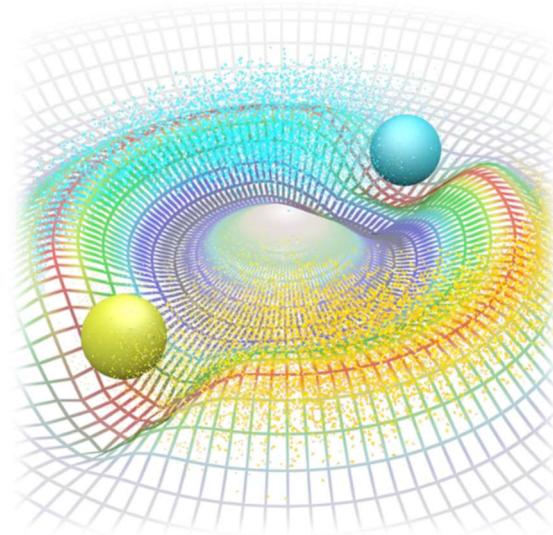


EAPOT

User Manual v1.2.0



2021-1-1
By B. N. Yao and R. F. Zhang
Beihang University

Contents

1.	Introduction	1
1.1.	Overview of EAPOT Studio	1
1.2.	Features of EAPOT Studio	3
1.3.	Startup of EAPOT Studio.....	4
1.4.	Framework of EAPOT Studio.....	5
1.5.	License statement.....	8
2.	Potentials of EAPOTs	11
2.1.	EAM model.....	11
2.2.	FS/TB model	14
2.3.	BOP model	16
2.4.	SW model.....	17
2.5.	MEAM model	18
3.	Potentials of EAPOTc	23
3.1.	EAM cross model.....	23
3.2.	FS/TB cross model.....	26
3.3.	BOP cross model.....	26
3.4.	SW cross model.....	27
3.5.	MEAM cross model	28
4.	Potential format.....	30
5.	Potential fitting	34
5.1.	Choose potential functions.....	36
5.2.	Prepare fitting targets	37
5.3.	Set minimization algorithms	41
5.4.	Add other constraints	44
5.5.	Run optimizations	46
5.6.	Potential fitting task manager	48
6.	Potential presetting.....	52
6.1.	Create structures for DFT calculations	53
6.2.	Parse properties from DFT calculations.....	55
7.	Structure modeling	58
8.	Potential plotting	60
8.1.	Potential function curves.....	61
8.2.	Potential energy surface	63

