_H_TE4

    ri = ri_PdCuH100_TE1_TE2_TE3_TE4;
    index_Pd = index_PdCuH100_TE1_TE2_TE3_TE4_Pd;
    index_Cu = index_PdCuH100_TE1_TE2_TE3_TE4_Cu;
    index_H_TE1  = index_PdCuH100_TE1_TE2_TE3_TE4_H_TE1;
    index_H_TE2  = index_PdCuH100_TE1_TE2_TE3_TE4_H_TE2;
    index_H_TE3  = index_PdCuH100_TE1_TE2_TE3_TE4_H_TE3;
    index_H_TE4  = index_PdCuH100_TE1_TE2_TE3_TE4_H_TE4;

    a = a_PdCuH100_T;
```

```
XX = 1;
YY = 1;
ZZ = 1;

xx = 0.75;
yy = 0.25;
zz = 0.25;

% Pd Central
Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
    @f_PdPd,X0_PdPd,...
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_PdH,rc_1);
% Cu Central
Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
    @f_PdPd,X0_CuCu,...
    @F_Pd,X1_CuCu,...
    @phi_PdPd,X2_CuCu,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_12 =    Ec3(ri,a,index_H_TE1,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_22 =    Ec3(ri,a,index_H_TE2,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);
```

```
    %  H Central
    Ec_H_32 =    Ec3(ri,a,index_H_TE3,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_42 =    Ec3(ri,a,index_H_TE4,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H100_TE1_TE2_TE3_TE4   = ( xx * Ec_Pd_12  + yy
    * Ec_Cu_12 + zz * Ec_H_12 + zz * Ec_H_22 + zz * Ec_H_32 + zz
    * Ec_H_42)/(xx + yy + 4*zz);

%*****************************************************************
% For Pd075Cu025H100_TE1_TE2_TE7_TE8
%*****************************************************************
    global ri_PdCuH100_TE1_TE2_TE7_TE8
    global index_PdCuH100_TE1_TE2_TE7_TE8_Pd
    global index_PdCuH100_TE1_TE2_TE7_TE8_Cu
    global index_PdCuH100_TE1_TE2_TE7_TE8_H_TE1
    global index_PdCuH100_TE1_TE2_TE7_TE8_H_TE2
    global index_PdCuH100_TE1_TE2_TE7_TE8_H_TE7
    global index_PdCuH100_TE1_TE2_TE7_TE8_H_TE8

    ri = ri_PdCuH100_TE1_TE2_TE7_TE8;
    index_Pd = index_PdCuH100_TE1_TE2_TE7_TE8_Pd;
    index_Cu = index_PdCuH100_TE1_TE2_TE7_TE8_Cu;
    index_H_TE1  = index_PdCuH100_TE1_TE2_TE7_TE8_H_TE1;
    index_H_TE2  = index_PdCuH100_TE1_TE2_TE7_TE8_H_TE2;
    index_H_TE7  = index_PdCuH100_TE1_TE2_TE7_TE8_H_TE7;
    index_H_TE8  = index_PdCuH100_TE1_TE2_TE7_TE8_H_TE8;

    a = a_PdCuH100_TE1_TE2_TE7_TE8;

    XX = 1;
    YY = 1;
```

```
ZZ = 1;

xx = 0.75;
yy = 0.25;
zz = 0.25;

% Pd Central
Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
@f_PdPd,X0_PdPd,...
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_PdH,rc_1);
% Cu Central
Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
    @f_PdPd,X0_CuCu,...
    @F_Pd,X1_CuCu,...
    @phi_PdPd,X2_CuCu,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_CuH,rc_1);

% H Central
Ec_H_12 =    Ec3(ri,a,index_H_TE1,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

% H Central
Ec_H_22 =    Ec3(ri,a,index_H_TE2,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

% H Central
Ec_H_32 =    Ec3(ri,a,index_H_TE7,3,ZZ,...
```

```
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

% H Central
Ec_H_42 =    Ec3(ri,a,index_H_TE8,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

Ec_Pd075Cu025H100_TE1_TE2_TE7_TE8   = ( xx * Ec_Pd_12  + yy
* Ec_Cu_12 + zz * Ec_H_12 + zz * Ec_H_22 + zz * Ec_H_32 + zz
* Ec_H_42)/(xx + yy + 4*zz);

d0 = (Ec_Pd075Cu025H000 - (Ec_exp_PdCuH000))^2;
d1  = (Ec_Pd075Cu025H025_O1 - (Ec_exp_PdCuH025_O1))^2;
d2  = (Ec_Pd075Cu025H025_O2 - (Ec_exp_PdCuH025_O2))^2;
d3  = (Ec_Pd075Cu025H025_TE1 - (Ec_exp_PdCuH025_TE1))^2;
d4  = (Ec_Pd075Cu025H050_O1_O2 - (Ec_exp_PdCuH050_O1_O2))^2;
d5  = (Ec_Pd075Cu025H050_O2_O2 - (Ec_exp_PdCuH050_O2_O2))^2;
d6  = (Ec_Pd075Cu025H050_TE3_TE4 -
(Ec_exp_PdCuH050_TE3_TE4))^2;
d7  = (Ec_Pd075Cu025H050_TE1_TE8 -
(Ec_exp_PdCuH050_TE1_TE8))^2;
d8  = (Ec_Pd075Cu025H075_O1_O2_O2 -
(Ec_exp_PdCuH075_O1_O2_O2))^2;
d9  = (Ec_Pd075Cu025H075_O2_O2_O2 -
(Ec_exp_PdCuH075_O2_O2_O2))^2;
d10  = (Ec_Pd075Cu025H075_TE1_TE2_TE3 -
(Ec_exp_PdCuH075_TE1_TE2_TE3))^2;
d11  = (Ec_Pd075Cu025H075_TE1_TE2_TE8 -
(Ec_exp_PdCuH075_TE1_TE2_TE8))^2;
d12  = (Ec_Pd075Cu025H100_O1_O2_O2_O2 -
(Ec_exp_PdCuH100_O1_O2_O2_O2))^2;
d13  = (Ec_Pd075Cu025H100_TE1_TE2_TE3_TE4 -
(Ec_exp_PdCuH100_TE1_TE2_TE3_TE4))^2;
d14  = (Ec_Pd075Cu025H100_TE1_TE2_TE7_TE8 -
(Ec_exp_PdCuH100_TE1_TE2_TE7_TE8))^2;
```

```
    f  =  sqrt(d1 +  d2  +  d3  +  d4  +  d5  +  d6  +  d7 +  d8
    +  d9  +  d10  +  d11 + d12 + d13 + d14)

end


%*************************************************************
function [Ec_,dEda_]=
fitProperties_1(r,a,N,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
F,dFdrho,d2Fdrho2,X1,phi_11,dphidr_11,d2phidr2_11,X2,rc_1,...
t2,x,f_22,dfdr_22,d2fdr2_22,X3,phi_12,dphidr_12,d2phidr2_12,X4,r
c_2)
    Ec_ =
    Ec(r,a,index,t1,y,f_11,X0,F,X1,phi_11,X2,rc_1,t2,x,f_22,X3,ph
    i_12,X4,rc_2);

    dEda_ =
    dEda(r,a,index,t1,y,f_11,dfdr_11,X0,dFdrho,X1,dphidr_11,X2,rc
    _1,...
        t2,x,f_22,dfdr_22,X3,dphidr_12,X4,rc_2);
end

%*************************************************************
% Property Functions Used In Parameters Fitting
%*************************************************************
function [Ec_,dEda_,C11,C12,C44,Bm_,Ev_] =
fitProperties(r,a,N,index,...
t1,y,f_11,dfdr_11,d2fdr2_11,X0,F,dFdrho,d2Fdrho2,X1,phi_11,...
dphidr_11,d2phidr2_11,X2,rc_1,t2,x,f_22,dfdr_22,d2fdr2_22,X3,...
    phi_12,dphidr_12,d2phidr2_12,X4,rc_2)
    % Cohesive Energy
    Ec_ =
    Ec(r,a,index,t1,y,f_11,X0,F,X1,phi_11,X2,rc_1,t2,x,f_22,X3,..
    .
    phi_12,X4,rc_2);
    % First Derivative of Ec with respect to a
    dEda_ =
    dEda(r,a,index,t1,y,f_11,dfdr_11,X0,dFdrho,X1,dphidr_11,X2,..
    .
    rc_1,t2,x,f_22,dfdr_22,X3,dphidr_12,X4,rc_2);
    % C11 Elastic Constant
    C11 =
    Cijkl(1,1,1,1,r,a,N,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
    dFdrho,d2Fdrho2,X1,dphidr_11,d2phidr2_11,X2,rc_1,...
    t2,x,f_22,dfdr_22,d2fdr2_22,X3,dphidr_12,d2phidr2_12,X4,rc_2)
    ;
    % C12 Elastic Constant
```

```
    C12 =
    Cijkl(1,1,2,2,r,a,N,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
    dFdrho,d2Fdrho2,X1,dphidr_11,d2phidr2_11,X2,rc_1,...
    t2,x,f_22,dfdr_22,d2fdr2_22,X3,dphidr_12,d2phidr2_12,X4,rc_2)
    ;
    % C44 Elastic Constant
    C44 =
    Cijkl(2,3,2,3,r,a,N,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
    dFdrho,d2Fdrho2,X1,dphidr_11,d2phidr2_11,X2,rc_1,...
    t2,x,f_22,dfdr_22,d2fdr2_22,X3,dphidr_12,d2phidr2_12,X4,rc_2)
    ;
    % Bulk Modulus
    Bm_ = Bm(r,a,N,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
    dFdrho,d2Fdrho2,X1,d2phidr2_11,X2,rc_1,t2,x,f_22,dfdr_22,...
    d2fdr2_22,X3,d2phidr2_12,X4,rc_2);
    % Vacancy Formation
    Ev_ = 0;
    % Ev_ = Ev(index,t1,a,phi_11,X2,f_11,X0,F,X1,r,rc_1);
end

%*************************************************************
% Cohesive Energy for Interstitial Solid Solution
%*************************************************************
function f =
Ec3(ri,a,index,t1,y,f_11,X0,F_11,X1,phi_11,X2,rc_1,t2,x,...
    f_22,X3,phi_12,X4,rc_2,t3,z,f_33,X5,phi_13,X6,rc_3)

    f = 0;
    Rho_12 = 0;
    Phi_12 = 0;
    Rho_13 = 0;
    Phi_13 = 0;

    Rho_11 = y * Rho(index,t1,ri,a,rc_1,f_11,X0);
    Phi_11 = y * Phi(index,t1,ri,a,rc_1,phi_11,X2);

    if nargin >= 19
        Rho_12  = x * Rho(index,t2,ri,a,rc_2,f_22,X3);
        Phi_12  = x * Phi(index,t2,ri,a,rc_2,phi_12,X4);
    end

    if nargin >= 26
        Rho_13  = z * Rho(index,t3,ri,a,rc_3,f_33,X5);
        Phi_13  = z * Phi(index,t3,ri,a,rc_3,phi_13,X6);
    end

    Rho_1 = Rho_11 + Rho_12 + Rho_13;
```

```
    F_1 = F_11(Rho_1,X1);
    f = F_1 + 0.5 * Phi_11 + 0.5 * Phi_12 + 0.5 * Phi_13;
end


%*************************************************************
% Elestic Constants for an Interstitial Solid Solution
%*************************************************************
function f =
Cijkl(i,j,k,l,r,a,N,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
    dFdrho,d2Fdrho2,X1,dphidr_11,d2phidr2_11,X2,rc_1,t2,...
    x,f_22,dfdr_22,d2fdr2_22,X3,dphidr_12,d2phidr2_12,X4,rc_2)

    Rho_12 = 0;
    Vij_12 = 0;
    Vkl_12 = 0;
    Wijkl_12 = 0;
    Bijkl_12 = 0;

    Rho_11   = y * Rho(index,t1,r,a,rc_1,f_11,X0);
    Vij_11   = y * Vij(index,t1,r,a,rc_1,dfdr_11,X0,i,j,0,0);
    Vkl_11   = y * Vij(index,t1,r,a,rc_1,dfdr_11,X0,0,0,k,l);
    Wijkl_11 = y *
    Wijkl(index,t1,r,a,rc_1,dfdr_11,d2fdr2_11,X0,i,j,k,l);
    Bijkl_11 = y *
    Bijkl(index,t1,r,a,rc_1,dphidr_11,d2phidr2_11,X2,i,j,k,l);

    if nargin == 31
        Rho_12   = x * Rho(index,t2,r,a,rc_2,f_22,X3);
        Vij_12   = x *
    Vij(index,t2,r,a,rc_2,dfdr_22,X3,i,j,0,0);
        Vkl_12   = x *
    Vij(index,t2,r,a,rc_2,dfdr_22,X3,0,0,k,l);
        Wijkl_12 = x *
        Wijkl(index,t2,r,a,rc_2,dfdr_22,d2fdr2_22,X3,i,j,k,l);
        Bijkl_12 = x *
        Bijkl(index,t2,r,a,rc_2,dphidr_12,d2phidr2_12,X4,i,j,k,l)
        ;
    end
    Rho_1      = Rho_11 + Rho_12;
    dFdrho_1   = dFdrho(Rho_1,X1);
    d2Fdrho2_1 = d2Fdrho2(Rho_1,X1);
    % Elastic constants Cijkl
    V = (a^3)/N;
    Cijkl = (1/V)*( d2Fdrho2_1*(Vij_11 + Vij_12)*(Vkl_11 +
    Vkl_12) +...
        dFdrho_1*(Wijkl_11 + Wijkl_12) + Bijkl_11 + Bijkl_12 );
```

```
    f = Cijkl * 1.602176462;
end

%*************************************************************
% Stress for an Interstitial solid solution
%*************************************************************
function f =
S(i,j,r,a,N,index,t1,y,f_11,dfdr_11,X0,dFdrho,X1,dphidr_11,X2,rc
_1,t2,x,f_22,dfdr_22,X3,dphidr_12,X4,rc_2)

    Rho_12 = 0;
    Vij_12 = 0;
    Aij_12 = 0;
    %
    Rho_11 = y * Rho(index,t1,r,a,rc_1,f_11,X0);
    Vij_11 = y * Vij(index,t1,r,a,rc_1,dfdr_11,X0,i,j,0,0);
    Aij_11 = y * Aij(index,t1,r,a,rc_1,dphidr_11,X2,i,j,0,0);
    if nargin == 24
        Rho_12 = x * Rho(index,t2,r,a,rc_2,f_22,X3);
        Vij_12 = x * Vij(index,t2,r,a,rc_2,dfdr_22,X3,i,j,0,0);
        Aij_12 = x *
        Aij(index,t2,r,a,rc_2,dphidr_12,X4,i,j,0,0);
    end
    Rho_1     = Rho_11 + Rho_12;
    dFdrho_1  = dFdrho(Rho_1,X1);
    V = (a^3)/N;
    f = (1/V) * (Aij_11 + Aij_12 + dFdrho_1 * (Vij_11 +
    Vij_12));
end

%*************************************************************
% First derivative of Cohesive Energy with respect to a for an
% Interstitial solid solution
%*************************************************************
function  f =
dEda(ri,a,index,t1,y,f_11,dfdr_11,X0,dFdrho,X1,dphidr_11,X2,rc_1
,t2,x,f_22,dfdr_22,X3,dphidr_12,X4,rc_2)

    Rho_12    = 0;
    dRhoda_12 = 0;
    dPhida_12 = 0;

    Rho_11    = y * Rho(index,t1,ri,a,rc_1,f_11,X0);
    dRhoda_11 = y * dRhoda(index,t1,ri,a,rc_1,dfdr_11,X0);
    dPhida_11 = y * dPhida(index,t1,ri,a,rc_1,dphidr_11,X2);

    if nargin == 21
```

```
        Rho_12    = x * Rho(index,t2,ri,a,rc_2,f_22,X3);
        dRhoda_12 = x * dRhoda(index,t2,ri,a,rc_2,dfdr_22,X3);
        dPhida_12 = x * dPhida(index,t2,ri,a,rc_2,dphidr_12,X4);
    end

    Rho_1     = Rho_11    + Rho_12;
    dRhoda_1  = dRhoda_11 + dRhoda_12;
    dFda_1 = dFda(dFdrho,Rho_1,dRhoda_1,X1);
    f = dFda_1 + 0.5 * dPhida_11 + 0.5 * dPhida_12;
end

%*************************************************************
% Second derivative of Cohesive Energy with respect to a for
Interstitial
% Solid Solution
%*************************************************************
function  f =
d2Eda2(ri,a,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
    dFdrho,d2Fdrho2,X1,d2phidr2_11,X2,rc_1,t2,x,...
    f_22,dfdr_22,d2fdr2_22,X3,d2phidr2_12,X4,rc_2)

    Rho_12      = 0;
    dRhoda_12   = 0;
    d2Rhoda2_12 = 0;
    d2Phida2_12 = 0;

    Rho_11      = y * Rho(index,t1,ri,a,rc_1,f_11,X0);
    dRhoda_11   = y * dRhoda(index,t1,ri,a,rc_1,dfdr_11,X0);
    d2Rhoda2_11 = y * d2Rhoda2(index,t1,ri,a,rc_1,d2fdr2_11,X0);
    d2Phida2_11 = y *
    d2Phida2(index,t1,ri,a,rc_1,d2phidr2_11,X2);

    if nargin == 24
        Rho_12      = x * Rho(index,t2,ri,a,rc_2,f_22,X3);
        dRhoda_12   = x * dRhoda(index,t2,ri,a,rc_2,dfdr_22,X3);
        d2Rhoda2_12 = x *
        d2Rhoda2(index,t2,ri,a,rc_2,d2fdr2_22,X3);
        d2Phida2_12 = x *
        d2Phida2(index,t2,ri,a,rc_2,d2phidr2_12,X4);
    end
    Rho_1       = Rho_11 + Rho_12;
    dRhoda_1    = dRhoda_11 + dRhoda_12;
    d2Rhoda2_1  = d2Rhoda2_11 + d2Rhoda2_12;
    d2Fda2_1 =
    d2Fda2(dFdrho,d2Fdrho2,Rho_1,dRhoda_1,d2Rhoda2_1,X1);
    f = d2Fda2_1 + 0.5 * d2Phida2_11 + 0.5 * d2Phida2_12;
end
```

```
%*********************************************************************
% Bulk modulous
%*********************************************************************
function f = Bm(ri,a,N,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
    dFdrho,d2Fdrho2,X1,d2phidr2_11,X2,rc_1,t2,x,f_22,...
    dfdr_22,d2fdr2_22,X3,d2phidr2_12,X4,rc_2)
    d2Eda2_ = 0;
    if nargin == 16
        d2Eda2_ =
        d2Eda2(ri,a,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
        dFdrho,d2Fdrho2,X1,d2phidr2_11,X2,rc_1);
    end
    if nargin == 25
        d2Eda2_ =
        d2Eda2(ri,a,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
        dFdrho,d2Fdrho2,X1,d2phidr2_11,X2,rc_1,t2,x,f_22,...
        dfdr_22,d2fdr2_22,X3,d2phidr2_12,X4,rc_2);
    end
    V = a^3/N;
    f = ((a)^2/(9.0*V))*d2Eda2_;
    f = f * 1.602176462;
end

%*********************************************************************
% Vacancy formation Energy
%*********************************************************************
function f = Ev(index,type,a,phi,X0,f,X1,F,X2,rij,rc)
    sum0 = 0;
    sum1 = 0;
    sum2 = 0;
    rho = Rho(index,type,rij,a,rc,f,X1);
    n = length(rij(:,1));
    for i = 1:n
        id = rij(i,1);
        if i ~= index && id == type
            l1 = (a/2)*rij(i,2);
            l2 = (a/2)*rij(i,3);
            l3 = (a/2)*rij(i,4);
            ri = sqrt(l1^2+l2^2+l3^2);
            if ri <= rc
                q0 = phi(ri,X0);
                f1 = f(ri,X1);
                sum0 = sum0 + q0;
                sum1 = sum1 + F(rho,X2);
                sum2 = sum2 + F(rho-f1,X2);
            end
```

```
        end
    end
    f = -0.5 * sum0 + sum2 - sum1;
end

%*****************************************************************
% The Embedding Atom Model EAM
%*****************************************************************

% Total pair potential energy for a central atom as function of
interatomic
% distance
% index: array index of central atom
% type : Id of atom type
% ri   : atoms position in Lattice Crystal Structure
% a    : the lattice constant
% rc   : the cuttof radius
% phi  : the pair potential function
% X    : array of pair potential parameters
function f = Phi(index,type,ri,a,rc,phi,X)
    f = Sumfunc(index,type,ri,a,rc,phi,X);
end

% Total first derivative of pair potential with respect to
interatomic
% distance rij
function f = dPhidr(index,type,ri,a,rc,dphidr,X)
    f = Sumfunc(index,type,ri,a,rc,dphidr,X);
end

% Total second derivative of pair potential with respect to
interatomic
% distance rij
function f = d2Phidr2(index,type,ri,a,rc,d2phidr2,X)
    f = Sumfunc(index,type,ri,a,rc,d2phidr2,X);
end

% Total first derivative of pair potential with respect to a
function f = dPhida(index,type,ri,a,rc,dphidr,X)
    f = SumfuncM(index,type,ri,a,rc,dphidr,X);
end

% Total second derivative of pair potential with respect to a
function f = d2Phida2(index,type,ri,a,rc,d2phidr2,X)
    f = SumfuncM2(index,type,ri,a,rc,d2phidr2,X);
end
```

```
% Total Electron Density for a central atom as function of
interatomic
% distance
% index: array index of central atom
% type : Id of atom type
% ri   : atoms position in Lattice Crystal Structure
% a    : the lattice constant
% rc   : the cuttof radius
% f    : the atomic density function
% X    : array of atomic density function parameters
function f = Rho(index,type,ri,a,rc,f,X)
    f = Sumfunc(index,type,ri,a,rc,f,X);
end

% Total first derivative of electron density with respect to
interatomic
% distance rij
function f = dRhodr(index,type,ri,a,rc,dfdr,X)
    f = Sumfunc(index,type,ri,a,rc,dfdr,X);
end

% Total second derivative of electron density with respect to
interatomic
% distance rij
function f = d2Rhodr2(index,type,ri,a,rc,d2fdr2,X)
    f = Sumfunc(index,type,ri,a,d2fdr2,X);
end

% Total first derivative of electron density with respect a
function f = dRhoda(index,type,ri,a,rc,dfdr,X)
    f = SumfuncM(index,type,ri,a,rc,dfdr,X);
end

% Total second derivative of electron density with respect a
function f = d2Rhoda2(index,type,ri,a,rc,d2fdr2,X)
    f = SumfuncM2(index,type,ri,a,rc,d2fdr2,X);
end

% Total first derivative of electron density with respect
ri,rj,rk, and rl
function f = dRhodrij(index,type,ri,a,rc,dfdr,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,dfdr,X,i,j,k,l,1);
end

% second derivative of electron density of rij
function f = d2Rhodr2ijkl(index,type,ri,a,rc,dfdr,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,dfdr,X,i,j,k,l,2);
```

```
end

% First derivative of Atomic Electron Density with respect to ri
function f = dfdri(index,type,ri,a,rc,dfdr,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,dfdr,X,i,j,k,l,1);
end

%  First derivative of Atomic Electron Density with respect to
ri,rj,rk and rl
function f = dfdrijkl(index,type,ri,a,rc,dfdr,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,dfdr,X,i,j,k,l,3);
end

%  Second derivative of Atomic Electron Density with respect to
ri,rj,rk and rl
function f = d2fdr2ijkl(index,type,ri,a,rc,d2fdr2,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,d2fdr2,X,i,j,k,l,2);
end

% First derivative of pair potential with respect to ri
function f = dphidri(index,type,ri,a,rc,dphidr,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,dphidr,X,i,j,k,l,1);
end

% First derivative of pair potential with respect to ri,rj,rk
and rl
function f = dPhidrijkl(index,type,ri,a,rc,dphidr,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,dphidr,X,i,j,k,l,3);
end

% Second derivative of pair potential with respect to ri,rj,rk
and rl
function f = d2Phidr2ijkl(index,type,ri,a,rc,d2phidr2,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,d2phidr2,X,i,j,k,l,2);
end

% Function Used in the calculations of Elastic Constants at
equlibrum
function f = Vij(index,type,ri,a,rc,dfdr,X,i,j,k,l)
    f = delta(i,j) *
    delta(k,l)*dfdri(index,type,ri,a,rc,dfdr,X,i,j,k,l);
end

% Function Used in the calculations of Elastic Constants at
equlibrum
function f = Wijkl(index,type,ri,a,rc,dfdr,d2fdr2,X,i,j,k,l)
    f = d2fdr2ijkl(index,type,ri,a,rc,d2fdr2,X,i,j,k,l) - ...
```

```
        dfdrijkl(index,type,ri,a,rc,dfdr,X,i,j,k,l)+ ...
        delta(i,l) * dfdri(index,type,ri,a,rc,dfdr,X,0,0,k,l);
end

% Function Used in the calculations of Elastic Constants at
equlibrum
function f = Bijkl(index,type,ri,a,rc,dphidr,d2phidr2,X,i,j,k,l)
    f = d2Phidr2ijkl(index,type,ri,a,rc,d2phidr2,X,i,j,k,l) -
    ...
    dPhidrijkl(index,type,ri,a,rc,dphidr,X,i,j,k,l)+ ...
    delta(i,l) * dphidri(index,type,ri,a,rc,dphidr,X,0,0,k,l);
    f = 0.5 * f;
end

%*************************************************************
%   Park Hijazi Pd EAM Potential and Derivatives
%*************************************************************
% Pd Embedding Energy Function as a function of electron density
function f = F_Pd(rho,X)
    f    = 0;
    Fe   = X(1);
    rhoe = X(2);
    n    = X(3);
    if rho > 0
        f = -Fe*(1-n*log(rho/rhoe))*(rho/rhoe)^n;
    end
end

% First derivative of Pd Embedding Energy Function with respect
to electron density
function f = dFdrho_Pd(rho,X)
    f    = 0;
    Fe   = X(1);
    rhoe = X(2);
    n    = X(3);
    if rho > 0
        F = F_Pd(rho,X);
        f = (Fe*n/rho)*(rho/rhoe)^n+(n/rhoe)*F*(rho/rhoe)^(-1);
    end
end

% Second derivative of Pd Embedding Energy Function with respect
to electron density
function f = d2Fdrho2_Pd(rho,X)
    f    = 0;
    Fe   = X(1);
    rhoe = X(2);
```

```
    n    = X(3);
    if rho > 0
        F  = F_Pd(rho,X);
        dF = dFdrho_Pd(rho,X);
        f  = -
        (n/rho^2)*Fe*(rho/rhoe)^n+(n^2/rho)*(1/rhoe)*Fe*(rho/rhoe
        )^(n-1)+...
       (n/rhoe)*dF*(rho/rhoe)^(-1)-(n/rhoe^2)*F*(rho/rhoe)^(-2);
    end
end

% First derivative of Pd Embedding Energy Function with respect
to rij
function f = dFdr(dFdrho,rho,dRhodr,X)
    f = dFdrho(rho,X) * dRhodr;
end

% Second derivative of Pd Embedding Energy Function with respect
to rij
function f = d2Fdr2(dFdrho,d2Fdrho2,rho,dRhodr,d2Rhodr2,X)
    f = d2Fdrho2(rho,X) * dRhodr^2  + dFdrho(rho,X) * d2Rhodr2;
end

% First derivative of Pd Embedding Energy Function with respect
to a
function f = dFda(dFdrho,rho,dRhoda,X)
    f = dFdrho(rho,X) * dRhoda;
end

% Second derivative of Pd Embedding Energy Function with respect
to a
function f = d2Fda2(dFdrho,d2Fdrho2,rho,dRhoda,d2Rhoda2,X)
    f = d2Fdrho2(rho,X) * dRhoda^2  + dFdrho(rho,X) * d2Rhoda2;
end

% Pd Atomic Electron Density as a function of interatomic
distance rij
function f = f_PdPd(rij,X)
    fe = X(1);
    Xi = X(2);
    re = X(3);
    f = fe*exp(-Xi*(rij-re));
end

% First derivative Pd Atomic Electron Density function with
respect to rij
function f = dfdr_PdPd(rij,X)
```

```
    fe = X(1);
    Xi = X(2);
    re = X(3);
    f = -fe*Xi*exp(-Xi*(rij-re));
end

%  First derivative Pd Atomic Electron Density function with
respect to rij
function f = d2fdr2_PdPd(rij,X)
    fe = X(1);
    Xi = X(2);
    re = X(3);
    f  = fe*Xi^2*exp(-Xi*(rij-re));
end

% Pd-Pd pair potential as a function of interatomic distance rij
function f = phi_PdPd(rij,X)
    phie = X(1);
    S    = X(2);
    B    = X(3);
    re   = X(4);
    f    = -phie*(1+S*(rij/re-1))*exp(-B*(rij/re-1));
end

% First derivative of Pd-Pd pair potential with respect to rij
function f = dphidr_PdPd(r,X)
    phie = X(1);
    S    = X(2);
    B    = X(3);
    re   = X(4);
    f    = (-1/re)*(phie*S*exp(-B*(r/re-1))+B*phi_PdPd(r,X));
end

% Second derivative of Pd-Pd pair potential with respect to rij
function f = d2phidr2_PdPd(r,X)
    phie = X(1);
    S    = X(2);
    B    = X(3);
    re   = X(4);
    f    = (B/re)*((phie*S/re)*exp(-B*(r/re-1))-
    dphidr_PdPd(r,X));
end

%*************************************************************
% Hydrogen EAM Potential and Derivatives
%*************************************************************
```

```
% H-H unnormalized pair potential as a function of interatomic
distance rij
function f = phi_HH_u(rij,X)
    DHH     = X(1);
    alphaHH = X(2);
    betaHH  = X(3);
    r0HH    = X(4);
    f = DHH*(betaHH*exp(-alphaHH*(rij-r0HH)) - ...
    alphaHH*exp(-betaHH*(rij-r0HH)));
end

% First derivative of H-H unnormalized pair potential with
respect to rij
function f = dphidr_HH_u(r,X)
    DHH     = X(1);
    alphaHH = X(2);
    betaHH  = X(3);
    r0HH    = X(4);
    f = DHH*( -alphaHH * betaHH*exp(-alphaHH*(r-r0HH)) +...
    betaHH*alphaHH*exp(-betaHH*(r-r0HH)) );
end

% Secon derivative of H-H unnormalized pair potential with
respect to rij
function f = d2phidr2_HH_u(r,X)
    DHH     = X(1);
    alphaHH = X(2);
    betaHH  = X(3);
    r0HH    = X(4);
    f = DHH*( (alphaHH^2) * betaHH*exp(-alphaHH*(r-r0HH)) - ...
    (betaHH^2)*alphaHH*exp(-betaHH*(r-r0HH)) );
end

% H-H normalized pair potential as a function of interatomic
distance rij
function f = phi_HH(rij,X)
    X2_HH = X(1:1:4);
    X0_HH = X(5:1:6);
    X1_HH = X(7:1:11);
    rho0H = X(11);
    f = phi_HH_u(rij,X2_HH) + 2 * dFdrho_H_u(rho0H,X1_HH) *
    f_HH(rij,X0_HH);
end

% First derivative of H-H normalized pair potential with respect
to rij
function f = dphidr_HH(rij,X)
```

```
    X2_HH = X(1:1:4);
    X0_HH = X(5:1:6);
    X1_HH = X(7:1:11);
    rho0H = X(11);
    f = dphidr_HH_u(rij,X2_HH) + 2 * dFdrho_H_u(rho0H,X1_HH) *
    ...
    dfdr_HH(rij,X0_HH);
end

% Second derivative of H-H normalized pair potential with
respect to rij
function f = d2phidr2_HH(rij,X)
    X2_HH = X(1:1:4);
    X0_HH = X(5:1:6);
    X1_HH = X(7:1:11);
    rho0H = X(11);
    f = d2phidr2_HH_u(rij,X2_HH) + 2 * dFdrho_H_u(rho0H,X1_HH) *
    ...
    d2fdr2_HH(rij,X0_HH);
end

% H Atomic Electron Density as a function of interatomic
distance
function f = f_HH(r,X)
    CH = X(1);
    DH = X(2);
    f = CH*exp(-DH*r);
end

% First derivative H Atomic Electron Density with respect to rij
function f = dfdr_HH(r,X)
    CH = X(1);
    DH = X(2);
    f = -DH*CH*exp(-DH*r);
end

% Second derivative H Atomic Electron Density with respect to
rij
function f = d2fdr2_HH(r,X)
    CH = X(1);
    DH = X(2);
    f = (DH^2)*CH*exp(-DH*r);
end

% H unormalized Embedding Energy function
function f = F_H_u(rho,X)
    aH = X(1);
```

```
    bH = X(2);
    cH = X(3);
    dH = X(4);
    EH = 0.0540638;
    %     EH = 0.055;
    f = -cH * ( (1/(2+dH))*(rho+EH)^(2+dH) - ((aH +
    bH)/(1+dH))*...
    (rho+EH)^(1+dH) + ((aH * bH)/dH) *(rho+EH)^dH );
end

% First derivative of H unnormalized Embedding function with
respect to Rho
function f = dFdrho_H_u(rho,X)
    aH = X(1);
    bH = X(2);
    cH = X(3);
    dH = X(4);
    EH = 0.0540638;
    %     EH = 0.055;
    f = -cH * ( (rho+EH)^(1+dH) - (aH + bH) * (rho+EH)^(dH) +...
    (aH * bH) *(rho+EH)^(dH-1) );
end

% Second derivative of H unnormalized Embedding function with
respect to Rho
function f = d2Fdrho2_H_u(rho,X)
    aH = X(1);
    bH = X(2);
    cH = X(3);
    dH = X(4);
    EH = 0.0540638;
    %     EH = 0.055;
    f = -cH * ( (1+dH)*(rho+EH)^(dH) - dH*(aH + bH) *
    (rho+EH)^(dH-1) + ...
    (dH-1)*(aH * bH) *(rho+EH)^(dH-2) );
end

% H normalized Embedding Energy function
function f = F_H(rho,X)
    X1_HH = X(7:1:10);
    rho0H = X(11);
    f = F_H_u(rho,X1_HH) - dFdrho_H_u(rho0H,X1_HH) * rho;
end

% First derivative of H normalized Embedding function with
respect to Rho
function f = dFdrho_H(rho,X)
```

```
    X1_HH = X(7:1:10);
    rho0H = X(11);
    f = dFdrho_H_u(rho,X1_HH) - dFdrho_H_u(rho0H,X1_HH);
end

% Second derivative of H normalized Embedding function with
respect to Rho
function f = d2Fdrho2_H(rho,X)
    X1_HH = X(7:1:10);
    f = d2Fdrho2_H_u(rho,X1_HH);
end

% Pd-H pair potential as a function of interatomic destance rij
function f = phi_PdH(r,X)
    DPdH      = X(1);
    alphaPdH  = X(2);
    betaPdH   = X(3);
    r0PdH     = X(4);
    f = DPdH*( betaPdH*exp(-alphaPdH*(r-r0PdH)) - ...
    alphaPdH*exp(-betaPdH*(r-r0PdH)) );
end

% First derivative of Pd-H pair potential with respect to
interatomic
% destance rij

function f = dphidr_PdH(r,X)
    DPdH      = X(1);
    alphaPdH  = X(2);
    betaPdH   = X(3);
    r0PdH     = X(4);
    f = DPdH*( -alphaPdH*betaPdH*exp(-alphaPdH*(r-r0PdH)) +...
    betaPdH*alphaPdH*exp(-betaPdH*(r-r0PdH)) );
end

% Second derivative of Pd-H pair potential with respect to
interatomic
% destance rij
function f = d2phidr2_PdH(r,X)
    DPdH      = X(1);
    alphaPdH  = X(2);
    betaPdH   = X(3);
    r0PdH     = X(4);
    f = DPdH*( (alphaPdH^2)*betaPdH*exp(-alphaPdH*(r-r0PdH)) -
    ...
    (betaPdH^2)*alphaPdH*exp(-betaPdH*(r-r0PdH)) );
end
```

```
%*****************************************************************
% Summation function
%*****************************************************************
function f = Sumfunc(index,type,ri,a,rc,func,X)
    f = 0;
    cl1 = (a/2)*ri(index,2);
    cl2 = (a/2)*ri(index,3);
    cl3 = (a/2)*ri(index,4);
    n = length(ri(:,1));
    for i = 1:n
        id = ri(i,1);
        if i ~= index && id == type
            l1 = (a/2)*ri(i,2);
            l2 = (a/2)*ri(i,3);
            l3 = (a/2)*ri(i,4);
            rij = sqrt((l1-cl1)^2 + (l2-cl2)^2 + (l3-cl3)^2);
            if rij <= rc
                f = f + func(rij,X);
            end
        end
    end
end

function f = SumfuncMP(index,type,ri,a,rc,func,X,p)
    f = 0;
    cl1 = (a/2)*ri(index,2);
    cl2 = (a/2)*ri(index,3);
    cl3 = (a/2)*ri(index,4);
    n = length(ri(:,1));
    for i = 1:n
        id = ri(i,1);
        if i ~= index && id == type
            l1 = (a/2)*ri(i,2);
            l2 = (a/2)*ri(i,3);
            l3 = (a/2)*ri(i,4);
            rij = sqrt((l1-cl1)^2 + (l2-cl2)^2 + (l3-cl3)^2);
            if rij <= rc
                f = f + func(rij,X) * (rij/a)^p;
            end
        end
    end
end

function f = SumfuncM(index,type,ri,a,rc,func,X)
    f = 0;
    cl1 = (a/2)*ri(index,2);
```

```
    cl2 = (a/2)*ri(index,3);
    cl3 = (a/2)*ri(index,4);
    n = length(ri(:,1));
    for i = 1:n
        id = ri(i,1);
        if i ~= index && id == type
            l1 = (a/2)*ri(i,2);
            l2 = (a/2)*ri(i,3);
            l3 = (a/2)*ri(i,4);
            rij = sqrt((l1-cl1)^2 + (l2-cl2)^2 + (l3-cl3)^2);
            if rij <= rc
                f = f + func(rij,X) * rij/a;
            end
        end
    end
end

function f = SumfuncM2(index,type,ri,a,rc,func,X)
    f = 0;
    cl1 = (a/2)*ri(index,2);
    cl2 = (a/2)*ri(index,3);
    cl3 = (a/2)*ri(index,4);
    n = length(ri(:,1));
    for i = 1:n
        id = ri(i,1);
        if i ~= index && id == type
            l1 = (a/2)*ri(i,2);
            l2 = (a/2)*ri(i,3);
            l3 = (a/2)*ri(i,4);
            rij = sqrt((l1-cl1)^2 + (l2-cl2)^2 + (l3-cl3)^2);
            if rij <= rc
                f = f + func(rij,X) * (rij/a)^2;
            end
        end
    end
end

function f = Sumfuncrijkl(index,type,r,a,rc,func,X,i,j,k,l,p)
    f = 0;
    ri = 1;
    rj = 1;
    rk = 1;
    rl = 1;
    cl(1) = (a/2)*r(index,2);
    cl(2) = (a/2)*r(index,3);
    cl(3) = (a/2)*r(index,4);
    n = length(r(:,1));
```

```
    for ii = 1:n
        id = r(ii,1);
        if ii ~= index && id == type
            l1(1) = (a/2)*r(ii,2);
            l1(2) = (a/2)*r(ii,3);
            l1(3) = (a/2)*r(ii,4);
            rij = sqrt((l1(1)-cl(1))^2 + (l1(2)-cl(2))^2 +
            (l1(3)-cl(3))^2);
            if rij < rc
                if i ~= 0
                    ri = l1(i)-cl(i);
                end
                if j ~= 0
                    rj = l1(j)-cl(j);
                end
                if k ~= 0
                    rk = l1(k)-cl(k);
                end
                if l ~= 0
                    rl = l1(l)-cl(l);
                end
                f = f + func(rij,X) * ri*rj*rk*rl/rij^p;
            end
        end
    end
end

function f = delta(i,j)
    if(i == j)
        f = 1;
        else
        f = 0;
    end
end

%*****************************************************************
% Expermental data for FCC metals
%*****************************************************************
function [afcce,Ecfcce,c11e,c12e,c44e,Bme,Eve,abcce,Ecbcce,cpre]
= parameters(el)
    index = 1;
    elements = ['Cu';'Al';'Au';'Cu';'Ni';'Pd';'Pt';'Hi'];
    for i = 1:length(elements)
        x = elements(i,:);
        if el == x
            index = i;
        end
```

```
    end
    afcc =  [4.09;4.05;4.08;3.615;3.52;3.89;3.92;3.38];
    Ecfcc = [2.85;3.36;3.93;3.54;4.45;3.91;5.77;2.119];
    %   Elastic Constants in dyne/cm^2 10^12
    c11 =   [1.24;1.14;1.86;1.70;2.465;2.3412;3.47;0.0];
    c12 =   [0.934;0.619;1.57;1.225;1.473;1.7614;2.51;0.0];
    c44 =
    [0.461;0.316;0.42;0.758;1.247;0.7117;0.765;0.05*1.6021766208]
    ;
    cpr =   [1;1;1;1;1;1;1;0.05*1.6021766208];
    % Unrelaxed
    Evfcc = [1.1;0.866;0.9;1.3;1.70;1.54;1.60;0.0];
    B =
    [1.04;0.793;1.67;1.38;1.804;1.9547;2.83;0.5*1.6021766208];  %
    from Rayne
    abcc =  [3.32;3.3;3.18;2.87;2.89;3.16;3.21;0.0];
    Ecbcc = [1;1;1;3.49;1;1;1;1];

    afcce = afcc(index);
    Ecfcce = Ecfcc(index);
    Eve = Evfcc(index);
    c11e = c11(index);
    c12e = c12(index);
    c44e = c44(index);
    Bme = B(index);
    abcce = abcc(index);
    Ecbcce = Ecbcc(index);
    cpre   = cpr(index);
end

%*****************************************************************
% Expermental and bb-initio data for PdH
%*****************************************************************
function [afcce,Ecfcce,C44fcce,Cprfcce,Bmfcce] =
PdH_parameters(el)
    index1 = 1;
    elements = ['Pd000H';'PdH000';'PdH025';'PdH050';'PdH075';...
    'PdH100';'PdHT50';'PdHT75'];
    [n,m]  = size(elements);
    for i = 1:n
        x = elements(i,:);
        if el == x
            index1 = i;
        end
    end
    a  =  [3.38;3.89;3.9421;4.0007;4.0593;4.1179;4.0007;4.0593];
```