1. Introduction

1.1. Overview of EAPOT Studio

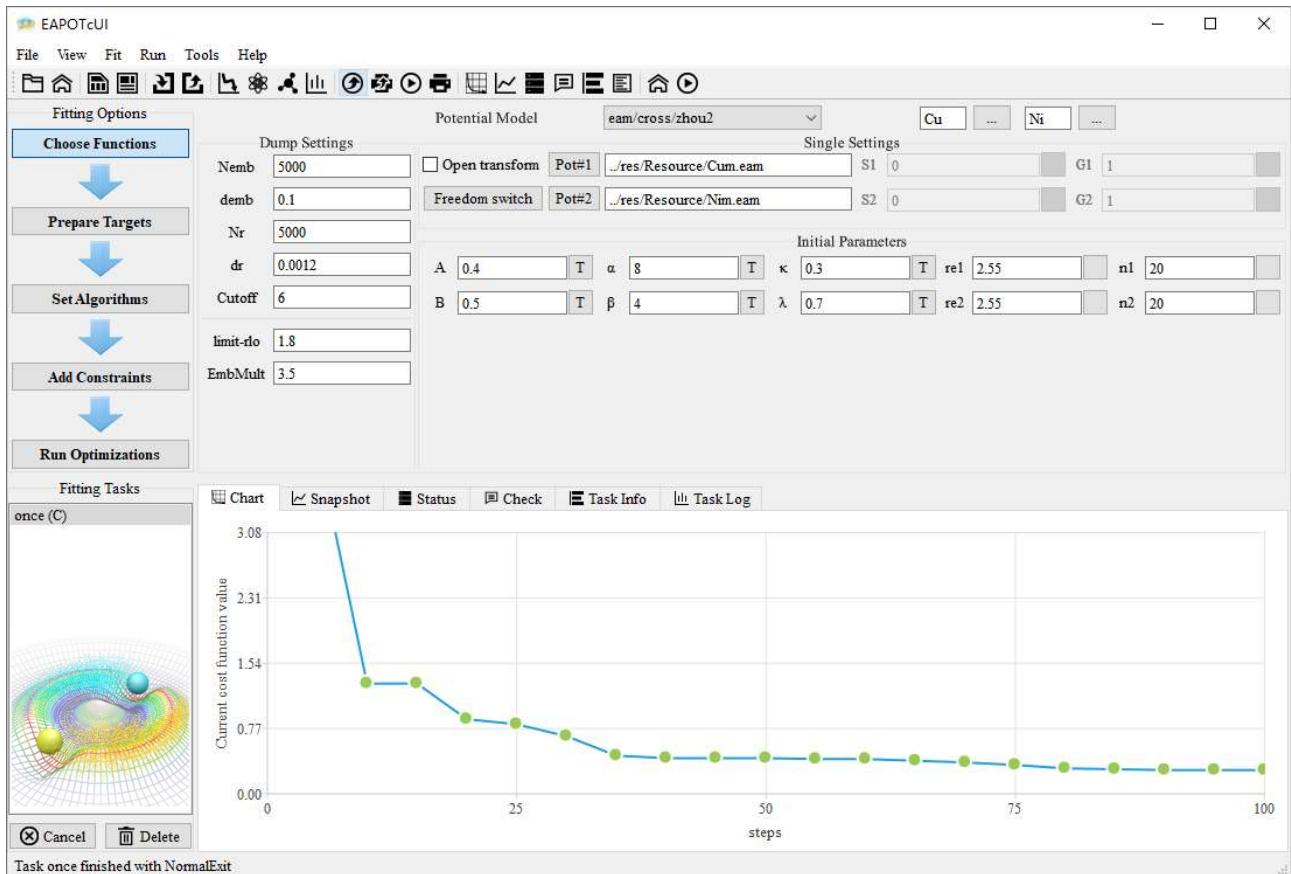


Fig. 1.1 Screenshot of the main graphical user interface of EAPOT Studio

EAPOT Studio (abbreviated as EAPOT) [1] is an integrated software platform developed by the research group of Professor Ruifeng Zhang at Beihang University in recent five years to build empirical interatomic potential for metallic and covalent solids. It includes single component system: EAPOTs [Copyright No.: 2019sr0190574], multiple component system: EAPOTc [Copyright No.: 2019sr1395181] and verification system: EACHK [Copyright No.: 2020sr0748849]. Through the convenient interactive

interface, it can realize the choice of various potential function forms, the setting of fitting target parameters, the integration of various local and global optimization algorithms, and the verification and analysis window so as to provide users with a whole process integration platform. The software platform not only provides the high-throughput automatic fitting process based on first-principles calculation but also provides the multiple combinations and multi-level objective optimization scheme of "energy-stress-force-elasticity," which ensures that the fitting strategy can meet the demands of different simulation scenarios. Compared with other similar software such as potfit, meamfit, atomicrex, etc., this software exhibits a series of advantages, e.g., more concise control interface, richer function form, more powerful optimization algorithms, and higher integration degree.

EAPOTs is the first core module of EAPOT Studio used to construct and evaluate the potential for single elemental solids. The supported potential functions include Zhou's EAM many-body potential function, Voter's EAM many-body potential function, Mishin's EAM many-body potential function, Finnis FS many-body potential function, Ackland's FS many-body potential function, Dai's FS many-body potential function, Rosato's TB many-body potential function, Li's TB many-body potential function, Stillinger-Weber three-body potential function, Tersoff's and Brenner's bond order potential (BOP) function, the MEAM many-body potential function of Baskes and the second-order MEAM many-body potential function of Lee.

EAPOTc is the second core module of EAPOT Studio to construct and evaluate the cross potential for compound solids. The supported EAM many-body potential

functions include those proposed by Morse, Johnson, Zhou, Demkowicz, Dai, Mishin, Ackland, and cubic spline. In addition, it also has good support for the construction and verification of the cross parts of the Stillinger-Weber three-body potential, Tersoff's and Brenner's BOP potential, and MEAM many-body potential.