```
    Ec =
    [2.119;3.91;3.64024;3.48770;3.37031;3.26966;3.50000;3.50000];
    %   Elastic Constants with conv from eV/A^3 to dyne/cm^2
    10^12
    C44 = [0.05,0.7117/1.6021766208,0.459,0.42,0.382,0.3440,...
    0.3810,0.3810] * 1.6021766208;  % Pd from Rayne
    Cpr = [0.05,.2899/1.6021766208,0.055,0.135,0.164,0.146,...
    0.1630,0.1630] * 1.6021766208;  % Pd from Rayne
    Bm =  [0.50,1.9547/1.6021766208,1.05,1.05,1.05,1.05,...
    1.05,1.05] * 1.6021766208;  % Pd from Rayne, Zhou values
    added
    afcce  = a(index1);
    Ecfcce = Ec(index1);
    C44fcce  = C44(index1);
    Cprfcce  = Cpr(index1);
    Bmfcce  = Bm(index1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Pd Cutoff Transformed Functions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function f = phi_PdPd_cut(rij,X)
    global rc_1;
    global NP;
    f = phi_PdPd(rij,X) - phi_PdPd(rc_1,X) + (rc_1/NP) * (1-
    (rij/rc_1)^NP )* dphidr_PdPd(rc_1,X);
end

function f = dphidr_PdPd_cut(rij,X)
    global rc_1;
    global NP;
    f = dphidr_PdPd(rij,X) - (rij/rc_1)^(NP-1) *
    dphidr_PdPd(rc_1,X);
end

function f = d2phidr2_PdPd_cut(rij,X)
    global rc_1;
    global NP;
    f = d2phidr2_PdPd(rij,X)- ((NP-1)/rc_1) * (rij/rc_1)^(NP-2)
    * dphidr_PdPd(rc_1,X);
end

function f = f_PdPd_cut(rij,X)
    global rc_1;
    global NP;
    f = f_PdPd(rij,X) - f_PdPd(rc_1,X) + (rc_1/NP) * (1-
    (rij/rc_1)^NP) * dfdr_PdPd(rc_1,X);
```

```
end

function f = dfdr_PdPd_cut(rij,X)
    global rc_1;
    global NP;
    f = dfdr_PdPd(rij,X) - (rij/rc_1)^(NP-1) *
    dfdr_PdPd(rc_1,X);
end

function f = d2fdr2_PdPd_cut(rij,X)
    global rc_1;
    global NP;
    f = d2fdr2_PdPd(rij,X)- ((NP-1)/rc_1) * (rij/rc_1)^(NP-2) *
    dfdr_PdPd(rc_1,X);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Hydrogen Cutoff Transformed functions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function f = phi_HH_cut(rij,X)
    global rc_2;
    global NP2;
    f = phi_HH(rij,X) - phi_HH(rc_2,X) + (rc_2/NP2) * (1-
    (rij/rc_2)^NP2 )* dphidr_HH(rc_2,X);
end

function f = dphidr_HH_cut(rij,X)
    global rc_2;
    global NP2;
    f = dphidr_HH(rij,X) - (rij/rc_2)^(NP2-1) *
    dphidr_HH(rc_2,X);
end

function f = d2phidr2_HH_cut(rij,X)
    global rc_2;
    global NP2;
    f = d2phidr2_HH(rij,X)- ((NP2-1)/rc_2) * (rij/rc_2)^(NP2-2)
    * dphidr_HH(rc_2,X);
end

function f = f_HH_cut(rij,X)
    global rc_2;
    global NP2;
    f = f_HH(rij,X) - f_HH(rc_2,X) + (rc_2/NP2) * (1-
    (rij/rc_2)^NP2) * dfdr_HH(rc_2,X);
end
```

```
function f = dfdr_HH_cut(rij,X)
    global rc_2;
    global NP2;
    f = dfdr_HH(rij,X) - (rij/rc_2)^(NP2-1) * dfdr_HH(rc_2,X);
end

function f = d2fdr2_HH_cut(rij,X)
    global rc_2;
    global NP2;
    f = d2fdr2_HH(rij,X)- ((NP2-1)/rc_2) * (rij/rc_2)^(NP2-2) *
    dfdr_HH(rc_2,X);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PdH Cutoff Transformed Pair Potential
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function f = phi_PdH_cut(rij,X)
    global rc_3;
    global NP3;
    f = phi_PdH(rij,X) - phi_PdH(rc_3,X) + (rc_3/NP3) * (1-
    (rij/rc_3)^NP3 )* dphidr_PdH(rc_3,X);
end

function f = dphidr_PdH_cut(rij,X)
    global rc_3;
    global NP3;
    f = dphidr_PdH(rij,X) - (rij/rc_3)^(NP3-1) *
    dphidr_PdH(rc_3,X);
end

function f = d2phidr2_PdH_cut(rij,X)
    global rc_3;
    global NP3;
    f = d2phidr2_PdH(rij,X)- ((NP3-1)/rc_3) * (rij/rc_3)^(NP3-2)
    * dphidr_PdH(rc_3,X);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PdCu Pair Potential
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function f = phi_PdCu(rij,X)
    X0_a = X(1:3);
    X0_b = X(4:6);
    X2_a = X(7:10);
    X2_b = X(11:14);
    f_a = f_PdPd(rij,X0_a);
    f_b = f_PdPd(rij,X0_b);
```

```
    phi_a = phi_PdPd(rij,X2_a);
    phi_b = phi_PdPd(rij,X2_b);
    f = .5*( f_b*f_a^-1*phi_a + f_a*f_b^-1*phi_b );
end

function f = dphidr_PdCu(rij,X)
    X0_a = X(1:3);
    X0_b = X(4:6);
    X2_a = X(7:10);
    X2_b = X(11:14);
    f_a = f_PdPd(rij,X0_a);
    f_b = f_PdPd(rij,X0_b);
    phi_a = phi_PdPd(rij,X2_a);
    phi_b = phi_PdPd(rij,X2_b);
    dfdr_a = dfdr_PdPd(rij,X0_a);
    dfdr_b = dfdr_PdPd(rij,X0_b);
    dphidr_a = dphidr_PdPd(rij,X2_a);
    dphidr_b = dphidr_PdPd(rij,X2_b);
    f = .5*( ((-1*f_b*dfdr_a*f_a^-2 + dfdr_b*f_a^-1)*phi_a +
   (f_b*f_a^-1)*dphidr_a)...
    + ((-1*f_a*dfdr_b*f_b^-2 + dfdr_a*f_b^-1)*phi_b + (f_a*f_b^-
    1)*dphidr_b) );
end

function f = d2phidr2_PdCu(rij,X)
    X0_a = X(1:3);
    X0_b = X(4:6);
    X2_a = X(7:10);
    X2_b = X(11:14);
    f_a = f_PdPd(rij,X0_a);
    f_b = f_PdPd(rij,X0_b);
    phi_a = phi_PdPd(rij,X2_a);
    phi_b = phi_PdPd(rij,X2_b);
    dfdr_a = dfdr_PdPd(rij,X0_a);
    dfdr_b = dfdr_PdPd(rij,X0_b);
    dphidr_a = dphidr_PdPd(rij,X2_a);
    dphidr_b = dphidr_PdPd(rij,X2_b);
    d2fdr2_a = d2fdr2_PdPd(rij,X0_a);
    d2fdr2_b = d2fdr2_PdPd(rij,X0_b);
    d2phidr2_a = d2phidr2_PdPd(rij,X2_a);
    d2phidr2_b = d2phidr2_PdPd(rij,X2_b);
    Phi_ab =
    phi_ab(f_a,dfdr_a,d2fdr2_a,f_b,dfdr_b,d2fdr2_b,phi_a,dphidr_a
    ,d2phidr2_a,phi_b,dphidr_b,d2phidr2_b);
    Phi_ba =
    phi_ab(f_b,dfdr_b,d2fdr2_b,f_a,dfdr_a,d2fdr2_a,phi_b,dphidr_b
    ,d2phidr2_b,phi_a,dphidr_a,d2phidr2_a);
```

```
    f = .5*(Phi_ab + Phi_ba);
end

function f =
phi_ab(f_a,dfdr_a,d2fdr2_a,f_b,dfdr_b,d2fdr2_b,phi_a,dphidr_a,d2
phidr2_a,phi_b,dphidr_b,d2phidr2_b)
    aa = -1*(dfdr_b*dfdr_a*f_a^-2 + f_b*d2fdr2_a*f_a^-2 -
    2*f_b*(dfdr_a^2)*f_a^-3)...
        + (d2fdr2_b*f_a^-1 - dfdr_b*f_a^-2*dfdr_a);
    a = aa*phi_a + (-1*f_b*dfdr_a*f_a^-2 + dfdr_b*f_a^-
    1)*dphidr_a;
    b = dfdr_b*f_a^-1*dphidr_a - f_b*f_a^-2*dfdr_a*dphidr_a +
    f_b*f_a^-1*d2phidr2_a;
    f = a + b;
end
```

Check_Fit.m

```
function Check_Fit(x,plots1,plots2)

    warning off
    format short

    % Global variables
    global ri_PdH100_OC;
    global index_PdH100_OC_1;

    global ri_PdCuH000;
    global index_PdCuH000_Pd
    global index_PdCuH000_Cu

    global a_PdCuH025_o1;
    global a_PdCuH025_o2;
    global a_PdCuH025_T;
    global a_PdCuH050_O1_O2;
    global a_PdCuH050_O2_O2;
    global a_PdCuH050_T;
    global a_PdCuH075_O1_O2_O2;
    global a_PdCuH075_O2_O2_O2;
    global a_PdCuH075_T;
    global a_PdCuH100_O1_O2_O2_O2;
    global a_PdCuH100_T;

    global Ec_exp_PdCuH000;
    global Ec_exp_PdCuH025_O1;
    global Ec_exp_PdCuH025_O2;
```

```
global Ec_exp_PdCuH025_T;
global Ec_exp_PdCuH050_O1_O2;
global Ec_exp_PdCuH050_O2_O2;
global Ec_exp_PdCuH050_T;
global Ec_exp_PdCuH075_O1_O2_O2;
global Ec_exp_PdCuH075_O2_O2_O2;
global Ec_exp_PdCuH075_T;
global Ec_exp_PdCuH100_O1_O2_O2_O2;
global Ec_exp_PdCuH100_T;

global a_PdCuH025_TE1
global a_PdCuH050_TE3_TE4
global a_PdCuH050_TE1_TE8
global a_PdCuH075_TE1_TE2_TE3
global a_PdCuH075_TE1_TE2_TE8
global a_PdCuH075_TE5_TE6_TE7
global a_PdCuH100_TE1_TE2_TE3_TE4
global a_PdCuH100_TE1_TE2_TE7_TE8
global a_PdCuH100_TE1_TE5_TE6_TE7

global Ec_exp_PdCuH025_TE1
global Ec_exp_PdCuH050_TE3_TE4
global Ec_exp_PdCuH050_TE1_TE8
global Ec_exp_PdCuH075_TE1_TE2_TE3
global Ec_exp_PdCuH075_TE1_TE2_TE8
global Ec_exp_PdCuH075_TE5_TE6_TE7
global Ec_exp_PdCuH100_TE1_TE2_TE3_TE4
global Ec_exp_PdCuH100_TE1_TE2_TE7_TE8
global Ec_exp_PdCuH100_TE1_TE5_TE6_TE7

global ri_PdCuH025_o1;
global index_PdCuH025_o1_Pd
global index_PdCuH025_o1_Cu
global index_PdCuH025_o1_H

global ri_PdCuH025_o2;
global index_PdCuH025_o2_Pd
global index_PdCuH025_o2_Cu
global index_PdCuH025_o2_H

global ri_PdCuH025_T_25_25_25;
global index_PdCuH025_T_Pd_25_25_25
global index_PdCuH025_T_Cu_25_25_25
global index_PdCuH025_T_H_25_25_25

global ri_PdCuH025_T_75_75_75;
global index_PdCuH025_T_Pd_75_75_75
```

```
global index_PdCuH025_T_Cu_75_75_75
global index_PdCuH025_T_H_75_75_75

global ri_PdCuH025_T_25_75_75;
global index_PdCuH025_T_Pd_25_75_75
global index_PdCuH025_T_Cu_25_75_75
global index_PdCuH025_T_H_25_75_75

global ri_PdCuH050_O1_O2;
global index_PdCuH050_O1_O2_Pd
global index_PdCuH050_O1_O2_Cu
global index_PdCuH050_O1_O2_H_O1
global index_PdCuH050_O1_O2_H_O2

global ri_PdCuH050_O2_O2;
global index_PdCuH050_O2_O2_Pd
global index_PdCuH050_O2_O2_Cu
global index_PdCuH050_O2_O2_H

global ri_PdCuH075_O1_O2_O2;
global index_PdCuH075_O1_O2_O2_Pd
global index_PdCuH075_O1_O2_O2_Cu
global index_PdCuH075_O1_O2_O2_H_O1
global index_PdCuH075_O1_O2_O2_H_O2

global ri_PdCuH075_O2_O2_O2;
global index_PdCuH075_O2_O2_O2_Pd
global index_PdCuH075_O2_O2_O2_Cu
global index_PdCuH075_O2_O2_O2_H

global ri_PdCuH075_T;
global index_PdCuH075_T_Pd
global index_PdCuH075_T_Cu
global index_PdCuH075_T_H

global ri_PdCuH100_O1_O2_O2_O2;
global index_PdCuH100_O1_O2_O2_O2_Pd
global index_PdCuH100_O1_O2_O2_O2_Cu
global index_PdCuH100_O1_O2_O2_O2_H_O1
global index_PdCuH100_O1_O2_O2_O2_H_O2

global ri_PdCuH100_T;
global index_PdCuH100_T_Pd
global index_PdCuH100_T_Cu
global index_PdCuH100_T_H

global NP
```

```
global NP2
global NP3

global x_Pd
global x_Cu

global S_Cu
global S_Pd

global x_PdH

global rc_1
global rc_2
global rc_3

% Pd Experimental data

[aFcce_Pd,EcFcce_Pd,C11e_Pd,C12e_Pd,C44e_Pd,Bme_Pd,Eve_Pd,aBc
ce_Pd,...
EcBcce_Pd] = parameters('Pd');

re = aFcce_Pd/sqrt(2);
Fe = EcFcce_Pd-Eve_Pd;
N  = 4;
V  = aFcce_Pd^3/N;
fe = EcFcce_Pd/V;

fe_S = S_Pd*EcFcce_Pd/V;

% Pd Fitting Parameters
Xi   = x_Pd(1);
phie = x_Pd(2);
S    = x_Pd(3);
B    = x_Pd(4);
n    = x_Pd(5);
rhoe = x_Pd(6);

%  Rho PdPd fitting parameters
X0_PdPd = [fe,Xi,re];
X0_PdPd_S = [fe_S,Xi,re];
%  F PdPd fitting parameters
X1_PdPd = [Fe,rhoe,n];
%  Phi PdPd fitting parametters
X2_PdPd = [phie,S,B,re];

% Cu Experimental data
```

```
[aFcce_Cu,EcFcce_Cu,C11e_Cu,C12e_Cu,C44e_Cu,Bme_Cu,Eve_Cu,aBc
ce_Cu,...
EcBcce_Cu] = parameters('Cu');

re = aFcce_Cu/sqrt(2);
Fe = EcFcce_Cu-Eve_Cu;
N  = 4;
V  = aFcce_Cu^3/N;
fe = EcFcce_Cu/V;

fe_S = S_Cu*EcFcce_Cu/V;

% Cu Fitting Parameters
Xi   = x_Cu(1);
phie = x_Cu(2);
S    = x_Cu(3);
B    = x_Cu(4);
n    = x_Cu(5);
rhoe = x_Cu(6);

%  Rho PdPd fitting parameters
X0_CuCu = [fe,Xi,re];
X0_CuCu_S = [fe_S,Xi,re];
%  F PdPd fitting parameters
X1_CuCu = [Fe,rhoe,n];
%  Phi PdPd fitting parametters
X2_CuCu = [phie,S,B,re];

X5_PdCu = [X0_PdPd_S X0_CuCu_S X2_PdPd X2_CuCu];

% Phi_HH
DHH =  x_PdH(1);
aHH =  x_PdH(2);
bHH =  x_PdH(3);
% f_HH
CH = x_PdH(4);
DH = x_PdH(5);
% Phi_PdH
DPdH = x_PdH(6);
aPdH = x_PdH(7);
bPdH = x_PdH(8);
% F_H
aH = x_PdH(9);
bH = x_PdH(10);
cH = x_PdH(11);
dH = x_PdH(12);
%
```

```
r0PdH = x_PdH(13);
r0HH =  x_PdH(14);

% f_HH
X0_HH  = [CH,DH];

% rho0H calculation
rho0H = 4.903820;

% F HH fitting parameters
X1_HH  = [aH,bH,cH,dH,rho0H];
% Phi HH fitting parameters
X2_HH  = [DHH,aHH,bHH,r0HH];
X22_HH = [X2_HH,X0_HH,X1_HH];
% Phi PdH fitting parameters
X4_PdH = [DPdH,aPdH,bPdH,r0PdH];

% Phi_CuH
DCuH = x(1);
aCuH = x(2);
bCuH = x(3);
r0CuH = x(4);

% Phi CuH fitting parameters
X4_CuH = [DCuH,aCuH,bCuH,r0CuH];

index_Pd = index_PdCuH000_Pd;
index_Cu = index_PdCuH000_Cu;
a = 3.821;

ri    = ri_PdCuH000;

XX = 1;
YY = 1;
ZZ = 1;

xx = 0.75;
yy = 0.25;

% Pd Central
Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
    @f_PdPd,X0_PdPd,...
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdCu,X5_PdCu,rc_1);
% Cu Central
```

```
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1);

    Ec_Pd075Cu025H000   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12 )/(xx
    + yy)

%*************************************************************
% For Pd075Cu025H025_O1
%*************************************************************
    ri = ri_PdCuH025_o1;
    index_Pd = index_PdCuH025_o1_Pd;
    index_Cu = index_PdCuH025_o1_Cu;
    index_H  = index_PdCuH025_o1_H;
    a = a_PdCuH025_o1;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    % H Central
```

```
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H025_O1   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12 +
    zz * Ec_H_12)/(xx + yy + zz)

%*****************************************************************
% For Pd075Cu025H025_O2
%*****************************************************************
    ri = ri_PdCuH025_o2;
    index_Pd = index_PdCuH025_o2_Pd;
    index_Cu = index_PdCuH025_o2_Cu;
    index_H  = index_PdCuH025_o2_H;
    a = a_PdCuH025_o2;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);
```

```
    % H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H025_O2   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12 +
    zz * Ec_H_12)/(xx + yy + zz)

%*************************************************************
% For Pd075Cu025H025_TE1
%*************************************************************
    ri = ri_PdCuH025_T_25_25_25;
    index_Pd = index_PdCuH025_T_Pd_25_25_25;
    index_Cu = index_PdCuH025_T_Cu_25_25_25;
    index_H  = index_PdCuH025_T_H_25_25_25;

    a = a_PdCuH025_T;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
```

```
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H025_TE1   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12 +
    zz * Ec_H_12)/(xx + yy + zz)

%*****************************************************************
% For Pd075Cu025H025_TE8
%*****************************************************************
    ri = ri_PdCuH025_T_75_75_75;
    index_Pd = index_PdCuH025_T_Pd_75_75_75;
    index_Cu = index_PdCuH025_T_Cu_75_75_75;
    index_H  = index_PdCuH025_T_H_75_75_75;

    a = a_PdCuH025_T;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
```

```
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H025_TE8   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12 +
    zz * Ec_H_12)/(xx + yy + zz)

%*****************************************************************
% For Pd075Cu025H025_TE6
%*****************************************************************
    ri = ri_PdCuH025_T_25_75_75;
    index_Pd = index_PdCuH025_T_Pd_25_75_75;
    index_Cu = index_PdCuH025_T_Cu_25_75_75;
    index_H  = index_PdCuH025_T_H_25_75_75;

    a = a_PdCuH025_T;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
```

```
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H025_TE6   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12 +
    zz * Ec_H_12)/(xx + yy + zz)

%*****************************************************************
% For Pd075Cu025H025_O1_O2
%*****************************************************************
    ri = ri_PdCuH050_O1_O2;
    index_Pd = index_PdCuH050_O1_O2_Pd;
    index_Cu = index_PdCuH050_O1_O2_Cu;
    index_H  = index_PdCuH050_O1_O2_H_O1;
    % index_H  = index_PdCuH050_O1_O2_H_O2;
    a = a_PdCuH050_O1_O2;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.50;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
```

```
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    % O1   H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H050_O1_O2_O1central   = ( xx * Ec_Pd_12  + yy
    * Ec_Cu_12 + zz * Ec_H_12)/(xx + yy + zz)

%*****************************************************************
% For Pd075Cu025H025_O1_O2
%*****************************************************************
    ri = ri_PdCuH050_O1_O2;
    index_Pd = index_PdCuH050_O1_O2_Pd;
    index_Cu = index_PdCuH050_O1_O2_Cu;
    % index_H  = index_PdCuH050_O1_O2_H_O1;
    index_H  = index_PdCuH050_O1_O2_H_O2;

    a = a_PdCuH050_O1_O2;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.50;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
```

```
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    % O2   H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H050_O1_O2_O2central   = ( xx * Ec_Pd_12  + yy
    * Ec_Cu_12 + zz * Ec_H_12)/(xx + yy + zz)

%*****************************************************************
% For Pd075Cu025H025_O1_O2
%*****************************************************************
    ri = ri_PdCuH050_O1_O2;
    index_Pd = index_PdCuH050_O1_O2_Pd;
    index_Cu = index_PdCuH050_O1_O2_Cu;
    index_H_O1  = index_PdCuH050_O1_O2_H_O1;
    index_H_O2  = index_PdCuH050_O1_O2_H_O2;

    a = a_PdCuH050_O1_O2;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
```

```
zz = 0.25;
zzz = 0.25;

% Pd Central
Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
    @f_PdPd,X0_PdPd,...
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_PdH,rc_1);
% Cu Central
Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
    @f_PdPd,X0_CuCu,...
    @F_Pd,X1_CuCu,...
    @phi_PdPd,X2_CuCu,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_12 =    Ec3(ri,a,index_H_O1,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_22 =    Ec3(ri,a,index_H_O2,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

Ec_Pd075Cu025H050_O1_O2   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12
+ zz * Ec_H_12 + zzz * Ec_H_22)/(xx + yy + zz + zzz)

%*************************************************************
% For Pd075Cu025H025_O2_O2
```

```
%*************************************************************
    ri = ri_PdCuH050_O2_O2;
    index_Pd = index_PdCuH050_O2_O2_Pd;
    index_Cu = index_PdCuH050_O2_O2_Cu;
    index_H  = index_PdCuH050_O2_O2_H;

    a = a_PdCuH050_O2_O2;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.50;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H050_O2_O2   = ( xx * Ec_Pd_12  + yy * Ec_Cu_12
    + zz * Ec_H_12)/(xx + yy + zz)
```

```
%*****************************************************************
% For Pd075Cu025H050_TE1_TE2
%*****************************************************************
    global ri_PdCuH050_TE1_TE2;
    global index_PdCuH050_TE1_TE2_Pd
    global index_PdCuH050_TE1_TE2_Cu
    global index_PdCuH050_TE1_TE2_H_TE1
    global index_PdCuH050_TE1_TE2_H_TE2

    ri = ri_PdCuH050_TE1_TE2;
    index_Pd = index_PdCuH050_TE1_TE2_Pd;
    index_Cu = index_PdCuH050_TE1_TE2_Cu;
    index_H_TE1  = index_PdCuH050_TE1_TE2_H_TE1;
    index_H_TE2  = index_PdCuH050_TE1_TE2_H_TE2;

    a = a_PdCuH050_T;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.50;
    zzz = 0.25;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    % H Central
```

```
    Ec_H_12 =    Ec3(ri,a,index_H_TE1,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H050_TE1_TE2__TE1central   = ( xx * Ec_Pd_12  +
    yy * Ec_Cu_12 + zz * Ec_H_12)/(xx + yy + zz)

    % H Central
    Ec_H_22 =    Ec3(ri,a,index_H_TE2,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H050_TE1_TE2   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zzz * Ec_H_12 + zzz * Ec_H_22)/(xx + yy + zzz +
    zzz)

%*****************************************************************
% For Pd075Cu025H050_TE1_TE8
%*****************************************************************
    global ri_PdCuH050_TE1_TE8;
    global index_PdCuH050_TE1_TE8_Pd
    global index_PdCuH050_TE1_TE8_Cu
    global index_PdCuH050_TE1_TE8_H_TE1
    global index_PdCuH050_TE1_TE8_H_TE8

    ri = ri_PdCuH050_TE1_TE8;
    index_Pd = index_PdCuH050_TE1_TE8_Pd;
    index_Cu = index_PdCuH050_TE1_TE8_Cu;
    index_H_TE1  = index_PdCuH050_TE1_TE8_H_TE1;
    index_H_TE8  = index_PdCuH050_TE1_TE8_H_TE8;

    a = a_PdCuH050_T;

    XX = 1;
    YY = 1;
    ZZ = 1;
```

```
xx = 0.75;
yy = 0.25;
zz = 0.50;
zzz = 0.25;

% Pd Central
Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
    @f_PdPd,X0_PdPd,...
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_PdH,rc_1);
% Cu Central
Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
    @f_PdPd,X0_CuCu,...
    @F_Pd,X1_CuCu,...
    @phi_PdPd,X2_CuCu,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_CuH,rc_1);

% H Central
Ec_H_12 =    Ec3(ri,a,index_H_TE1,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

Ec_Pd075Cu025H050_TE1_TE8__TE1central   = ( xx * Ec_Pd_12  +
yy * Ec_Cu_12 + zz * Ec_H_12)/(xx + yy + zz)

% H Central
Ec_H_22 =    Ec3(ri,a,index_H_TE8,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);
```

```
    Ec_Pd075Cu025H050_TE1_TE8    = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zzz * Ec_H_12+ zzz * Ec_H_22)/(xx + yy + zzz +
    zzz)

%*************************************************************
% For Pd075Cu025H050_TE3_TE4
%*************************************************************
    global ri_PdCuH050_TE3_TE4
    global index_PdCuH050_TE3_TE4_Pd
    global index_PdCuH050_TE3_TE4_Cu
    global index_PdCuH050_TE3_TE4_H_TE3
    global index_PdCuH050_TE3_TE4_H_TE4

    ri = ri_PdCuH050_TE3_TE4;
    index_Pd = index_PdCuH050_TE3_TE4_Pd;
    index_Cu = index_PdCuH050_TE3_TE4_Cu;
    index_H_TE3  = index_PdCuH050_TE3_TE4_H_TE3;
    index_H_TE4  = index_PdCuH050_TE3_TE4_H_TE4;

    a = a_PdCuH050_T;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;
    zzz = 0.25;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
```

```
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_12 =    Ec3(ri,a,index_H_TE3,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_22 =    Ec3(ri,a,index_H_TE4,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H050_TE3_TE4   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zz * Ec_H_12 + zzz * Ec_H_22)/(xx + yy + zz + zzz)

%*****************************************************************
% For Pd075Cu025H050_TE3_TE6
%*****************************************************************
    global ri_PdCuH050_TE3_TE6
    global index_PdCuH050_TE3_TE6_Pd
    global index_PdCuH050_TE3_TE6_Cu
    global index_PdCuH050_TE3_TE6_H_TE3
    global index_PdCuH050_TE3_TE6_H_TE6

    ri = ri_PdCuH050_TE3_TE6;
    index_Pd = index_PdCuH050_TE3_TE6_Pd;
    index_Cu = index_PdCuH050_TE3_TE6_Cu;
    index_H_TE3  = index_PdCuH050_TE3_TE6_H_TE3;
    index_H_TE6  = index_PdCuH050_TE3_TE6_H_TE6;

    a = a_PdCuH050_T;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
```

```
yy = 0.25;
zz = 0.25;
zzz = 0.25;

% Pd Central
Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
    @f_PdPd,X0_PdPd,...
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_PdH,rc_1);
% Cu Central
Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
    @f_PdPd,X0_CuCu,...
    @F_Pd,X1_CuCu,...
    @phi_PdPd,X2_CuCu,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_CuH,rc_1);

% H Central
Ec_H_12 =    Ec3(ri,a,index_H_TE3,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

% H Central
Ec_H_22 =    Ec3(ri,a,index_H_TE6,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

Ec_Pd075Cu025H050_TE3_TE6   = ( xx * Ec_Pd_12  + yy *
Ec_Cu_12 + zz * Ec_H_12 + zzz * Ec_H_22)/(xx + yy + zz + zzz)

%*************************************************************
```

```
% For Pd075Cu025H075_O1_O2_O2
%*************************************************************
    ri = ri_PdCuH075_O1_O2_O2;
    index_Pd = index_PdCuH075_O1_O2_O2_Pd;
    index_Cu = index_PdCuH075_O1_O2_O2_Cu;
    index_H_O1  = index_PdCuH075_O1_O2_O2_H_O1;
    index_H_O2  = index_PdCuH075_O1_O2_O2_H_O2;

    a = a_PdCuH075_O1_O2_O2;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;
    zzz = 0.50;
    zzzz = 0.75;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_12 =    Ec3(ri,a,index_H_O1,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
```

```
        @phi_PdH,X4_CuH,rc_1);

    % H Central
    Ec_H_22 =    Ec3(ri,a,index_H_O2,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H075_O1_O2_O2_O1central   = ( xx * Ec_Pd_12  +
    yy * Ec_Cu_12 + zzzz * Ec_H_12)/(xx + yy + zzzz)

    Ec_Pd075Cu025H075_O1_O2_O2_O2central   = ( xx * Ec_Pd_12  +
    yy * Ec_Cu_12 + zzzz * Ec_H_22)/(xx + yy + zzzz)

    Ec_Pd075Cu025H075_O1_O2_O2   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zz * Ec_H_12 + zzz * Ec_H_22)/(xx + yy + zz + zzz)

%*************************************************************
% For Pd075Cu025H075_O2_O2_O2
%*************************************************************
    ri = ri_PdCuH075_O2_O2_O2;
    index_Pd = index_PdCuH075_O2_O2_O2_Pd;
    index_Cu = index_PdCuH075_O2_O2_O2_Cu;
    index_H  = index_PdCuH075_O2_O2_O2_H;

    a = a_PdCuH075_O2_O2_O2;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.75;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
```

```
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H075_O2_O2_O2   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zz * Ec_H_12)/(xx + yy + zz)

%*************************************************************
% For Pd075Cu025H075_T
%*************************************************************
    ri = ri_PdCuH075_T;
    index_Pd = index_PdCuH075_T_Pd;
    index_Cu = index_PdCuH075_T_Cu;
    index_H  = index_PdCuH075_T_H;

    a = a_PdCuH075_T;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.75;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
```

```
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H075_TE5_TE6_TE7   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zz * Ec_H_12)/(xx + yy + zz)

%*****************************************************************
% For Pd075Cu025H075_TE1_TE2_TE3
%*****************************************************************
    global ri_PdCuH075_TE1_TE2_TE3
    global index_PdCuH075_TE1_TE2_TE3_Pd
    global index_PdCuH075_TE1_TE2_TE3_Cu
    global index_PdCuH075_TE1_TE2_TE3_H_TE1
    global index_PdCuH075_TE1_TE2_TE3_H_TE2
    global index_PdCuH075_TE1_TE2_TE3_H_TE3

    ri = ri_PdCuH075_TE1_TE2_TE3;
    index_Pd = index_PdCuH075_TE1_TE2_TE3_Pd;
    index_Cu = index_PdCuH075_TE1_TE2_TE3_Cu;
    index_H_TE1  = index_PdCuH075_TE1_TE2_TE3_H_TE1;
    index_H_TE2  = index_PdCuH075_TE1_TE2_TE3_H_TE2;
    index_H_TE3  = index_PdCuH075_TE1_TE2_TE3_H_TE3;

    a = a_PdCuH075_T;

    XX = 1;
```

```
YY = 1;
ZZ = 1;

xx = 0.75;
yy = 0.25;
zz = 0.25;
zzz = 0.25;
zzzz = 0.25;

% Pd Central
Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
    @f_PdPd,X0_PdPd,...
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_PdH,rc_1);
% Cu Central
Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
    @f_PdPd,X0_CuCu,...
    @F_Pd,X1_CuCu,...
    @phi_PdPd,X2_CuCu,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_12 =    Ec3(ri,a,index_H_TE1,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_22 =    Ec3(ri,a,index_H_TE2,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);
```

```
    %  H Central
    Ec_H_32 =    Ec3(ri,a,index_H_TE3,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H075_TE1_TE2_TE3   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zz * Ec_H_12 + zzz * Ec_H_22 + zzzz * Ec_H_32)/(xx
    + yy + zz + zzz + zzzz)

%*****************************************************************
% For Pd075Cu025H075_TE1_TE2_TE8
%*****************************************************************
    global ri_PdCuH075_TE1_TE2_TE8
    global index_PdCuH075_TE1_TE2_TE8_Pd
    global index_PdCuH075_TE1_TE2_TE8_Cu
    global index_PdCuH075_TE1_TE2_TE8_H_TE1
    global index_PdCuH075_TE1_TE2_TE8_H_TE2
    global index_PdCuH075_TE1_TE2_TE8_H_TE8

    ri = ri_PdCuH075_TE1_TE2_TE8;
    index_Pd = index_PdCuH075_TE1_TE2_TE8_Pd;
    index_Cu = index_PdCuH075_TE1_TE2_TE8_Cu;
    index_H_TE1  = index_PdCuH075_TE1_TE2_TE8_H_TE1;
    index_H_TE2  = index_PdCuH075_TE1_TE2_TE8_H_TE2;
    index_H_TE8  = index_PdCuH075_TE1_TE2_TE8_H_TE8;

    a = a_PdCuH075_T;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;
    zzz = 0.25;
    zzzz = 0.25;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
```

```
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_PdH,rc_1);
% Cu Central
Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
    @f_PdPd,X0_CuCu,...
    @F_Pd,X1_CuCu,...
    @phi_PdPd,X2_CuCu,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_12 =    Ec3(ri,a,index_H_TE1,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_22 =    Ec3(ri,a,index_H_TE2,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_32 =    Ec3(ri,a,index_H_TE8,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);
```

```
    Ec_Pd075Cu025H075_TE1_TE2_TE8   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zz * Ec_H_12 + zzz * Ec_H_22 + zzzz * Ec_H_32)/(xx
    + yy + zz + zzz + zzzz)

%*************************************************************
% For Pd075Cu025H100_O1_O2_O2_O2
%*************************************************************
    ri = ri_PdCuH100_O1_O2_O2_O2;
    index_Pd = index_PdCuH100_O1_O2_O2_O2_Pd;
    index_Cu = index_PdCuH100_O1_O2_O2_O2_Cu;
    index_H_O1  = index_PdCuH100_O1_O2_O2_O2_H_O1;
    index_H_O2  = index_PdCuH100_O1_O2_O2_O2_H_O2;

    a = a_PdCuH100_O1_O2_O2_O2;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 0.25;
    zzz = 0.75;
    zzzz = 1.00;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_12 =    Ec3(ri,a,index_H_O1,3,ZZ,...
        @f_HH,X0_HH,...
```

```
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_22 =    Ec3(ri,a,index_H_O2,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H100_O1_O2_O2_O2_O1central  =  ( xx * Ec_Pd_12
    + yy * Ec_Cu_12 + zzzz * Ec_H_12)/(xx + yy + zzzz)

    Ec_Pd075Cu025H100_O1_O2_O2_O2_O2central   = ( xx * Ec_Pd_12
    + yy * Ec_Cu_12 + zzzz * Ec_H_22)/(xx + yy + zzzz)

    Ec_Pd075Cu025H100_O1_O2_O2_O2   = ( xx * Ec_Pd_12  + yy *
    Ec_Cu_12 + zz * Ec_H_12 + zzz * Ec_H_22)/(xx + yy + zz + zzz)

%*****************************************************************
% For Pd075Cu025H100_T
%*****************************************************************
    ri = ri_PdCuH100_T;
    index_Pd = index_PdCuH100_T_Pd;
    index_Cu = index_PdCuH100_T_Cu;
    index_H  = index_PdCuH100_T_H;

    a = a_PdCuH100_T;

    XX = 1;
    YY = 1;
    ZZ = 1;

    xx = 0.75;
    yy = 0.25;
    zz = 1.00;

    % Pd Central
    Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
        @f_PdPd,X0_PdPd,...
```

```
        @F_Pd,X1_PdPd,...
        @phi_PdPd,X2_PdPd,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_PdH,rc_1);
    % Cu Central
    Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
        @f_PdPd,X0_CuCu,...
        @F_Pd,X1_CuCu,...
        @phi_PdPd,X2_CuCu,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdCu,X5_PdCu,rc_1,...
        3,ZZ,@f_HH,X0_HH,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_12 =    Ec3(ri,a,index_H,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H100_TE1_TE5_TE6_TE7   = ( xx * Ec_Pd_12  + yy
    * Ec_Cu_12 + zz * Ec_H_12)/(xx + yy + zz)

%*****************************************************************
% For Pd075Cu025H100_TE1_TE2_TE3_TE4
%*****************************************************************
    global ri_PdCuH100_TE1_TE2_TE3_TE4
    global index_PdCuH100_TE1_TE2_TE3_TE4_Pd
    global index_PdCuH100_TE1_TE2_TE3_TE4_Cu
    global index_PdCuH100_TE1_TE2_TE3_TE4_H_TE1
    global index_PdCuH100_TE1_TE2_TE3_TE4_H_TE2
    global index_PdCuH100_TE1_TE2_TE3_TE4_H_TE3
    global index_PdCuH100_TE1_TE2_TE3_TE4_H_TE4

    ri = ri_PdCuH100_TE1_TE2_TE3_TE4;
    index_Pd = index_PdCuH100_TE1_TE2_TE3_TE4_Pd;
    index_Cu = index_PdCuH100_TE1_TE2_TE3_TE4_Cu;
    index_H_TE1  = index_PdCuH100_TE1_TE2_TE3_TE4_H_TE1;
    index_H_TE2  = index_PdCuH100_TE1_TE2_TE3_TE4_H_TE2;
    index_H_TE3  = index_PdCuH100_TE1_TE2_TE3_TE4_H_TE3;
    index_H_TE4  = index_PdCuH100_TE1_TE2_TE3_TE4_H_TE4;
```

```
a = a_PdCuH100_T;

XX = 1;
YY = 1;
ZZ = 1;

xx = 0.75;
yy = 0.25;
zz = 0.25;

% Pd Central
Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
    @f_PdPd,X0_PdPd,...
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_PdH,rc_1);
% Cu Central
Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
    @f_PdPd,X0_CuCu,...
    @F_Pd,X1_CuCu,...
    @phi_PdPd,X2_CuCu,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_12 =    Ec3(ri,a,index_H_TE1,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_22 =    Ec3(ri,a,index_H_TE2,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
```

```
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_32 =    Ec3(ri,a,index_H_TE3,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_42 =    Ec3(ri,a,index_H_TE4,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H100_TE1_TE2_TE3_TE4   = ( xx * Ec_Pd_12  + yy
    * Ec_Cu_12 + zz * Ec_H_12 + zz * Ec_H_22 + zz * Ec_H_32 + zz
    * Ec_H_42)/(xx + yy + 4*zz)

%*************************************************************
% For Pd075Cu025H100_TE1_TE2_TE7_TE8
%*************************************************************
    global ri_PdCuH100_TE1_TE2_TE7_TE8
    global index_PdCuH100_TE1_TE2_TE7_TE8_Pd
    global index_PdCuH100_TE1_TE2_TE7_TE8_Cu
    global index_PdCuH100_TE1_TE2_TE7_TE8_H_TE1
    global index_PdCuH100_TE1_TE2_TE7_TE8_H_TE2
    global index_PdCuH100_TE1_TE2_TE7_TE8_H_TE7
    global index_PdCuH100_TE1_TE2_TE7_TE8_H_TE8

    ri = ri_PdCuH100_TE1_TE2_TE7_TE8;
    index_Pd = index_PdCuH100_TE1_TE2_TE7_TE8_Pd;
    index_Cu = index_PdCuH100_TE1_TE2_TE7_TE8_Cu;
    index_H_TE1  = index_PdCuH100_TE1_TE2_TE7_TE8_H_TE1;
    index_H_TE2  = index_PdCuH100_TE1_TE2_TE7_TE8_H_TE2;
    index_H_TE7  = index_PdCuH100_TE1_TE2_TE7_TE8_H_TE7;
    index_H_TE8  = index_PdCuH100_TE1_TE2_TE7_TE8_H_TE8;

    a = a_PdCuH100_T;
```

```
XX = 1;
YY = 1;
ZZ = 1;

xx = 0.75;
yy = 0.25;
zz = 0.25;

% Pd Central
Ec_Pd_12 =    Ec3(ri,a,index_Pd,1,XX,...
    @f_PdPd,X0_PdPd,...
    @F_Pd,X1_PdPd,...
    @phi_PdPd,X2_PdPd,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_PdH,rc_1);
% Cu Central
Ec_Cu_12 =    Ec3(ri,a,index_Cu,2,YY,...
    @f_PdPd,X0_CuCu,...
    @F_Pd,X1_CuCu,...
    @phi_PdPd,X2_CuCu,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdCu,X5_PdCu,rc_1,...
    3,ZZ,@f_HH,X0_HH,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_12 =    Ec3(ri,a,index_H_TE1,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);

%  H Central
Ec_H_22 =    Ec3(ri,a,index_H_TE2,3,ZZ,...
    @f_HH,X0_HH,...
    @F_H,X22_HH,...
    @phi_HH,X22_HH,rc_1,...
    1,XX,@f_PdPd,X0_PdPd,...
    @phi_PdH,X4_PdH,rc_1,...
    2,YY,@f_PdPd,X0_CuCu,...
    @phi_PdH,X4_CuH,rc_1);
```

```
    %  H Central
    Ec_H_32 =    Ec3(ri,a,index_H_TE7,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    %  H Central
    Ec_H_42 =    Ec3(ri,a,index_H_TE8,3,ZZ,...
        @f_HH,X0_HH,...
        @F_H,X22_HH,...
        @phi_HH,X22_HH,rc_1,...
        1,XX,@f_PdPd,X0_PdPd,...
        @phi_PdH,X4_PdH,rc_1,...
        2,YY,@f_PdPd,X0_CuCu,...
        @phi_PdH,X4_CuH,rc_1);

    Ec_Pd075Cu025H100_TE1_TE2_TE7_TE8   = ( xx * Ec_Pd_12  + yy
    * Ec_Cu_12 + zz * Ec_H_12 + zz * Ec_H_22 + zz * Ec_H_32 + zz
    * Ec_H_42)/(xx + yy + 4*zz)

%*****************************************************************
% Table
%*****************************************************************
    Model1    = [Ec_Pd075Cu025H025_O1; Ec_Pd075Cu025H025_O2;
    Ec_Pd075Cu025H025_TE1;...
    Ec_Pd075Cu025H050_O1_O2;Ec_Pd075Cu025H050_O2_O2;Ec_Pd075Cu025
    H050_TE3_TE4;...
    Ec_Pd075Cu025H050_TE1_TE8;Ec_Pd075Cu025H075_O1_O2_O2;Ec_Pd075
    Cu025H075_O2_O2_O2;...
    Ec_Pd075Cu025H075_TE1_TE2_TE3;Ec_Pd075Cu025H075_TE1_TE2_TE8;E
    c_Pd075Cu025H100_O1_O2_O2_O2;Ec_Pd075Cu025H100_TE1_TE2_TE3_TE
    4;Ec_Pd075Cu025H100_TE1_TE2_TE7_TE8];