Both core modules support the following fitting pipelines: model creation, energy fitting (cohesive energy, vacancy energy, interstitial energy, stacking fault energy, surface energy, etc.), force fitting, stress fitting, modulus fitting, etc. Different settings of fitting targets can meet the demands of different simulation scenarios. The optimization algorithms of both modules include not only local optimization algorithms, such as conjugate gradient method, simplex algorithm, and Powell algorithm but also support global optimization ones, such as particle swarm optimization, simulated annealing, and differential evolution, as well as their arbitrary combinations. The software platform can run on a single processor or multiple task distribution. It is written in highly portable C++ language. It has high running efficiency and easy to expand new functionalities and can be built into a library with an interface to EAPOTs or EAPOTc.

1.2. Features of EAPOT Studio

- a user-friendly GUI, as shown in Fig. 1.1
- a high-throughput (HT) flowchart for model construction with various modifications and data communication with first-principles calculations
- different simulation scenarios by means of multiple combinations of "energy-stress-force-elasticity" and multi-level objective, optimization schemes

- a variety of potential function forms for elemental metallic and covalent solids
- the custom definition of functional forms via Python script
- the robust training and validation system for high-efficiency evaluations

1.3. Startup of EAPOT Studio

Please open the software by clicking [bin/EAPOTxUI.exe](#), where x is s for EAPOTs and c for EAPOTc. Fig. 1.1 provides the overview of the EAPOTx file directory, which includes the EAPOTx file directory with several sub-directories, and Fig. 1.2 shows the general module hierarchy for the EAPOTs or EAPOTc.

Table 1.1 Overview of EAPOT Studio file directory

directory	description
/bin	Executable program
/example	Potential fitting examples
/lammmps	LAMMPS (for potential checking) [2]
/Python	Python interface
/Python/asut	Atomic Simulation Utility Toolkit
/Python/eapot	EAPOT Python interface
/Python/eachk	Potential checking module
/res	Resource files
/src	Program configuration source
/struct	Atomic structure files for potential checking
/tasks	Submitted potential fitting tasks

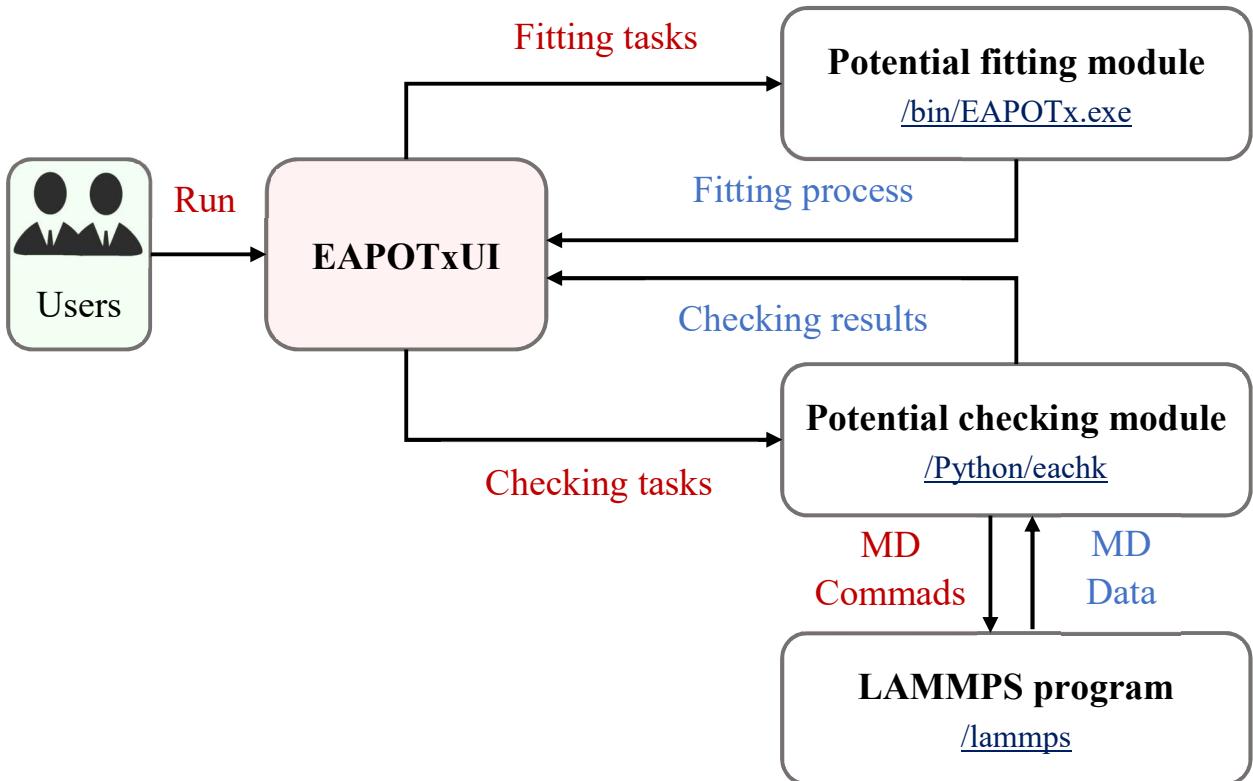


Fig. 1.2 The general module hierarchy for EAPOTs or EAPOTc

1.4. The framework of EAPOT Studio

There are four top windows for the framework of the EAPOTs or EAPOTc, which can be activated by the menu or toolbar buttons, as shown in Fig. 1.3:

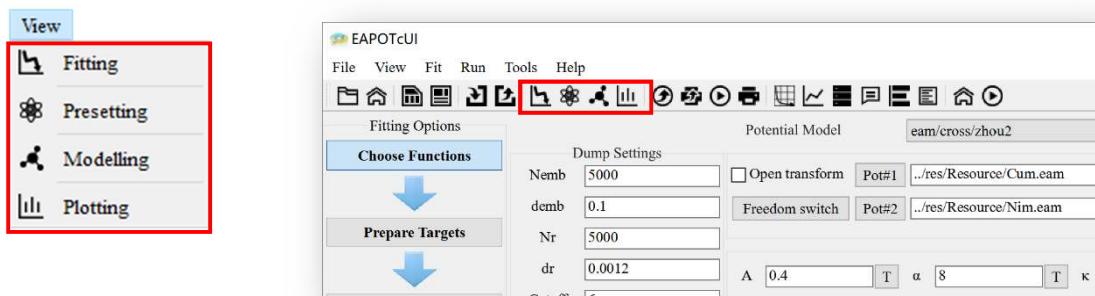


Fig. 1.3 Top switch window

1.Potential fitting window

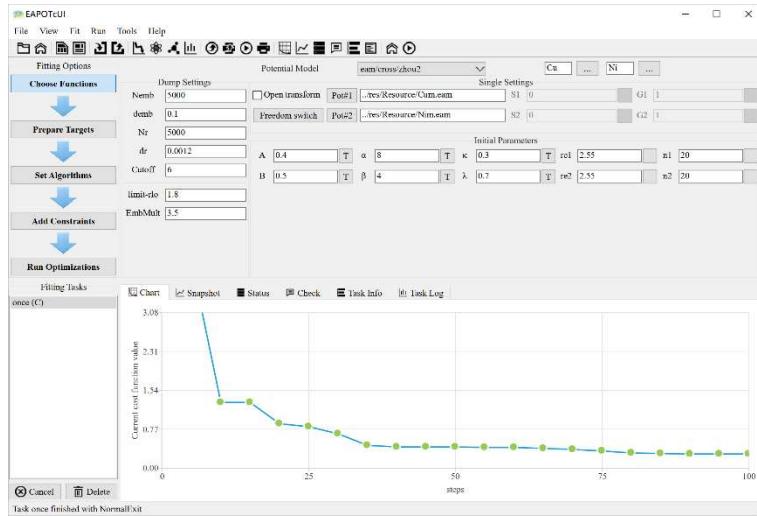


Fig. 1.4 Screenshot of the potential fitting window

The potential fitting page, as shown in Fig. 1.4, is mainly used for the potential fitting task creation and submission.