    Siesta1 = [Ec_exp_PdCuH025_O1; Ec_exp_PdCuH025_O2;
    Ec_exp_PdCuH025_TE1;...
    Ec_exp_PdCuH050_O1_O2; Ec_exp_PdCuH050_O2_O2;
    Ec_exp_PdCuH050_TE3_TE4; Ec_exp_PdCuH050_TE1_TE8;...
    Ec_exp_PdCuH075_O1_O2_O2; Ec_exp_PdCuH075_O2_O2_O2;
    Ec_exp_PdCuH075_TE1_TE2_TE3; Ec_exp_PdCuH075_TE1_TE2_TE8;...
    Ec_exp_PdCuH100_O1_O2_O2_O2;
    Ec_exp_PdCuH100_TE1_TE2_TE3_TE4;
    Ec_exp_PdCuH100_TE1_TE2_TE7_TE8];
```

```
Rownames =
{'PdCuH025_O1','PdCuH025_O2','PdCuH025_TE1','PdCuH050_O1_O2',
...
'PdCuH050_O2_O2','PdCuH050_TE3_TE4','PdCuH050_TE1_TE8','PdCuH
075_O1_O2_O2',...
'PdCuH075_O2_O2_O2','PdCuH075_TE1_TE2_TE3','PdCuH075_TE1_TE2_
TE8','PdCuH100_O1_O2_O2_O2','PdCuH100_TE1_TE2_TE3_TE4','PdCuH
100_TE1_TE2_TE7_TE8'};

Columnames = {'Siesta';'Model'};
Results =
table(Siesta1,Model1,'VariableNames',Columnames,'RowNames',Ro
wnames)

ar = 2.1075;
fig_height = 6;
fig_width = ar*fig_height;

%  Two Body Potential
figure(1)
j = 1;
for r = 0:0.01:6
    Phi_CuH(j) = phi_PdH(r,X4_CuH);
    j = j + 1;
end
r = [0:0.01:6];
plot(r,Phi_CuH,'black-','markersize',3, 'linewidth',4)
xlabel('r (A^o)','fontsize',24)
ylabel('Two Body Potential (eV)','fontsize',24)
legend('CuH')
set(gca,'FontSize',24,'FontWeight','bold','linewidth',3)
axis([0.0,6,-1,2.5])
fig = figure(1);
fig.Units = 'inches';
fig.Position = [2 2 fig_width fig_height];

% Comparison Two Body Potential
figure(2)
j = 1;
for r = 0:0.01:6
    Phi_PdPd(j)    = phi_PdPd(r,X2_PdPd);
    Phi_CuCu(j)    = phi_PdPd(r,X2_CuCu);
    Phi_HH(j)  = phi_HH(r,X22_HH);
    Phi_PdH(j) = phi_PdH(r,X4_PdH);
    Phi_CuH(j) = phi_PdH(r,X4_CuH);
    Phi_PdCu(j) = phi_PdCu(r,X5_PdCu);
```

```
    j = j + 1;
end
r = [0:0.01:6];
plot(r,Phi_HH,'black:',r,Phi_PdH,'black-
.',r,Phi_PdPd,'black--',r,Phi_CuCu,'g:',r,Phi_CuH,'black-
',r,Phi_PdCu,'b:','markersize',3, 'linewidth',4)
xlabel('r (A^o)','fontsize',24)
ylabel('Two Body Potential (eV)','fontsize',24)
legend('HH','PdH','PdPd','CuCu','CuH','PdCu')
set(gca,'FontSize',24,'FontWeight','bold','linewidth',3)
axis([0.0,6,-1,2.5])
fig = figure(2);
fig.Units = 'inches';
fig.Position = [2 2 fig_width fig_height];

% Comparison Two Body Potential
figure(22)
plot(r,Phi_HH,'k:',r,Phi_PdH,'k-.',r,Phi_CuH,'k--
','markersize',3, 'linewidth',4)
xlabel('r (A^o)','fontsize',24)
ylabel('Two Body Potential (eV)','fontsize',24)
legend('HH','PdH','CuH')
set(gca,'FontSize',24,'FontWeight','bold','linewidth',3)
axis([0.0,6,-1,2.5])
fig = figure(22);
fig.Units = 'inches';
fig.Position = [2 2 fig_width fig_height];

% Comparison Two Body Potential
figure(23)
plot(r,Phi_PdPd,'k-',r,Phi_CuCu,'k:',r,Phi_PdCu,'k--
','markersize',3, 'linewidth',4)
xlabel('r (A^o)','fontsize',24)
ylabel('Two Body Potential (eV)','fontsize',24)
legend('PdPd','CuCu','PdCu')
set(gca,'FontSize',24,'FontWeight','bold','linewidth',3)
axis([1.5,6,-.3,1])
fig = figure(23);
fig.Units = 'inches';
fig.Position = [2 2 fig_width fig_height];

x = [1:12]';

figure (3)
x = [1:14]';
Model1    = [Ec_Pd075Cu025H025_O1; Ec_Pd075Cu025H025_O2;
Ec_Pd075Cu025H025_TE1;...
```

```
Ec_Pd075Cu025H050_O1_O2;Ec_Pd075Cu025H050_O2_O2;Ec_Pd075Cu025
H050_TE3_TE4;...

Ec_Pd075Cu025H050_TE1_TE8;Ec_Pd075Cu025H075_O1_O2_O2;Ec_Pd075
Cu025H075_O2_O2_O2;...

Ec_Pd075Cu025H075_TE1_TE2_TE3;Ec_Pd075Cu025H075_TE1_TE2_TE8;E
c_Pd075Cu025H100_O1_O2_O2_O2;Ec_Pd075Cu025H100_TE1_TE2_TE3_TE
4;Ec_Pd075Cu025H100_TE1_TE2_TE7_TE8];

    Siesta1 = [Ec_exp_PdCuH025_O1; Ec_exp_PdCuH025_O2;
Ec_exp_PdCuH025_TE1;...
        Ec_exp_PdCuH050_O1_O2; Ec_exp_PdCuH050_O2_O2;
Ec_exp_PdCuH050_TE3_TE4; Ec_exp_PdCuH050_TE1_TE8;...
        Ec_exp_PdCuH075_O1_O2_O2; Ec_exp_PdCuH075_O2_O2_O2;
Ec_exp_PdCuH075_TE1_TE2_TE3; Ec_exp_PdCuH075_TE1_TE2_TE8;...
        Ec_exp_PdCuH100_O1_O2_O2_O2;
Ec_exp_PdCuH100_TE1_TE2_TE3_TE4;
Ec_exp_PdCuH100_TE1_TE2_TE7_TE8];
plot(x,Model1,'kO',x,Siesta1,'blackd','markersize',10,
'linewidth',3)
set(gca,'FontSize',12,'FontWeight','bold','linewidth',3)
xlabel('x','fontsize',24)
ylabel('E_c (eV/atom)','fontsize',24)
legend('Model','Siesta','Location','Northwest')
legend({},'fontsize',18)
xticks([1 2 3 4 5 6 7 8 9 10 11 12 13 14])
xticklabels({'PdCuH025 O1','PdCuH025 O2','PdCuH025
TE1','PdCuH050 O1 O2',...
'PdCuH050 O2 O2','PdCuH050 TE3 TE4','PdCuH050 TE1
TE8','PdCuH075 O1 O2 O2',...
'PdCuH075 O2 O2 O2','PdCuH075 TE123','PdCuH075
TE128','PdCuH100 O1 O2 O2 O2','PdCuH100 TE1234','PdCuH100
TE1278'})
xtickangle(45)
fig = figure(3);
fig.Units = 'inches';
fig.Position = [2 2 fig_width fig_height];

x = [1:11]';

Siesta = [Ec_exp_PdCuH025_O1; Ec_exp_PdCuH025_O2;
Ec_exp_PdCuH025_T;...
Ec_exp_PdCuH050_O1_O2; Ec_exp_PdCuH050_O2_O2;
Ec_exp_PdCuH050_T;...
```

```
Ec_exp_PdCuH075_O1_O2_O2; Ec_exp_PdCuH075_O2_O2_O2;
Ec_exp_PdCuH075_T;...
Ec_exp_PdCuH100_O1_O2_O2_O2; Ec_exp_PdCuH100_T];

Fitting = [Ec_Pd075Cu025H025_O1; Ec_Pd075Cu025H025_O2; ...
Ec_Pd075Cu025H050_O1_O2;Ec_Pd075Cu025H050_O2_O2;...

Ec_Pd075Cu025H075_O1_O2_O2;Ec_Pd075Cu025H075_O2_O2_O2;...
Ec_Pd075Cu025H100_O1_O2_O2_O2]';

Model        =
[Ec_Pd075Cu025H050_TE1_TE2;Ec_Pd075Cu025H050_TE1_TE8;...
Ec_Pd075Cu025H050_TE3_TE4;Ec_Pd075Cu025H050_TE3_TE6;...

Ec_Pd075Cu025H075_TE5_TE6_TE7;Ec_Pd075Cu025H075_TE1_TE2_TE3;.
..

Ec_Pd075Cu025H075_TE1_TE2_TE8;Ec_Pd075Cu025H100_TE1_TE5_TE6_T
E7;...

Ec_Pd075Cu025H100_TE1_TE2_TE3_TE4;Ec_Pd075Cu025H100_TE1_TE2_T
E7_TE8];

Siesta = [Ec_exp_PdCuH025_O1; Ec_exp_PdCuH025_O2;
Ec_exp_PdCuH025_T;...
Ec_exp_PdCuH050_O1_O2; Ec_exp_PdCuH050_O2_O2;
Ec_exp_PdCuH050_T; Ec_exp_PdCuH050_T;...
Ec_exp_PdCuH075_O1_O2_O2; Ec_exp_PdCuH075_O2_O2_O2;
Ec_exp_PdCuH075_T;...
Ec_exp_PdCuH100_O1_O2_O2_O2; Ec_exp_PdCuH100_T];

Zhou_Experimental = [-3.396;-3.337;-3.373;-3.180;-3.147;-
3.171;-3.171;-3.035;-2.944;-3.025;-2.863;-2.919];

Zhou = [-3.49499221;-3.2892551;-3.4706926;-3.23099;-
3.086482;-3.168897;-3.168897;-3.070289;-2.9575078;-
3.031653827;-2.95657414;-3.0171908];

% PError       = abs(Model-
Experimental).*100./abs(Experimental);
Rownames =
{'PdCuH050_TE1_TE2','PdCuH050_TE1_TE8','PdCuH050_TE3_TE4',...
'PdCuH050_TE3_TE8','PdCuH075_TE5_TE6_TE7','PdCuH075_TE1_TE2_T
E3',...
'PdCuH075_TE1_TE2_TE8','PdCuH100_TE1_TE5_TE6_TE7',...
'PdCuH100_TE1_TE2_TE3_TE4','PdCuH100_TE1_TE2_TE7_TE8'};
```

```
    Columnames = {'Model'};

    Results =
    table(Model,'VariableNames',Columnames,'RowNames',Rownames);

end

%*************************************************************
function [Ec_,dEda_]=
fitProperties_1(r,a,N,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
F,dFdrho,d2Fdrho2,X1,phi_11,dphidr_11,d2phidr2_11,X2,rc_1,...
t2,x,f_22,dfdr_22,d2fdr2_22,X3,phi_12,dphidr_12,d2phidr2_12,X4,r
c_2)

    Ec_ =
    Ec(r,a,index,t1,y,f_11,X0,F,X1,phi_11,X2,rc_1,t2,x,f_22,X3,ph
    i_12,X4,rc_2);

    dEda_ =
    dEda(r,a,index,t1,y,f_11,dfdr_11,X0,dFdrho,X1,dphidr_11,X2,rc
    _1,...
    t2,x,f_22,dfdr_22,X3,dphidr_12,X4,rc_2);
end

%*************************************************************
% Property Functions Used In Parameters Fitting
%*************************************************************
function [Ec_,dEda_,C11,C12,C44,Bm_,Ev_] =
fitProperties(r,a,N,index,...
t1,y,f_11,dfdr_11,d2fdr2_11,X0,F,dFdrho,d2Fdrho2,X1,phi_11,...
dphidr_11,d2phidr2_11,X2,rc_1,t2,x,f_22,dfdr_22,d2fdr2_22,X3,...
phi_12,dphidr_12,d2phidr2_12,X4,rc_2)
    % Cohesive Energy
    Ec_ =
    Ec(r,a,index,t1,y,f_11,X0,F,X1,phi_11,X2,rc_1,t2,x,f_22,X3,..
    .
    phi_12,X4,rc_2);
    % First Derivative of Ec with respect to a
    dEda_ =
    dEda(r,a,index,t1,y,f_11,dfdr_11,X0,dFdrho,X1,dphidr_11,X2,..
    .
    rc_1,t2,x,f_22,dfdr_22,X3,dphidr_12,X4,rc_2);
    % C11 Elastic Constant
    C11 =
    Cijkl(1,1,1,1,r,a,N,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
    dFdrho,d2Fdrho2,X1,dphidr_11,d2phidr2_11,X2,rc_1,...
```

```
    t2,x,f_22,dfdr_22,d2fdr2_22,X3,dphidr_12,d2phidr2_12,X4,rc_2)
    ;
    % C12 Elastic Constant
    C12 =
    Cijkl(1,1,2,2,r,a,N,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
     dFdrho,d2Fdrho2,X1,dphidr_11,d2phidr2_11,X2,rc_1,...
     t2,x,f_22,dfdr_22,d2fdr2_22,X3,dphidr_12,d2phidr2_12,X4,rc_2
     );
    % C44 Elastic Constant
    C44 =
    Cijkl(2,3,2,3,r,a,N,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
    dFdrho,d2Fdrho2,X1,dphidr_11,d2phidr2_11,X2,rc_1,...
    t2,x,f_22,dfdr_22,d2fdr2_22,X3,dphidr_12,d2phidr2_12,X4,rc_2)
    ;
    % Bulk Modulus
    Bm_ = Bm(r,a,N,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
    dFdrho,d2Fdrho2,X1,d2phidr2_11,X2,rc_1,t2,x,f_22,dfdr_22,...
    d2fdr2_22,X3,d2phidr2_12,X4,rc_2);
    % Vacancy Formation
    % Ev_ = Ev(index,t1,a,phi_11,X2,f_11,X0,F,X1,r,rc_1);
    Ev_ = 0;
end

%*****************************************************************
% Cohesive Energy for Interstitial Solid Solution
%*****************************************************************
function f =
Ec(ri,a,index,t1,y,f_11,X0,F_11,X1,phi_11,X2,rc_1,t2,x,...
f_22,X3,phi_12,X4,rc_2)

    f = 0;
    Rho_12 = 0;
    Phi_12 = 0;

    Rho_11 = y * Rho(index,t1,ri,a,rc_1,f_11,X0);
    Phi_11 = y * Phi(index,t1,ri,a,rc_1,phi_11,X2);

    if nargin == 19
        Rho_12  = x * Rho(index,t2,ri,a,rc_2,f_22,X3);
        Phi_12  = x * Phi(index,t2,ri,a,rc_2,phi_12,X4);
    end

    Rho_1 = Rho_11 + Rho_12;
    F_1 = F_11(Rho_1,X1);
    f = F_1 + 0.5 * Phi_11 + 0.5 * Phi_12;
end
```

```
function f =
Ec3(ri,a,index,t1,y,f_11,X0,F_11,X1,phi_11,X2,rc_1,t2,x,...
f_22,X3,phi_12,X4,rc_2,t3,z,f_33,X5,phi_13,X6,rc_3)

    f = 0;
    Rho_12 = 0;
    Phi_12 = 0;
    Rho_13 = 0;
    Phi_13 = 0;

    Rho_11 = y * Rho(index,t1,ri,a,rc_1,f_11,X0);
    Phi_11 = y * Phi(index,t1,ri,a,rc_1,phi_11,X2);

    if nargin >= 19
        Rho_12  = x * Rho(index,t2,ri,a,rc_2,f_22,X3);
        Phi_12  = x * Phi(index,t2,ri,a,rc_2,phi_12,X4);
    end

    if nargin >= 26
        Rho_13  = z * Rho(index,t3,ri,a,rc_3,f_33,X5);
        Phi_13  = z * Phi(index,t3,ri,a,rc_3,phi_13,X6);
    end

    Rho_1 = Rho_11 + Rho_12 + Rho_13;
    F_1 = F_11(Rho_1,X1);
    f = F_1 + 0.5 * Phi_11 + 0.5 * Phi_12 + 0.5 * Phi_13;
end


%*****************************************************************
% Elestic Constants for an Interstitial Solid Solution
%*****************************************************************
function f =
Cijkl(i,j,k,l,r,a,N,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
dFdrho,d2Fdrho2,X1,dphidr_11,d2phidr2_11,X2,rc_1,t2,...
x,f_22,dfdr_22,d2fdr2_22,X3,dphidr_12,d2phidr2_12,X4,rc_2)

    Rho_12 = 0;
    Vij_12 = 0;
    Vkl_12 = 0;
    Wijkl_12 = 0;
    Bijkl_12 = 0;

    Rho_11   = y * Rho(index,t1,r,a,rc_1,f_11,X0);
    Vij_11   = y * Vij(index,t1,r,a,rc_1,dfdr_11,X0,i,j,0,0);
```

```
    Vkl_11   = y * Vij(index,t1,r,a,rc_1,dfdr_11,X0,0,0,k,l);
    Wijkl_11 = y *
    Wijkl(index,t1,r,a,rc_1,dfdr_11,d2fdr2_11,X0,i,j,k,l);
    Bijkl_11 = y *
    Bijkl(index,t1,r,a,rc_1,dphidr_11,d2phidr2_11,X2,i,j,k,l);

    if nargin == 31
        Rho_12   = x * Rho(index,t2,r,a,rc_2,f_22,X3);
        Vij_12   = x *
        Vij(index,t2,r,a,rc_2,dfdr_22,X3,i,j,0,0);
        Vkl_12   = x *
        Vij(index,t2,r,a,rc_2,dfdr_22,X3,0,0,k,l);
        Wijkl_12 = x *
        Wijkl(index,t2,r,a,rc_2,dfdr_22,d2fdr2_22,X3,i,j,k,l);
        Bijkl_12 = x *
        Bijkl(index,t2,r,a,rc_2,dphidr_12,d2phidr2_12,X4,i,j,k,l)
        ;
    end
    Rho_1      = Rho_11 + Rho_12;
    dFdrho_1   = dFdrho(Rho_1,X1);
    d2Fdrho2_1 = d2Fdrho2(Rho_1,X1);
    Elastic constants Cijkl
    V = (a^3)/N;
    Cijkl = (1/V)*( d2Fdrho2_1*(Vij_11 + Vij_12)*(Vkl_11 +
    Vkl_12) +...
    dFdrho_1*(Wijkl_11 + Wijkl_12) + Bijkl_11 + Bijkl_12 );
    f = Cijkl * 1.602176462;
end

%*************************************************************
% Stress for an Interstitial solid solution
%*************************************************************
function f =
S(i,j,r,a,N,index,t1,y,f_11,dfdr_11,X0,dFdrho,X1,dphidr_11,X2,rc
_1,...
t2,x,f_22,dfdr_22,X3,dphidr_12,X4,rc_2)

    Rho_12 = 0;
    Vij_12 = 0;
    Aij_12 = 0;

    Rho_11 = y * Rho(index,t1,r,a,rc_1,f_11,X0);
    Vij_11 = y * Vij(index,t1,r,a,rc_1,dfdr_11,X0,i,j,0,0);
    Aij_11 = y * Aij(index,t1,r,a,rc_1,dphidr_11,X2,i,j,0,0);
    if nargin == 24
        Rho_12 = x * Rho(index,t2,r,a,rc_2,f_22,X3);
        Vij_12 = x * Vij(index,t2,r,a,rc_2,dfdr_22,X3,i,j,0,0);
```

```
        Aij_12 = x *
        Aij(index,t2,r,a,rc_2,dphidr_12,X4,i,j,0,0);
    end
    Rho_1     = Rho_11 + Rho_12;
    dFdrho_1  = dFdrho(Rho_1,X1);
    V = (a^3)/N;
    f = (1/V) * (Aij_11 + Aij_12 + dFdrho_1 * (Vij_11 +
    Vij_12));
end

%*************************************************************
% First derivative of Cohesive Energy with respect to a for an
% Interstitial solid solution
%*************************************************************
function  f =
dEda(ri,a,index,t1,y,f_11,dfdr_11,X0,dFdrho,X1,dphidr_11,X2,rc_1
,...
t2,x,f_22,dfdr_22,X3,dphidr_12,X4,rc_2)

    Rho_12    = 0;
    dRhoda_12 = 0;
    dPhida_12 = 0;

    Rho_11    = y * Rho(index,t1,ri,a,rc_1,f_11,X0);
    dRhoda_11 = y * dRhoda(index,t1,ri,a,rc_1,dfdr_11,X0);
    dPhida_11 = y * dPhida(index,t1,ri,a,rc_1,dphidr_11,X2);

    if nargin == 21
        Rho_12    = x * Rho(index,t2,ri,a,rc_2,f_22,X3);
        dRhoda_12 = x * dRhoda(index,t2,ri,a,rc_2,dfdr_22,X3);
        dPhida_12 = x * dPhida(index,t2,ri,a,rc_2,dphidr_12,X4);
    end

    Rho_1     = Rho_11    + Rho_12;
    dRhoda_1  = dRhoda_11 + dRhoda_12;
    dFda_1 = dFda(dFdrho,Rho_1,dRhoda_1,X1);
    f = dFda_1 + 0.5 * dPhida_11 + 0.5 * dPhida_12;
end