2.Potential presetting window

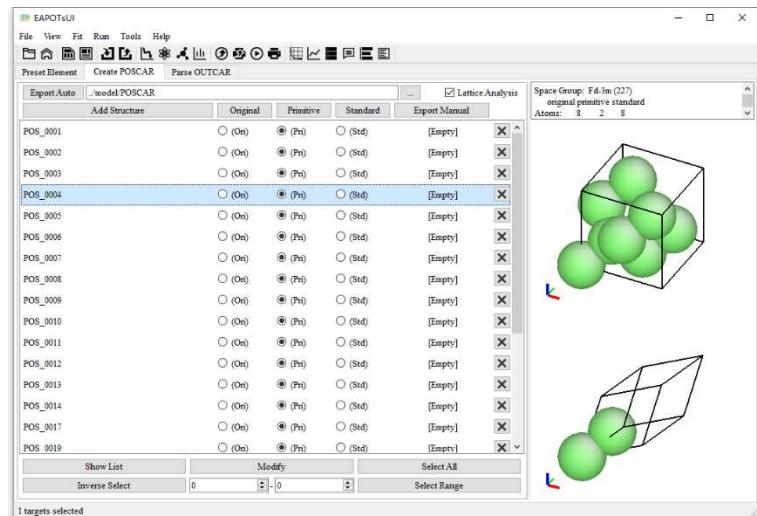


Fig. 1.5 Screenshot of the potential presetting window

The potential presetting window, as shown in Fig. 1.5, is used for the structure preparation and DFT calculations parsing.

3. Structure modeling window

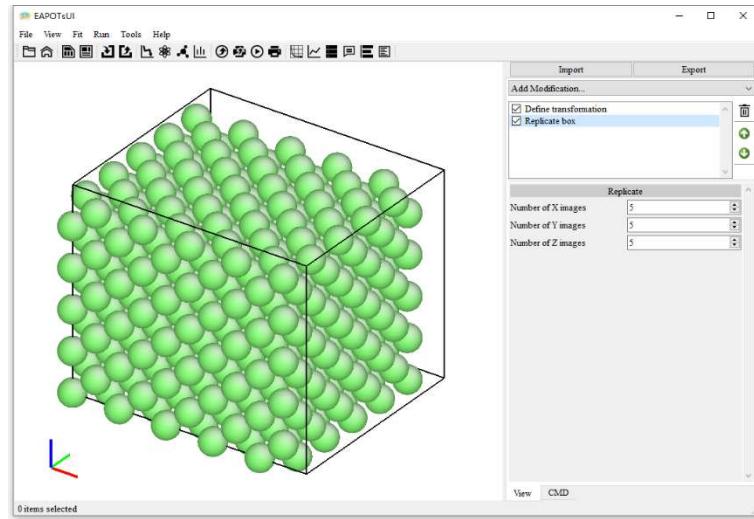


Fig. 1.6 Screenshot of the structure modeling window

The potential presetting window, as shown in Fig. 1.6, is used for fine editing of atomic structure.

4. Potential plotting window

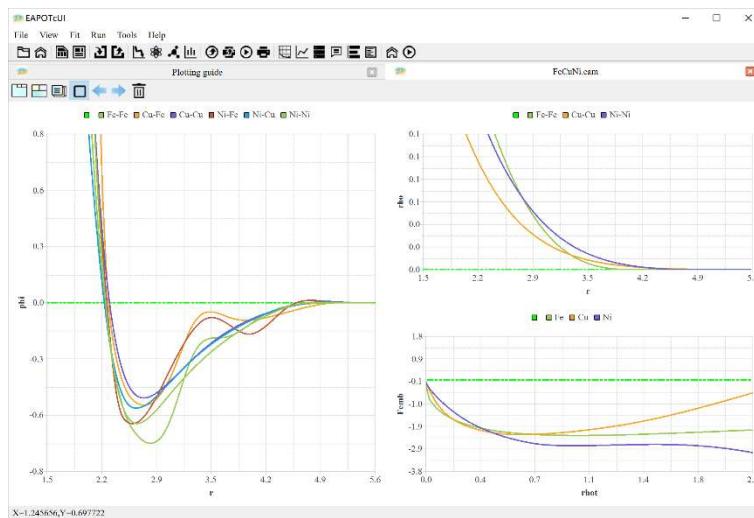


Fig. 1.7 Screenshot of the potential plotting window

The potential plotting window, as shown in Fig. 1.7, is used for potential curve visualization and energy surface visualization.

1.5. License statement

The basic modules of EAPOT Studio are currently copyrighted and distributed free of charge for academic, scientific, and educational, and non-commercial users with our permission. Commercial users may also use this software at no cost until a license is established. Some advanced modules implemented by special algorithms with technical support are also available when a payment contract is signed. This software is provided ‘as it is’ without any expressed or implied warranty.