%*************************************************************
% Second derivative of Cohesive Energy with respect to a for
Interstitial
% Solid Solution
%*************************************************************
function  f =
d2Eda2(ri,a,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
dFdrho,d2Fdrho2,X1,d2phidr2_11,X2,rc_1,t2,x,...
```

```
f_22,dfdr_22,d2fdr2_22,X3,d2phidr2_12,X4,rc_2)

    Rho_12      = 0;
    dRhoda_12   = 0;
    d2Rhoda2_12 = 0;
    d2Phida2_12 = 0;

    Rho_11      = y * Rho(index,t1,ri,a,rc_1,f_11,X0);
    dRhoda_11   = y * dRhoda(index,t1,ri,a,rc_1,dfdr_11,X0);
    d2Rhoda2_11 = y * d2Rhoda2(index,t1,ri,a,rc_1,d2fdr2_11,X0);
    d2Phida2_11 = y *
    d2Phida2(index,t1,ri,a,rc_1,d2phidr2_11,X2);

    if nargin == 24
        Rho_12      = x * Rho(index,t2,ri,a,rc_2,f_22,X3);
        dRhoda_12   = x * dRhoda(index,t2,ri,a,rc_2,dfdr_22,X3);
        d2Rhoda2_12 = x *
        d2Rhoda2(index,t2,ri,a,rc_2,d2fdr2_22,X3);
        d2Phida2_12 = x *
        d2Phida2(index,t2,ri,a,rc_2,d2phidr2_12,X4);
    end
    Rho_1       = Rho_11 + Rho_12;
    dRhoda_1    = dRhoda_11 + dRhoda_12;
    d2Rhoda2_1  = d2Rhoda2_11 + d2Rhoda2_12;
    d2Fda2_1 =
    d2Fda2(dFdrho,d2Fdrho2,Rho_1,dRhoda_1,d2Rhoda2_1,X1);
    f = d2Fda2_1 + 0.5 * d2Phida2_11 + 0.5 * d2Phida2_12;
end

%*****************************************************************
% Bulk modulus
%*****************************************************************
function f = Bm(ri,a,N,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
    dFdrho,d2Fdrho2,X1,d2phidr2_11,X2,rc_1,t2,x,f_22,...
    dfdr_22,d2fdr2_22,X3,d2phidr2_12,X4,rc_2)
    d2Eda2_ = 0;

    if nargin == 16
        d2Eda2_ =
        d2Eda2(ri,a,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
        dFdrho,d2Fdrho2,X1,d2phidr2_11,X2,rc_1);
    end
    if nargin == 25
        d2Eda2_ =
        d2Eda2(ri,a,index,t1,y,f_11,dfdr_11,d2fdr2_11,X0,...
        dFdrho,d2Fdrho2,X1,d2phidr2_11,X2,rc_1,t2,x,f_22,...
        dfdr_22,d2fdr2_22,X3,d2phidr2_12,X4,rc_2);
```

```
    end
    V = a^3/N;
    f = ((a)^2/(9.0*V))*d2Eda2_;
    f = f * 1.602176462;
end

%*****************************************************************
% Vacancy formation Energy
%*****************************************************************
function f = Ev(index,type,a,phi,X0,f,X1,F,X2,rij,rc)
    sum0 = 0;
    sum1 = 0;
    sum2 = 0;
    rho = Rho(index,type,rij,a,rc,f,X1);
    n = length(rij(:,1));
    for i = 1:n
        id = rij(i,1);
        if i ~= index && id == type
            l1 = (a/2)*rij(i,2);
            l2 = (a/2)*rij(i,3);
            l3 = (a/2)*rij(i,4);
            ri = sqrt(l1^2+l2^2+l3^2);
            if ri <= rc
                q0 = phi(ri,X0);
                f1 = f(ri,X1);
                sum0 = sum0 + q0;
                sum1 = sum1 + F(rho,X2);
                sum2 = sum2 + F(rho-f1,X2);
            end
        end
    end
    f = -0.5 * sum0 + sum2 - sum1;
end

%*****************************************************************
% The Embedding Atom Model EAM
%*****************************************************************
% Total pair potential energy for a central atom as function of
interatomic
% distance
% index: array index of central atom
% type : Id of atom type
% ri   : atoms position in Lattice Crystal Structure
% a    : the lattice constant
% rc   : the cuttof radius
% phi  : the pair potential function
% X    : array of pair potential parameters
```

```
function f = Phi(index,type,ri,a,rc,phi,X)
    f = Sumfunc(index,type,ri,a,rc,phi,X);
end

% Total first derivative of pair potential with respect to
interatomic
% distance rij
function f = dPhidr(index,type,ri,a,rc,dphidr,X)
    f = Sumfunc(index,type,ri,a,rc,dphidr,X);
end

% Total second derivative of pair potential with respect to
interatomic
% distance rij
function f = d2Phidr2(index,type,ri,a,rc,d2phidr2,X)
    f = Sumfunc(index,type,ri,a,rc,d2phidr2,X);
end

% Total first derivative of pair potential with respect to a
function f = dPhida(index,type,ri,a,rc,dphidr,X)
    f = SumfuncM(index,type,ri,a,rc,dphidr,X);
end

% Total second derivative of pair potential with respect to a
function f = d2Phida2(index,type,ri,a,rc,d2phidr2,X)
    f = SumfuncM2(index,type,ri,a,rc,d2phidr2,X);
end

% Total Electron Density for a central atom as function of
interatomic
% distance
% index: array index of central atom
% type : Id of atom type
% ri   : atoms position in Lattice Crystal Structure
% a    : the lattice constant
% rc   : the cuttof radius
% f    : the atomic density function
% X    : array of atomic density function parameters
function f = Rho(index,type,ri,a,rc,f,X)
    f = Sumfunc(index,type,ri,a,rc,f,X);
end

% Total first derivative of electron density with respect to
interatomic
% distance rij
function f = dRhodr(index,type,ri,a,rc,dfdr,X)
    f = Sumfunc(index,type,ri,a,rc,dfdr,X);
```

```
end

% Total second derivative of electron density with respect to
interatomic
% distance rij
function f = d2Rhodr2(index,type,ri,a,rc,d2fdr2,X)
    f = Sumfunc(index,type,ri,a,d2fdr2,X);
end

% Total first derivative of electron density with respect a
function f = dRhoda(index,type,ri,a,rc,dfdr,X)
    f = SumfuncM(index,type,ri,a,rc,dfdr,X);
end

% Total second derivative of electron density with respect a
function f = d2Rhoda2(index,type,ri,a,rc,d2fdr2,X)
    f = SumfuncM2(index,type,ri,a,rc,d2fdr2,X);
end

% Total first derivative of electron density with respect
ri,rj,rk, and rl
function f = dRhodrij(index,type,ri,a,rc,dfdr,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,dfdr,X,i,j,k,l,1);
end

% second derivative of electron density of rij
function f = d2Rhodr2ijkl(index,type,ri,a,rc,dfdr,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,dfdr,X,i,j,k,l,2);
end

% First derivative of Atomic Electron Density with respect to ri
function f = dfdri(index,type,ri,a,rc,dfdr,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,dfdr,X,i,j,k,l,1);
end

%  First derivative of Atomic Electron Density with respect to
ri,rj,rk and rl
function f = dfdrijkl(index,type,ri,a,rc,dfdr,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,dfdr,X,i,j,k,l,3);
end

%  Second derivative of Atomic Electron Density with respect to
ri,rj,rk and rl
function f = d2fdr2ijkl(index,type,ri,a,rc,d2fdr2,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,d2fdr2,X,i,j,k,l,2);
end
```

```
% First derivative of pair potential with respect to ri
function f = dphidri(index,type,ri,a,rc,dphidr,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,dphidr,X,i,j,k,l,1);
end

% First derivative of pair potential with respect to ri,rj,rk
and rl
function f = dPhidrijkl(index,type,ri,a,rc,dphidr,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,dphidr,X,i,j,k,l,3);
end

% Second derivative of pair potential with respect to ri,rj,rk
and rl
function f = d2Phidr2ijkl(index,type,ri,a,rc,d2phidr2,X,i,j,k,l)
    f = Sumfuncrijkl(index,type,ri,a,rc,d2phidr2,X,i,j,k,l,2);
end

% Function Used in the calculations of Elastic Constants at
equlibrum
function f = Vij(index,type,ri,a,rc,dfdr,X,i,j,k,l)
    f = delta(i,j) *
delta(k,l)*dfdri(index,type,ri,a,rc,dfdr,X,i,j,k,l);
end

% Function Used in the calculations of Elastic Constants at
equilibrium
function f = Wijkl(index,type,ri,a,rc,dfdr,d2fdr2,X,i,j,k,l)
    f = d2fdr2ijkl(index,type,ri,a,rc,d2fdr2,X,i,j,k,l) - ...
        dfdrijkl(index,type,ri,a,rc,dfdr,X,i,j,k,l)+ ...
        delta(i,l) * dfdri(index,type,ri,a,rc,dfdr,X,0,0,k,l);
end

% Function Used in the calculations of Elastic Constants at
equilibrium
function f = Bijkl(index,type,ri,a,rc,dphidr,d2phidr2,X,i,j,k,l)
    f = d2Phidr2ijkl(index,type,ri,a,rc,d2phidr2,X,i,j,k,l) -
    ...
    dPhidrijkl(index,type,ri,a,rc,dphidr,X,i,j,k,l)+ ...
    delta(i,l) * dphidri(index,type,ri,a,rc,dphidr,X,0,0,k,l);
    f = 0.5 * f;
end

%*************************************************************
%   Park Hijazi Pd EAM Potential and Derivatives
%*************************************************************
% Pd Embedding Energy Function as a function of electron density
function f = F_Pd(rho,X)
```

```
    f    = 0;
    Fe   = X(1);
    rhoe = X(2);
    n    = X(3);
    if rho > 0
        f = -Fe*(1-n*log(rho/rhoe))*(rho/rhoe)^n;
    end
end

% First derivative of Pd Embedding Energy Function with respect
to electron density
function f = dFdrho_Pd(rho,X)
    f    = 0;
    Fe   = X(1);
    rhoe = X(2);
    n    = X(3);
    if rho > 0
        F = F_Pd(rho,X);
        f = (Fe*n/rho)*(rho/rhoe)^n+(n/rhoe)*F*(rho/rhoe)^(-1);
    end
end

% Second derivative of Pd Embedding Energy Function with respect
to electron density
function f = d2Fdrho2_Pd(rho,X)
    f    = 0;
    Fe   = X(1);
    rhoe = X(2);
    n    = X(3);
    if rho > 0
        F  = F_Pd(rho,X);
        dF = dFdrho_Pd(rho,X);
        f  = -
        (n/rho^2)*Fe*(rho/rhoe)^n+(n^2/rho)*(1/rhoe)*Fe*(rho/rhoe
        )^(n-1)+...
         (n/rhoe)*dF*(rho/rhoe)^(-1)-(n/rhoe^2)*F*(rho/rhoe)^(-
        2);
    end
end

% First derivative of Pd Embedding Energy Function with respect
to rij
function f = dFdr(dFdrho,rho,dRhodr,X)
    f = dFdrho(rho,X) * dRhodr;
end
```

```
% Second derivative of Pd Embedding Energy Function with respect
to rij
function f = d2Fdr2(dFdrho,d2Fdrho2,rho,dRhodr,d2Rhodr2,X)
    f = d2Fdrho2(rho,X) * dRhodr^2  + dFdrho(rho,X) * d2Rhodr2;
end

% First derivative of Pd Embedding Energy Function with respect
to a
function f = dFda(dFdrho,rho,dRhoda,X)
    f = dFdrho(rho,X) * dRhoda;
end

% Second derivative of Pd Embedding Energy Function with respect
to a
function f = d2Fda2(dFdrho,d2Fdrho2,rho,dRhoda,d2Rhoda2,X)
    f = d2Fdrho2(rho,X) * dRhoda^2  + dFdrho(rho,X) * d2Rhoda2;
end

% Pd Atomic Electron Density as a function of interatomic
distance rij
function f = f_PdPd(rij,X)
    fe = X(1);
    Xi = X(2);
    re = X(3);
    f = fe*exp(-Xi*(rij-re));
end

% First derivative Pd Atomic Electron Density function with
respect to rij
function f = dfdr_PdPd(rij,X)
    fe = X(1);
    Xi = X(2);
    re = X(3);
    f = -fe*Xi*exp(-Xi*(rij-re));
end

% First derivative Pd Atomic Electron Density function with
respect to rij
function f = d2fdr2_PdPd(rij,X)
    fe = X(1);
    Xi = X(2);
    re = X(3);
    f  = fe*Xi^2*exp(-Xi*(rij-re));
end

% Pd-Pd pair potential as a function of interatomic distance rij
function f = phi_PdPd(rij,X)
```

```
    phie = X(1);
    S    = X(2);
    B    = X(3);
    re   = X(4);
    f    = -phie*(1+S*(rij/re-1))*exp(-B*(rij/re-1));
end

% First derivative of Pd-Pd pair potential with respect to rij
function f = dphidr_PdPd(r,X)
    phie = X(1);
    S    = X(2);
    B    = X(3);
    re   = X(4);
    f    = (-1/re)*(phie*S*exp(-B*(r/re-1))+B*phi_PdPd(r,X));
end

% Second derivative of Pd-Pd pair potential with respect to rij
function f = d2phidr2_PdPd(r,X)
    phie = X(1);
    S    = X(2);
    B    = X(3);
    re   = X(4);
    f    = (B/re)*((phie*S/re)*exp(-B*(r/re-1))-
    dphidr_PdPd(r,X));
end

%*************************************************************
% Hydrogen EAM Potential and Derivatives
%*************************************************************
% H-H unnormalized pair potential as a function of interatomic
distance rij
function f = phi_HH_u(rij,X)
    DHH     = X(1);
    alphaHH = X(2);
    betaHH  = X(3);
    r0HH    = X(4);
    f = DHH*(betaHH*exp(-alphaHH*(rij-r0HH)) - ...
        alphaHH*exp(-betaHH*(rij-r0HH)));
end

% First derivative of H-H unnormalized pair potential with
respect to rij
function f = dphidr_HH_u(r,X)
    DHH     = X(1);
    alphaHH = X(2);
    betaHH  = X(3);
    r0HH    = X(4);
```

```
    f = DHH*( -alphaHH * betaHH*exp(-alphaHH*(r-r0HH)) +...
        betaHH*alphaHH*exp(-betaHH*(r-r0HH)) );
end

% Secon derivative of H-H unnormalized pair potential with
respect to rij
function f = d2phidr2_HH_u(r,X)
    DHH     = X(1);
    alphaHH = X(2);
    betaHH  = X(3);
    r0HH    = X(4);
    f = DHH*( (alphaHH^2) * betaHH*exp(-alphaHH*(r-r0HH)) - ...
        (betaHH^2)*alphaHH*exp(-betaHH*(r-r0HH)) );
end

% H-H normalized pair potential as a function of interatomic
distance rij
function f = phi_HH(rij,X)
    X2_HH = X(1:1:4);
    X0_HH = X(5:1:6);
    X1_HH = X(7:1:11);
    rho0H = X(11);
    f = phi_HH_u(rij,X2_HH) + 2 * dFdrho_H_u(rho0H,X1_HH) *
    f_HH(rij,X0_HH);
end

% First derivative of H-H normalized pair potential with respect
to rij
function f = dphidr_HH(rij,X)
    X2_HH = X(1:1:4);
    X0_HH = X(5:1:6);
    X1_HH = X(7:1:11);
    rho0H = X(11);
    f = dphidr_HH_u(rij,X2_HH) + 2 * dFdrho_H_u(rho0H,X1_HH) *
    ...
    dfdr_HH(rij,X0_HH);
end

% Second derivative of H-H normalized pair potential with
respect to rij
function f = d2phidr2_HH(rij,X)
    X2_HH = X(1:1:4);
    X0_HH = X(5:1:6);
    X1_HH = X(7:1:11);
    rho0H = X(11);
    f = d2phidr2_HH_u(rij,X2_HH) + 2 * dFdrho_H_u(rho0H,X1_HH) *
     ...
```

```
      d2fdr2_HH(rij,X0_HH);
end

% H Atomic Electron Density as a function of interatomic
distance
function f = f_HH(r,X)
    CH = X(1);
    DH = X(2);
    f = CH*exp(-DH*r);
end

% First derivative H Atomic Electron Density with respect to rij
function f = dfdr_HH(r,X)
    CH = X(1);
    DH = X(2);
    f = -DH*CH*exp(-DH*r);
end

% Second derivative H Atomic Electron Density with respect to
rij
function f = d2fdr2_HH(r,X)
    CH = X(1);
    DH = X(2);
    f = (DH^2)*CH*exp(-DH*r);
end

% H unormalized Embedding Energy function
function f = F_H_u(rho,X)
    aH = X(1);
    bH = X(2);
    cH = X(3);
    dH = X(4);
    EH = 0.0540638;
    %     EH = 0.055;
    f = -cH * ( (1/(2+dH))*(rho+EH)^(2+dH) - ((aH +
    bH)/(1+dH))*...
    (rho+EH)^(1+dH) + ((aH * bH)/dH) *(rho+EH)^dH );
end

% First derivative of H unnormalized Embedding function with
respect to Rho
function f = dFdrho_H_u(rho,X)
    aH = X(1);
    bH = X(2);
    cH = X(3);
    dH = X(4);
    EH = 0.0540638;
```

```
    %     EH = 0.055;
    f = -cH * ( (rho+EH)^(1+dH) - (aH + bH) * (rho+EH)^(dH) +...
        (aH * bH) *(rho+EH)^(dH-1) );
end

% Second derivative of H unnormalized Embedding function with
respect to Rho
function f = d2Fdrho2_H_u(rho,X)
    aH = X(1);
    bH = X(2);
    cH = X(3);
    dH = X(4);
    EH = 0.0540638;
    %     EH = 0.055;
    f = -cH * ( (1+dH)*(rho+EH)^(dH) - dH*(aH + bH) *
    (rho+EH)^(dH-1) + ...
    (dH-1)*(aH * bH) *(rho+EH)^(dH-2) );
end

% H normalized Embedding Energy function
function f = F_H(rho,X)
    X1_HH = X(7:1:10);
    rho0H = X(11);
    f = F_H_u(rho,X1_HH) - dFdrho_H_u(rho0H,X1_HH) * rho;
end

% First derivative of H normalized Embedding function with
respect to Rho
function f = dFdrho_H(rho,X)
    X1_HH = X(7:1:10);
    rho0H = X(11);
    f = dFdrho_H_u(rho,X1_HH) - dFdrho_H_u(rho0H,X1_HH);
end

% Second derivative of H normalized Embedding function with
respect to Rho
function f = d2Fdrho2_H(rho,X)
    X1_HH = X(7:1:10);
    f = d2Fdrho2_H_u(rho,X1_HH);
end

% Pd-H pair potential as a function of interatomic destance rij
function f = phi_PdH(r,X)
    DPdH      = X(1);
    alphaPdH  = X(2);
    betaPdH   = X(3);
    r0PdH     = X(4);
```

```
    f = DPdH*( betaPdH*exp(-alphaPdH*(r-r0PdH)) - ...
        alphaPdH*exp(-betaPdH*(r-r0PdH)) );
end

% First derivative of Pd-H pair potential with respect to
interatomic
% destance rij
function f = dphidr_PdH(r,X)
    DPdH      = X(1);
    alphaPdH  = X(2);
    betaPdH   = X(3);
    r0PdH     = X(4);
    f = DPdH*( -alphaPdH*betaPdH*exp(-alphaPdH*(r-r0PdH)) +...
        betaPdH*alphaPdH*exp(-betaPdH*(r-r0PdH)) );
end

% Second derivative of Pd-H pair potential with respect to
interatomic
% destance rij
function f = d2phidr2_PdH(r,X)
    DPdH      = X(1);
    alphaPdH  = X(2);
    betaPdH   = X(3);
    r0PdH     = X(4);
    f = DPdH*( (alphaPdH^2)*betaPdH*exp(-alphaPdH*(r-r0PdH)) -
    ...
    (betaPdH^2)*alphaPdH*exp(-betaPdH*(r-r0PdH)) );
end

%*****************************************************************
% Summation function
%*****************************************************************
function f = Sumfunc(index,type,ri,a,rc,func,X)
    f = 0;
    cl1 = (a/2)*ri(index,2);
    cl2 = (a/2)*ri(index,3);
    cl3 = (a/2)*ri(index,4);
    n = length(ri(:,1));
    for i = 1:n
        id = ri(i,1);
        if i ~= index && id == type
            l1 = (a/2)*ri(i,2);
            l2 = (a/2)*ri(i,3);
            l3 = (a/2)*ri(i,4);
            rij = sqrt((l1-cl1)^2 + (l2-cl2)^2 + (l3-cl3)^2);
            if rij <= rc
                f = f + func(rij,X);
```

```
            end
        end
    end
end

function f = SumfuncMP(index,type,ri,a,rc,func,X,p)
    f = 0;
    cl1 = (a/2)*ri(index,2);
    cl2 = (a/2)*ri(index,3);
    cl3 = (a/2)*ri(index,4);
    n = length(ri(:,1));
    for i = 1:n
        id = ri(i,1);
        if i ~= index && id == type
            l1 = (a/2)*ri(i,2);
            l2 = (a/2)*ri(i,3);
            l3 = (a/2)*ri(i,4);
            rij = sqrt((l1-cl1)^2 + (l2-cl2)^2 + (l3-cl3)^2);
            if rij <= rc
                f = f + func(rij,X) * (rij/a)^p;
            end
        end
    end
end

function f = SumfuncM(index,type,ri,a,rc,func,X)
    f = 0;
    cl1 = (a/2)*ri(index,2);
    cl2 = (a/2)*ri(index,3);
    cl3 = (a/2)*ri(index,4);
    n = length(ri(:,1));
    for i = 1:n
        id = ri(i,1);
        if i ~= index && id == type
            l1 = (a/2)*ri(i,2);
            l2 = (a/2)*ri(i,3);
            l3 = (a/2)*ri(i,4);
            rij = sqrt((l1-cl1)^2 + (l2-cl2)^2 + (l3-cl3)^2);
            if rij <= rc
                f = f + func(rij,X) * rij/a;
            end
        end
    end
end

function f = SumfuncM2(index,type,ri,a,rc,func,X)
    f = 0;
```

```
    cl1 = (a/2)*ri(index,2);
    cl2 = (a/2)*ri(index,3);
    cl3 = (a/2)*ri(index,4);
    n = length(ri(:,1));
    for i = 1:n
        id = ri(i,1);
        if i ~= index && id == type
            l1 = (a/2)*ri(i,2);
            l2 = (a/2)*ri(i,3);
            l3 = (a/2)*ri(i,4);
            rij = sqrt((l1-cl1)^2 + (l2-cl2)^2 + (l3-cl3)^2);
            if rij <= rc
                f = f + func(rij,X) * (rij/a)^2;
            end
        end
    end
end

function f = Sumfuncrijkl(index,type,r,a,rc,func,X,i,j,k,l,p)
    f = 0;
    ri = 1;
    rj = 1;
    rk = 1;
    rl = 1;
    cl(1) = (a/2)*r(index,2);
    cl(2) = (a/2)*r(index,3);
    cl(3) = (a/2)*r(index,4);
    n = length(r(:,1));
    for ii = 1:n
        id = r(ii,1);
        if ii ~= index && id == type
            l1(1) = (a/2)*r(ii,2);
            l1(2) = (a/2)*r(ii,3);
            l1(3) = (a/2)*r(ii,4);
            rij = sqrt((l1(1)-cl(1))^2 + (l1(2)-cl(2))^2 +
            (l1(3)-cl(3))^2);
            if rij < rc
                if i ~= 0
                    ri = l1(i)-cl(i);
                end
                if j ~= 0
                    rj = l1(j)-cl(j);
                end
                if k ~= 0
                    rk = l1(k)-cl(k);
                end
                if l ~= 0
```

```
                    rl = l1(l)-cl(l);
                end
                f = f + func(rij,X) * ri*rj*rk*rl/rij^p;
            end
        end
    end
end

function f = delta(i,j)
    if(i == j)
        f = 1;
    else
        f = 0;
    end
end

%*****************************************************************
% Experimental data for FCC metals
%*****************************************************************
function [afcce,Ecfcce,c11e,c12e,c44e,Bme,Eve,abcce,Ecbcce,cpre]
= parameters(el)
    index = 1;
    elements = ['Cu';'Al';'Au';'Cu';'Ni';'Pd';'Pt';'Hi'];
    for i = 1:length(elements)
        x = elements(i,:);
        if el == x
            index = i;
        end
    end
    afcc =  [4.09;4.05;4.08;3.615;3.52;3.89;3.92;3.38];
    Ecfcc = [2.85;3.36;3.93;3.54;4.45;3.91;5.77;2.119];
    % Elastic Constants in dyne/cm^2 10^12
    c11 =   [1.24;1.14;1.86;1.70;2.465;2.3412;3.47;0.0];
    c12 =   [0.934;0.619;1.57;1.225;1.473;1.7614;2.51;0.0];
    c44 =
    [0.461;0.316;0.42;0.758;1.247;0.7117;0.765;0.05*1.6021766208]
    ;
    cpr =   [1;1;1;1;1;1;1;0.05*1.6021766208];
    % Unrelaxed
    Evfcc = [1.1;0.866;0.9;1.3;1.70;1.54;1.60;0.0];
    % Relaxed
    B =
    [1.04;0.793;1.67;1.38;1.804;1.9547;2.83;0.5*1.6021766208];
    abcc =  [3.32;3.3;3.18;2.87;2.89;3.16;3.21;0.0];
    Ecbcc = [1;1;1;3.49;1;1;1;1];
    afcce = afcc(index);
    Ecfcce = Ecfcc(index);
```

```
    Eve = Evfcc(index);
    c11e = c11(index);
    c12e = c12(index);
    c44e = c44(index);
    Bme = B(index);
    abcce = abcc(index);
    Ecbcce = Ecbcc(index);
    cpre   = cpr(index);
end

%*****************************************************************
% Experimental and bb-initio data for PdH
%*****************************************************************
function [afcce,Ecfcce,C44fcce,Cprfcce,Bmfcce] =
PdH_parameters(el)
    index1 = 1;
    elements = ['Pd000H';'PdH000';'PdH025';'PdH050';'PdH075';...
    'PdH100';'PdHT50';'PdHT75'];
    [n,m]  = size(elements);
    for i = 1:n
        x = elements(i,:);
        if el == x
            index1 = i;
        end
    end
    a  =  [3.38;3.89;3.9421;4.0007;4.0593;4.1179;4.0007;4.0593];
    Ec =
    [2.119;3.91;3.64024;3.48770;3.37031;3.26966;3.50000;3.50000];
    % Elastic Constants with conv from eV/A^3 to dyne/cm^2 10^12
    C44 = [0.05,0.7117/1.6021766208,0.459,0.42,0.382,0.3440,...
        0.3810,0.3810] * 1.6021766208;  % Pd from Rayne
    Cpr = [0.05,.2899/1.6021766208,0.055,0.135,0.164,0.146,...
        0.1630,0.1630] * 1.6021766208;  % Pd from Rayne
    Bm =  [0.50,1.9547/1.6021766208,1.05,1.05,1.05,1.05,...
        1.05,1.05] * 1.6021766208;  % Pd from Rayne
    afcce  = a(index1);
    Ecfcce = Ec(index1);
    C44fcce  = C44(index1);
    Cprfcce  = Cpr(index1);
    Bmfcce  = Bm(index1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Pd Cutoff Transformed Functions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function f = phi_PdPd_cut(rij,X)
    global rc_1;
```

```
    global NP;
    f = phi_PdPd(rij,X) - phi_PdPd(rc_1,X) + (rc_1/NP) * (1-
    (rij/rc_1)^NP )* dphidr_PdPd(rc_1,X);
end

function f = dphidr_PdPd_cut(rij,X)
    global rc_1;
    global NP;
    f = dphidr_PdPd(rij,X) - (rij/rc_1)^(NP-1) *
    dphidr_PdPd(rc_1,X);
end

function f = d2phidr2_PdPd_cut(rij,X)
    global rc_1;
    global NP;
    f = d2phidr2_PdPd(rij,X)- ((NP-1)/rc_1) * (rij/rc_1)^(NP-2)
    * dphidr_PdPd(rc_1,X);
end

function f = f_PdPd_cut(rij,X)
    global rc_1;
    global NP;
    f = f_PdPd(rij,X) - f_PdPd(rc_1,X) + (rc_1/NP) * (1-
    (rij/rc_1)^NP) * dfdr_PdPd(rc_1,X);
end

function f = dfdr_PdPd_cut(rij,X)
    global rc_1;
    global NP;
    f = dfdr_PdPd(rij,X) - (rij/rc_1)^(NP-1) *
    dfdr_PdPd(rc_1,X);
end

function f = d2fdr2_PdPd_cut(rij,X)
    global rc_1;
    global NP;
    f = d2fdr2_PdPd(rij,X)- ((NP-1)/rc_1) * (rij/rc_1)^(NP-2) *
    dfdr_PdPd(rc_1,X);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Hydrogen Cutoff Transformed functions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function f = phi_HH_cut(rij,X)
    global rc_2;
    global NP2;
```

```
    f = phi_HH(rij,X) - phi_HH(rc_2,X) + (rc_2/NP2) * (1-
(rij/rc_2)^NP2 )* dphidr_HH(rc_2,X);
end

function f = dphidr_HH_cut(rij,X)
    global rc_2;
    global NP2;
    f = dphidr_HH(rij,X) - (rij/rc_2)^(NP2-1) *
dphidr_HH(rc_2,X);
end

function f = d2phidr2_HH_cut(rij,X)
    global rc_2;
    global NP2;
    f = d2phidr2_HH(rij,X)- ((NP2-1)/rc_2) * (rij/rc_2)^(NP2-2)
* dphidr_HH(rc_2,X);
end

function f = f_HH_cut(rij,X)
    global rc_2;
    global NP2;
    f = f_HH(rij,X) - f_HH(rc_2,X) + (rc_2/NP2) * (1-
(rij/rc_2)^NP2) * dfdr_HH(rc_2,X);
end

function f = dfdr_HH_cut(rij,X)
    global rc_2;
    global NP2;
    f = dfdr_HH(rij,X) - (rij/rc_2)^(NP2-1) * dfdr_HH(rc_2,X);
end

function f = d2fdr2_HH_cut(rij,X)
    global rc_2;
    global NP2;
    f = d2fdr2_HH(rij,X)- ((NP2-1)/rc_2) * (rij/rc_2)^(NP2-2) *
    dfdr_HH(rc_2,X);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PdH Cutoff Transformed Pair Potential
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function f = phi_PdH_cut(rij,X)
    global rc_3;
    global NP3;
    f = phi_PdH(rij,X) - phi_PdH(rc_3,X) + (rc_3/NP3) * (1-
    (rij/rc_3)^NP3 )* dphidr_PdH(rc_3,X);
end
```

```
function f = dphidr_PdH_cut(rij,X)
    global rc_3;
    global NP3;
    f = dphidr_PdH(rij,X) - (rij/rc_3)^(NP3-1) *
    dphidr_PdH(rc_3,X);
end

function f = d2phidr2_PdH_cut(rij,X)
    global rc_3;
    global NP3;
    f = d2phidr2_PdH(rij,X)- ((NP3-1)/rc_3) * (rij/rc_3)^(NP3-2)
    * dphidr_PdH(rc_3,X);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PdCu Pair Potential
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function f = phi_PdCu(rij,X)
    X0_a = X(1:3);
    X0_b = X(4:6);
    X2_a = X(7:10);
    X2_b = X(11:14);
    f_a = f_PdPd(rij,X0_a);
    f_b = f_PdPd(rij,X0_b);
    phi_a = phi_PdPd(rij,X2_a);
    phi_b = phi_PdPd(rij,X2_b);
    f = .5*( f_b*f_a^-1*phi_a + f_a*f_b^-1*phi_b );
end

function f = dphidr_PdCu(rij,X)
    X0_a = X(1:3);
    X0_b = X(4:6);
    X2_a = X(7:10);
    X2_b = X(11:14);
    f_a = f_PdPd(rij,X0_a);
    f_b = f_PdPd(rij,X0_b);
    phi_a = phi_PdPd(rij,X2_a);
    phi_b = phi_PdPd(rij,X2_b);
    dfdr_a = dfdr_PdPd(rij,X0_a);
    dfdr_b = dfdr_PdPd(rij,X0_b);
    dphidr_a = dphidr_PdPd(rij,X2_a);
    dphidr_b = dphidr_PdPd(rij,X2_b);
    f = .5*( ((-1*f_b*dfdr_a*f_a^-2 + dfdr_b*f_a^-1)*phi_a +
(f_b*f_a^-1)*dphidr_a)...
        + ((-1*f_a*dfdr_b*f_b^-2 + dfdr_a*f_b^-1)*phi_b +
    (f_a*f_b^-1)*dphidr_b) );
```

```
end

function f = d2phidr2_PdCu(rij,X)
    X0_a = X(1:3);
    X0_b = X(4:6);
    X2_a = X(7:10);
    X2_b = X(11:14);
    f_a = f_PdPd(rij,X0_a);
    f_b = f_PdPd(rij,X0_b);
    phi_a = phi_PdPd(rij,X2_a);
    phi_b = phi_PdPd(rij,X2_b);
    dfdr_a = dfdr_PdPd(rij,X0_a);
    dfdr_b = dfdr_PdPd(rij,X0_b);
    dphidr_a = dphidr_PdPd(rij,X2_a);
    dphidr_b = dphidr_PdPd(rij,X2_b);
    d2fdr2_a = d2fdr2_PdPd(rij,X0_a);
    d2fdr2_b = d2fdr2_PdPd(rij,X0_b);
    d2phidr2_a = d2phidr2_PdPd(rij,X2_a);
    d2phidr2_b = d2phidr2_PdPd(rij,X2_b);
    Phi_ab =
    phi_ab(f_a,dfdr_a,d2fdr2_a,f_b,dfdr_b,d2fdr2_b,phi_a,dphidr_a
    ,d2phidr2_a,phi_b,dphidr_b,d2phidr2_b);
    Phi_ba =
    phi_ab(f_b,dfdr_b,d2fdr2_b,f_a,dfdr_a,d2fdr2_a,phi_b,dphidr_b
    ,d2phidr2_b,phi_a,dphidr_a,d2phidr2_a);
    f = .5*(Phi_ab + Phi_ba);
end

function f =
phi_ab(f_a,dfdr_a,d2fdr2_a,f_b,dfdr_b,d2fdr2_b,phi_a,dphidr_a,d2
phidr2_a,phi_b,dphidr_b,d2phidr2_b)
    aa = -1*(dfdr_b*dfdr_a*f_a^-2 + f_b*d2fdr2_a*f_a^-2 -
    2*f_b*(dfdr_a^2)*f_a^-3)...
    + (d2fdr2_b*f_a^-1 - dfdr_b*f_a^-2*dfdr_a);
    a = aa*phi_a + (-1*f_b*dfdr_a*f_a^-2 + dfdr_b*f_a^-
    1)*dphidr_a;
    b = dfdr_b*f_a^-1*dphidr_a - f_b*f_a^-2*dfdr_a*dphidr_a +
    f_b*f_a^-1*d2phidr2_a;
    f = a + b;
end
```

MATLAB PROGRAM FOR GENERATING DYNAMO FORMAT POTENTIAL FILE

Values for each pair potential $\phi(r)$ array are listed for all i, j element pairs. Since these interactions are symmetric ($i, j = j, i$) only ϕ arrays with $i \geq j$ are listed in the following order:

i, j = (1, 1), (2, 1), (2, 2), (3, 1), (3, 2), (3, 3), (4, 1), … ($N_{elements}$, $N_{elements}$).

```
function EAM_Potential_DYNAMO
    clc
    format short
    % Present Working directory
    pwd

    global fId;

    %   Pd and PdH
    x_Pd = [2.054085    0.216817    8.414105    7.221224
    0.999999    3.316887];
    x_PdH = [0.589510   1.104827    0.942490    2.145808
    0.942201...
    0.740938    2.373944    1.702142    8.370790    62.343273...
    0.000100    1.187000    1.300000    3.474173    4.903820];

    fId = fopen('PdAgCuH.eam.alloy','w');

    % re fit scaling only mixing rule potential

    S_AgPd = 1.8319;
    S_PdAg = 1.1063;

    % Pd Expermintal data
    [aFcce,EcFcce,c11e,c12e,c44e,~,Eve,aBcce,EcBcce] =
    parameters('Pd');

    % PdPd
    re = aFcce/sqrt(2);
    Fe = EcFcce-Eve;
    N  = 4;
    V  = aFcce^3/N;
    fe = EcFcce/V;
    x = [x_Pd, x_PdH];

    Xi   = x(1);
    phie = x(2);
    S    = x(3);
    B    = x(4);
    n    = x(5);
    rhoe = x(6);
```

```
fe_S = S_PdAg*EcFcce/V;
rhoe_S = S_PdAg*rhoe;

% Phi_HH
DHH =  x(7);
aHH =  x(8);
bHH =  x(9);
% f_HH
CH = x(10);
DH = x(11);

% Phi_PdH
DPdH = x(12);
aPdH = x(13);
bPdH = x(14);

% F_H
aH = x(15);
bH = x(16);
cH = x(17);
dH = x(18);
%
r0PdH = x(19);
r0HH  = x(20);
rho0H = x(21);

% Pd data
%  Rho PdPd fitting parameters
X0_PdPd = [fe_S,Xi,re];
%  F PdPd fitting parameters
X1_PdPd = [Fe,rhoe_S,n];
%  Phi PdPd fitting parametters
X2_PdPd = [phie,S,B,re];

%   F PdPd fitting parameters
X_F_PdPd   = [Fe,rhoe,n,0,0];
%   f Pd
X_f_PdPd   = [fe,Xi,re];
%   Phi PdPd fitting parametters
X_Phi_PdPd = [phie,S,B,re];

x_Ag = [1.584768032776473   0.154164182657381
8.491335427574112...
7.183185289310671   1.022270092304303   2.213230197455566];

% Ag Experimental data
```

```
[aFcce_Ag,EcFcce_Ag,C11e_Ag,C12e_Ag,C44e_Ag,Bme_Ag,Eve_Ag,aBc
ce_Ag,EcBcce_Ag] = parameters('Ag');

re = aFcce_Ag/sqrt(2);
Fe = EcFcce_Ag-Eve_Ag;
N  = 4;
V  = aFcce_Ag^3/N;
fe = EcFcce_Ag/V;

% Ag Fitting Parameters
Xi   = x_Ag(1);
phie = x_Ag(2);
S    = x_Ag(3);
B    = x_Ag(4);
n    = x_Ag(5);
rhoe = x_Ag(6);

fe_S = S_AgPd*EcFcce_Ag/V;
rhoe_S = S_AgPd*rhoe;

%  Rho PdPd fitting parameters
X0_AgAg = [fe_S,Xi,re];
%  F PdPd fitting parameters
X1_AgAg = [Fe,rhoe_S,n];
%  Phi PdPd fitting parametters
X2_AgAg = [phie,S,B,re];

X_Phi_PdAg = [X0_PdPd X0_AgAg X2_PdPd X2_AgAg];

%   F AgAg fitting parameters
X_F_AgAg   = [Fe,rhoe,n,0,0];
%   f Pd
X_f_AgAg   = [fe,Xi,re];
%   Phi PdPd fitting parametters
X_Phi_AgAg = [phie,S,B,re];

% H data
% F_H
X_F_HH  =  [aH,bH,cH,dH,rho0H];
% rho_HH
X_f_HH   = [CH,DH,0];
% Phi_HH
X_phi_HH = [DHH,aHH,bHH,r0HH];

% PdH data
% Vectores of fitting parameters
```

```
X_Phi_PdH = [DPdH,aPdH,bPdH,r0PdH];

%   PdAgH untransformed fit 1  f = 0.4417
X_Phi_AgH = [1.476745339175117   1.967649395899897
1.741864506174084   1.850016663067700];
%     X_Phi_AgH = [1  1   1   1];

% Phi_HH, rho_HH,   F_H
X_Phi_HH = [X_phi_HH, X_f_HH, X_F_HH];

%Parameters for PdCu
S_PdCu = 1.7097;
S_CuPd = 1.3419;

[aFcce,EcFcce,c11e,c12e,c44e,~,Eve,aBcce,EcBcce] =
parameters('Pd');

% PdPd
re = aFcce/sqrt(2);
Fe = EcFcce-Eve;
N  = 4;
V  = aFcce^3/N;
fe = EcFcce/V;
x = [x_Pd, x_PdH];

Xi   = x(1);
phie = x(2);
S    = x(3);
B    = x(4);
n    = x(5);
rhoe = x(6);

fe_S = S_PdCu*EcFcce/V;
rhoe_S = S_PdCu*rhoe;

% Phi_HH
DHH =  x(7);
aHH =  x(8);
bHH =  x(9);
% f_HH
CH = x(10);
DH = x(11);

% Phi_PdH
DPdH = x(12);
aPdH = x(13);
bPdH = x(14);
```

```
% F_H
aH = x(15);
bH = x(16);
cH = x(17);
dH = x(18);
%
r0PdH = x(19);
r0HH  = x(20);
rho0H = x(21);

% Pd data
%  Rho PdPd fitting parameters
X0_PdPd = [fe_S,Xi,re];
%  F PdPd fitting parameters
X1_PdPd = [Fe,rhoe_S,n];
%  Phi PdPd fitting parametters
X2_PdPd = [phie,S,B,re];

x_Cu = [2.504500   0.175425   8.713725...
    6.906629   0.560027   3.648665];
% Cu Experimental data

[aFcce_Cu,EcFcce_Cu,C11e_Cu,C12e_Cu,C44e_Cu,Bme_Cu,Eve_Cu,aBc
ce_Cu,EcBcce_Cu] = parameters('Cu');

re = aFcce_Cu/sqrt(2);
Fe = EcFcce_Cu-Eve_Cu;
N  = 4;
V  = aFcce_Cu^3/N;
fe = EcFcce_Cu/V;

% Cu Fitting Parameters
Xi   = x_Cu(1);
phie = x_Cu(2);
S    = x_Cu(3);
B    = x_Cu(4);
n    = x_Cu(5);
rhoe = x_Cu(6);

fe_S = S_CuPd*EcFcce_Cu/V;
rhoe_S = S_CuPd*rhoe;

%  Rho CuCu fitting parameters
X0_CuCu = [fe_S,Xi,re];
%  F CuCu fitting parameters
X1_CuCu = [Fe,rhoe_S,n];
```

```
%  Phi CuCu fitting parametters
X2_CuCu = [phie,S,B,re];

X_Phi_PdCu = [X0_PdPd X0_CuCu X2_PdPd X2_CuCu];

%   F CuCu fitting parameters
X_F_CuCu   = [Fe,rhoe,n,0,0];
%   f Cu
X_f_CuCu   = [fe,Xi,re];
%   Phi CuCu fitting parametters
X_Phi_CuCu = [phie,S,B,re];

% PdH data
X_Phi_PdH = [DPdH,aPdH,bPdH,r0PdH];

%   PdCuH untransformed fit 1  f = 0.258
X_Phi_CuH = [0.780004347905691   2.100001766883917
1.799998446869885   1.900008190080419];

% Phi_HH, rho_HH,   F_H
X_Phi_HH = [X_phi_HH, X_f_HH, X_F_HH];
%Parameters for AgCu

S_AgCu = 1.4626;
S_CuAg = 1.6275;


[aFcce_Ag,EcFcce_Ag,C11e_Ag,C12e_Ag,C44e_Ag,Bme_Ag,Eve_Ag,aBc
ce_Ag,EcBcce_Ag] = parameters('Ag');

re = aFcce_Ag/sqrt(2);
Fe = EcFcce_Ag-Eve_Ag;
N  = 4;
V  = aFcce_Ag^3/N;
fe = EcFcce_Ag/V;

% Ag Fitting Parameters
Xi   = x_Ag(1);
phie = x_Ag(2);
S    = x_Ag(3);
B    = x_Ag(4);
n    = x_Ag(5);
rhoe = x_Ag(6);

fe_S = S_AgCu*EcFcce_Ag/V;
rhoe_S = S_AgCu*rhoe;
```

```
%  Rho PdPd fitting parameters
X0_AgAg = [fe_S,Xi,re];
%  F PdPd fitting parameters
X1_AgAg = [Fe,rhoe_S,n];
%  Phi PdPd fitting parametters
X2_AgAg = [phie,S,B,re];


[aFcce_Cu,EcFcce_Cu,C11e_Cu,C12e_Cu,C44e_Cu,Bme_Cu,Eve_Cu,aBc
ce_Cu,EcBcce_Cu] = parameters('Cu');

re = aFcce_Cu/sqrt(2);
Fe = EcFcce_Cu-Eve_Cu;
N  = 4;
V  = aFcce_Cu^3/N;
fe = EcFcce_Cu/V;

% Cu Fitting Parameters
Xi   = x_Cu(1);
phie = x_Cu(2);
S    = x_Cu(3);
B    = x_Cu(4);
n    = x_Cu(5);
rhoe = x_Cu(6);

fe_S = S_CuAg*EcFcce_Cu/V;
rhoe_S = S_CuAg*rhoe;

%  Rho CuCu fitting parameters
X0_CuCu = [fe_S,Xi,re];
%  F CuCu fitting parameters
X1_CuCu = [Fe,rhoe_S,n];
%  Phi CuCu fitting parametters
X2_CuCu = [phie,S,B,re];

X_Phi_AgCu = [X0_AgAg X0_CuCu X2_AgAg X2_CuCu];

phi =
{'phi_PdPd';'phi_PdAg';'phi_AgAg';'phi_PdCu';'phi_AgCu';'phi_
CuCu';'phi_PdH';'phi_AgH';'phi_CuH';'phi_HH'};
f   = {'f_PdPd';'f_AgAg';'f_CuCu';'f_HH'};
F   = {'F_Pd';'F_Ag';'F_Cu';'F_H'};

[m n] = size(X_phi_HH);
X_Phi = zeros(length(phi),n);
X_Phi(1,1:length(X_Phi_PdPd)) = X_Phi_PdPd;
X_Phi(2,1:length(X_Phi_PdAg)) = X_Phi_PdAg;
```

```
    X_Phi(3,1:length(X_Phi_AgAg)) = X_Phi_AgAg;
    X_Phi(4,1:length(X_Phi_PdCu)) = X_Phi_PdCu;
    X_Phi(5,1:length(X_Phi_AgCu)) = X_Phi_AgCu;
    X_Phi(6,1:length(X_Phi_CuCu)) = X_Phi_CuCu;
    X_Phi(7,1:length(X_Phi_PdH))  = X_Phi_PdH;
    X_Phi(8,1:length(X_Phi_AgH))  = X_Phi_AgH;
    X_Phi(9,1:length(X_Phi_CuH))  = X_Phi_CuH;
    X_Phi(10,1:length(X_Phi_HH))  = X_Phi_HH;

    X_f   = [X_f_PdPd; X_f_AgAg; X_f_CuCu; X_f_HH];
    X_F   = [X_F_PdPd; X_F_AgAg; X_F_CuCu; X_F_HH];

    % element used to use

    element = ['Pd';'Ag';'Cu';'H '];
    edata   = {46,106.42,3.89,'FCC'; 47,107.87,4.09,'FCC';
    29,63.546,3.615,'FCC'; 1,1.01,3.38,'FCC'};

    rc   = 5.35;
    nr   = 3000;
    nrho = nr;
    dr   = rc/nr;
    drho = 50/nrho;
    r    = [0:dr:nr*dr];
    rho  = [0:drho:drho*nrho];

    alloy = ['Pd';'Ag';'Cu';'H '];


    WritePotential_setf2(alloy,element,edata,nrho,drho,nr,dr,rc,r
    ho,r,f,X_f,F,X_F,phi,X_Phi)
    fclose(fId);
end


function
WritePotential_setf2(alloy,element,edata,nrho,drho,nr,dr,rc,rho,
r,f,X_f,F,X_F,phi,X_phi)
    global fId;
    fprintf(fId,'setfl format\n');
    s = alloy(1,:);
    for i = 2:length(alloy(:,1))
        s = cat(2,s,alloy(i,:));
    end

    fprintf(fId,'%s functions (universal3)\n',s);
    fprintf(fId,'%s functions (universal3)\n',s);
```

```
    fprintf(fId,'%d  %s  %s  %s
    %s\n',length(alloy(:,1)),alloy(1,:),alloy(2,:),alloy(3,:),all
    oy(4,:));
    fprintf(fId,'%d\t%e\t%d\t%e\t%e\n',nrho,drho,nr,dr,rc);

    for i = 1:length(alloy(:,1))
        for index = 1:length(element)
            if alloy(i,:) == element(index,:)

                Fp = str2func(char(F(index,:)));
                X1 = X_F(index,:);
                EF = Embidding_function_2(rho,Fp,X1);

                fp = str2func(char(f(index,:)));
                X2 = X_f(index,:);
                Ef = Electron_density_2(r,fp,X2);

                numedata = cell2mat(edata(index,1:3));
                fprintf(fId,'%d\t%6.3f\t%6.3f\t',numedata(1,:));
                structure =
                string(cell2mat(edata(index,4:end)));
                fprintf(fId,'%s\n',structure);

                Write_F(rho,EF);
                Write_f(r,Ef);
            end
        end
    end
    for index = 1:length(phi)
        phip = str2func(char(phi(index,:)));
        X3 = X_phi(index,:);
        [PHI,RPHI] = Two_body_potential(r,phip,X3);
        Write_rPHI(r,PHI);
    end
end

function FF = Embidding_function_2(rho,F,X)
    for i=1:length(rho)
        if rho(i) ~= 0
            FF(i) = F(rho(i),X);
        else
            FF(i) = 0;
        end
    end
end

function ff = Electron_density_2(r,f,X)
```

```
    for i = 1:length(r)
        ff(i) = f(r(i),X);
    end
end


function [PHI,RPHI] = Two_body_potential(r,phi,X)
    for i = 1:length(r)
        PHI(i)  = phi(r(i),X);
        RPHI(i) = r(i)*PHI(i);
    end
end

% Park & Hijazi phi_PdPd
function f = phi_PdPd(rij,X)
    phie = X(1);
    S    = X(2);
    B    = X(3);
    re   = X(4);
    f    = -phie*(1+S*(rij/re-1))*exp(-B*(rij/re-1));
end

% dphidr_H
function f = dphidr_PdPd(r,X)
    phie = X(1);
    S    = X(2);
    B    = X(3);
    re   = X(4);
    f    = (-1/re)*(phie*S*exp(-B*(r/re-1))+B*phi_PdPd(r,X));
end

% d2phidr2_H
function f = d2phidr2_PdPd(r,X)
    phie = X(1);
    S    = X(2);
    B    = X(3);
    re   = X(4);
    f    = (B/re)*((phie*S/re)*exp(-B*(r/re-1))-
dphidr_PdPd(r,X));
end

% Electron density
function f = f_PdPd(rij,X)
    fe = X(1);
    Xi = X(2);
    re = X(3);
    f = fe*exp(-Xi*(rij-re));
```

```
end

% Park & Hijazi F_P
function f = F_Pd(rho,X)
    f = 0;
    Fe   = X(1);
    rhoe = X(2);
    n    = X(3);
    if rho > 0.0
        f    = -Fe*(1-n*log(rho/rhoe))*(rho/rhoe)^n;
    end
end

% Zhou PhiHH Equation (5.7) page 62
% QHH(r) = DHH(HH_exp_?_HH_r ? r0,HH__ ? _HH_exp_?_HH_r ? r0,HH
,
function f = phi_HH_u(rij,X)
    DHH     = X(1);
    alphaHH = X(2);
    betaHH  = X(3);
    r0HH    = X(4);
    f = DHH*(betaHH*exp(-alphaHH*(rij-r0HH)) - alphaHH*exp(-
    betaHH*(rij-r0HH)));
end

function f = dphidr_HH_u(r,X)
    DHH     = X(1);
    alphaHH = X(2);
    betaHH  = X(3);
    r0HH    = X(4);
    f = DHH*( -alphaHH * betaHH*exp(-alphaHH*(r-r0HH)) +
    betaHH*alphaHH*exp(-betaHH*(r-r0HH)) );
end

% d2phidr2_H
function f = d2phidr2_HH_u(r,X)
    DHH     = X(1);
    alphaHH = X(2);
    betaHH  = X(3);
    r0HH    = X(4);
    f = DHH*( (alphaHH^2) * betaHH*exp(-alphaHH*(r-r0HH)) -
    (betaHH^2)*alphaHH*exp(-betaHH*(r-r0HH)) );
end

% H-H normalized pair potential as a function of interatomic
distance rij
function f = phi_HH(rij,X)
```

```
    X2_HH = X(1:1:4);
    X0_HH = X(5:1:7);
    X1_HH = X(8:1:12);
    rho0H = X(12);
    f = phi_HH_u(rij,X2_HH) + 2 * dFdrho_H_u(rho0H,X1_HH) *
    f_HH(rij,X0_HH);
end

% First derivative of H-H normalized pair potential with respect
to rij
function f = dphidr_HH(rij,X)
    X2_HH = X(1:1:4);
    X0_HH = X(5:1:7);
    X1_HH = X(8:1:12);
    rho0H = X(12);
    f = dphidr_HH_u(rij,X2_HH) + 2 * dFdrho_H_u(rho0H,X1_HH) *
    ...
    dfdr_HH(rij,X0_HH);
end

% Second derivative of H-H normalized pair potential with
respect to rij
function f = d2phidr2_HH(rij,X)
    X2_HH = X(1:1:4);
    X0_HH = X(5:1:7);
    X1_HH = X(8:1:12);
    rho0H = X(12);
    f = d2phidr2_HH_u(rij,X2_HH) + 2 * dFdrho_H_u(rho0H,X1_HH) *
    ...
    d2fdr2_HH(rij,X0_HH);
end

% H Atomic Electron Density as a function of interatomic
distance
function f = f_HH(r,X)
    CH = X(1);
    DH = X(2);
    f = CH*exp(-DH*r);
end

% First derivative H Atomic Electron Density with respect to rij
function f = dfdr_HH(r,X)
    CH = X(1);
    DH = X(2);
    f = -DH*CH*exp(-DH*r);
end
```

```
% Second derivative H Atomic Electron Density with respect to
rij
function f = d2fdr2_HH(r,X)
    CH = X(1);
    DH = X(2);
    f = (DH^2)*CH*exp(-DH*r);
end

% H unormalized Embedding Energy function
function f = F_H_u(rho,X)
    aH = X(1);
    bH = X(2);
    cH = X(3);
    dH = X(4);
    EH = 0.0540638;
    f = -cH * ( (1/(2+dH))*(rho+EH)^(2+dH) - ((aH +
    bH)/(1+dH))*...
     (rho+EH)^(1+dH) + ((aH * bH)/dH) *(rho+EH)^dH );
end

% First derivative of H unnormalized Embedding function with
respect to Rho
function f = dFdrho_H_u(rho,X)
    aH = X(1);
    bH = X(2);
    cH = X(3);
    dH = X(4);
    EH = 0.0540638;
    f = -cH * ( (rho+EH)^(1+dH) - (aH + bH) * (rho+EH)^(dH) +...
        (aH * bH) *(rho+EH)^(dH-1) );
end

% Second derivative of H unnormalized Embedding function with
respect to Rho
function f = d2Fdrho2_H_u(rho,X)
    aH = X(1);
    bH = X(2);
    cH = X(3);
    dH = X(4);
    EH = 0.0540638;
    f = -cH * ( (1+dH)*(rho+EH)^(dH) - dH*(aH + bH) *
    (rho+EH)^(dH-1) + ...
     (dH-1)*(aH * bH) *(rho+EH)^(dH-2) );
end

% H normalized Embedding Energy function
function f = F_H(rho,X)
```

```
    X1_HH = X(1:1:4);
    rho0H = X(5);
    f = F_H_u(rho,X1_HH) - dFdrho_H_u(rho0H,X1_HH) * rho;
end

% First derivative of H normalized Embedding function with
respect to Rho
function f = dFdrho_H(rho,X)
    X1_HH = X(1:1:4);
    rho0H = X(5);
    f = dFdrho_H_u(rho,X1_HH) - dFdrho_H_u(rho0H,X1_HH);
end

% Second derivative of H normalized Embedding function with
respect to Rho
function f = d2Fdrho2_H(rho,X)
    X1_HH = X(1:1:4);
    f = d2Fdrho2_H_u(rho,X1_HH);
end

% Pd-H pair potential as a function of interatomic destance rij
function f = phi_PdH(r,X)
    DPdH      = X(1);
    alphaPdH  = X(2);
    betaPdH   = X(3);
    r0PdH     = X(4);
    f = DPdH*( betaPdH*exp(-alphaPdH*(r-r0PdH)) - ...
        alphaPdH*exp(-betaPdH*(r-r0PdH)) );
end

% First derivative of Pd-H pair potential with respect to
interatomic
% destance rij
function f = dphidr_PdH(r,X)
    DPdH      = X(1);
    alphaPdH  = X(2);
    betaPdH   = X(3);
    r0PdH     = X(4);
    f = DPdH*( -alphaPdH*betaPdH*exp(-alphaPdH*(r-r0PdH)) +...
        betaPdH*alphaPdH*exp(-betaPdH*(r-r0PdH)) );
end

% Second derivative of Pd-H pair potential with respect to
interatomic
% destance rij
function f = d2phidr2_PdH(r,X)
    DPdH      = X(1);
```

```
    alphaPdH  = X(2);
    betaPdH   = X(3);
    r0PdH     = X(4);
    f = DPdH*( (alphaPdH^2)*betaPdH*exp(-alphaPdH*(r-r0PdH)) -
    ...
    (betaPdH^2)*alphaPdH*exp(-betaPdH*(r-r0PdH)) );
end

function [afcce,Ecfcce,c11e,c12e,c44e,Bme,Eve,abcce,Ecbcce,cpre]
= parameters(el)
    index = 1;
    elements = ['Ag';'Al';'Au';'Cu';'Ni';'Pd';'Pt';'Hi'];
    for i = 1:length(elements)
        x = elements(i,:);
        if el == x
            index = i;
        end
    end
    afcc =  [4.09;4.05;4.08;3.615;3.52;3.89;3.92;3.38];
    Ecfcc = [2.85;3.36;3.93;3.54;4.45;3.91;5.77;2.119];
    c11 =   [1.24;1.14;1.86;1.70;2.465;2.341;3.47;0.0];
    c12 =   [0.934;0.619;1.57;1.225;1.473;1.761;2.51;0.0];
    c44 =   [0.461;0.316;0.42;0.758;1.247;0.712;0.765;0.05];
    cpr =   [1,1,1,1,1,1,1,0.05];
    % Unrelaxed
    Evfcc = [1.1;0.866;0.9;1.3;1.70;1.54;1.60;0.0];
    % Relaxed
    % Evfcc = [1.3;1.1;0.75;0.9;1.60;1.40;1.50;0.0];
    B =     [1.04;0.793;1.67;1.38;1.804;1.95;2.83;0.50];
    abcc =  [3.32;3.3;3.18;2.87;2.89;3.16;3.21;0.0];
    Ecbcc = [1;1;1;3.49;1;1;1;1];
    afcce = afcc(index);
    Ecfcce = Ecfcc(index);
    Eve = Evfcc(index);
    c11e = c11(index);
    c12e = c12(index);
    c44e = c44(index);
    Bme = B(index);
    abcce = abcc(index);
    Ecbcce = Ecbcc(index);
    cpre   = cpr(index);
end

function [afcce,Ecfcce] = PdH_parameters(el)
    index1 = 1;
```

```
    elements =
    ['Pd000H';'PdH000';'PdH025';'PdH050';'PdH075';'PdH100';'PdHT5
    0';'PdHT75'];
    [n,m]  = size(elements);
    for i = 1:n
        x = elements(i,:);
        if el == x
            index1 = i;
        end
    end

    % From Sandia Paper
    a  =  [3.38;3.89;3.9421;4.0007;4.0593;4.1179;4.0007;4.0593];
    Ec =
    [2.119;3.91;3.64024;3.48770;3.37031;3.26966;3.50000;3.50000];

    afcce  = a(index1);
    Ecfcce = Ec(index1);
end

function Write_F(rho,F)
    global fId;
    for i = 1:5:length(rho)-1
        for j = 0:4
            fprintf(fId,'%20.16e\t',F(i+j));
        end
        fprintf(fId,'\n');
    end
end


function Write_f(r,f)
    global fId;
    for i = 1:5
        f(length(r)-i)=0;
    end
    for i = 1:5:length(r)-1
        for j = 0:4
            fprintf(fId,'%20.16e\t',f(i+j));
        end
        fprintf(fId,'\n');
    end
end


function Write_rPHI(r,PHI)
    global fId;
```

```
    for i = 1:5
        PHI(length(r)-i)=0;
    end
    for i = 1:5:length(r)-1
        for j = 0:4
            fprintf(fId,'%20.16e\t',r(i+j)*PHI(i+j));
        end
        fprintf(fId,'\n');
    end
end

function f = phi_PdAg(rij,X)
    X0_a = X(1:3);
    X0_b = X(4:6);
    X2_a = X(7:10);
    X2_b = X(11:14);
    f_a = f_PdPd(rij,X0_a);
    f_b = f_PdPd(rij,X0_b);
    phi_a = phi_PdPd(rij,X2_a);
    phi_b = phi_PdPd(rij,X2_b);
    f = .5*( f_b*f_a^-1*phi_a + f_a*f_b^-1*phi_b );
end


function f = phi_AgCu(rij,X)
    X0_a = X(1:3);
    X0_b = X(4:6);
    X2_a = X(7:10);
    X2_b = X(11:14);
    f_a = f_AgAg(rij,X0_a);
    f_b = f_AgAg(rij,X0_b);
    phi_a = phi_AgAg(rij,X2_a);
    phi_b = phi_AgAg(rij,X2_b);
    f = .5*( f_b*f_a^-1*phi_a + f_a*f_b^-1*phi_b );
end


function f = phi_PdCu(rij,X)
    X0_a = X(1:3);
    X0_b = X(4:6);
    X2_a = X(7:10);
    X2_b = X(11:14);
    f_a = f_PdPd(rij,X0_a);
    f_b = f_PdPd(rij,X0_b);
    phi_a = phi_PdPd(rij,X2_a);
    phi_b = phi_PdPd(rij,X2_b);
    f = .5*( f_b*f_a^-1*phi_a + f_a*f_b^-1*phi_b );
```

```
end

function f = phi_AgH(r,X)
    DPdH      = X(1);
    alphaPdH  = X(2);
    betaPdH   = X(3);
    r0PdH     = X(4);
    f = DPdH*( betaPdH*exp(-alphaPdH*(r-r0PdH)) - ...
        alphaPdH*exp(-betaPdH*(r-r0PdH)) );
end


function f = phi_CuH(r,X)
    DPdH      = X(1);
    alphaPdH  = X(2);
    betaPdH   = X(3);
    r0PdH     = X(4);
    f = DPdH*( betaPdH*exp(-alphaPdH*(r-r0PdH)) - ...
        alphaPdH*exp(-betaPdH*(r-r0PdH)) );
end


function f = f_AgAg(rij,X)
    fe = X(1);
    Xi = X(2);
    re = X(3);
    f = fe*exp(-Xi*(rij-re));
end

function f = f_CuCu(rij,X)
    fe = X(1);
    Xi = X(2);
    re = X(3);
    f = fe*exp(-Xi*(rij-re));
end

% Park & Hijazi F_P
function f = F_Ag(rho,X)
    f = 0;
    Fe   = X(1);
    rhoe = X(2);
    n    = X(3);
    if rho > 0.0
        f    = -Fe*(1-n*log(rho/rhoe))*(rho/rhoe)^n;
    end
end
```

```
function f = F_Cu(rho,X)
    f = 0;
    Fe   = X(1);
    rhoe = X(2);
    n    = X(3);
    if rho > 0.0
        f    = -Fe*(1-n*log(rho/rhoe))*(rho/rhoe)^n;
    end
end

% Park & Hijazi phi_AgAg
function f = phi_AgAg(rij,X)
    phie = X(1);
    S    = X(2);
    B    = X(3);
    re   = X(4);
    f    = -phie*(1+S*(rij/re-1))*exp(-B*(rij/re-1));
end

function f = phi_CuCu(rij,X)
    phie = X(1);
    S    = X(2);
    B    = X(3);
    re   = X(4);
    f    = -phie*(1+S*(rij/re-1))*exp(-B*(rij/re-1));
en
```