The full EAPOT program package with the companion components may also be obtained upon request (email to zrfcms@buaa.edu.cn), and more details can be found at: <https://github.com/zrfcms/EAPOTs>, <https://github.com/zrfcms/EAPOTc>, or <http://www.eapot.top>.

Registration Form for EAPOT Studio

Name	
Title (e.g., Dr., Prof., Mr., Mrs, Ms)	
Affiliation (e.g., Dept., Univ., Co.)	
Address	
Country / Region	
License type (academic/educational/commercial)	
Email	
Fields	

Signature:

Date:

Reference

- [1] B.N. Yao, Z.R. Liu, R.F. Zhang, Computational Materials Science, 197 (2021) 110626.
- [2] S. Plimpton, Journal of Computational Physics, 117 (1995) 1-19.

2. Potentials of EAPOTs

2.1. EAM model

For metallic systems, the most widely used pair functional model for describing interatomic interactions in materials science is the embedded atom method (EAM) [1-3]. Several EAM potentials forms have been proposed with a similar expression, and the energy of an ion is defined by:

$$E_i = F \left(\sum_{i \neq j}^N \rho_{ij}(r_{ij}) \right) + \frac{1}{2} \sum_{i \neq j}^N \phi_{ij}(r_{ij}) \quad (2.1)$$

The equation has two parts: the first defines the energy for an ion i embed into the background electron charge density, while the second part is the pair interaction between the ions i and j separated by distance r_{ij} , and N is the number of neighbor atoms.

Included models are described in the following:

1. Zhou's potential functions

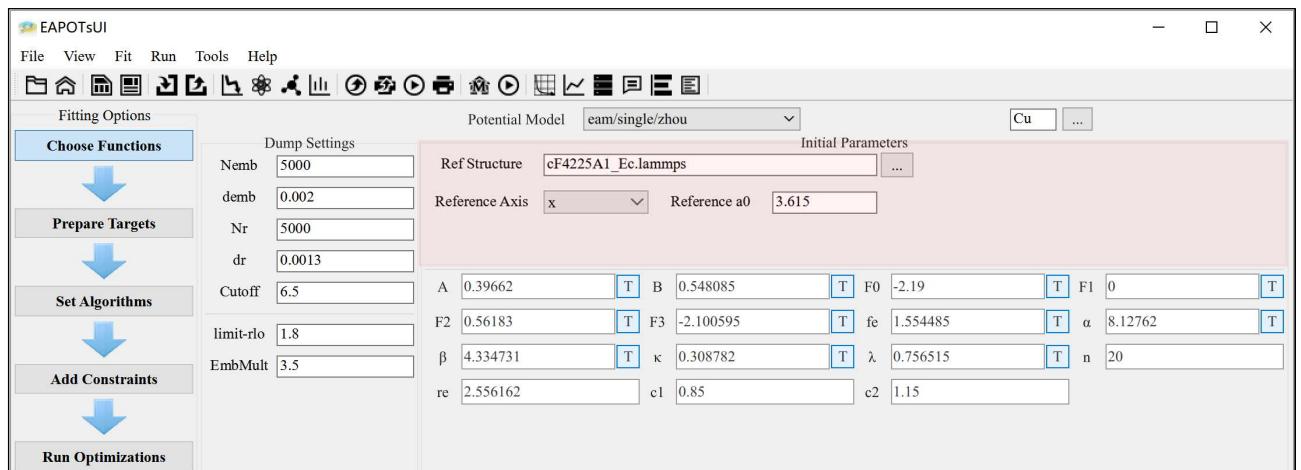


Fig. 2.1 Screenshot of the potential model by Zhou et al.

The generalized potential model proposed by Zhou et al. [4], which can provide a reasonable approximation of the interactions between different metal elements, can be calculated with electronic density and pair potential in

$$\rho_{ij}(r_{ij}) = \frac{f_e \exp\left[-\beta\left(\frac{r}{r_e} - 1\right)\right]}{1 + \left(\frac{r}{r_e} - \lambda\right)^n} \quad (2.2)$$

$$\phi_{ij}(r_{ij}) = \frac{A \exp\left[-\alpha\left(\frac{r}{r_e} - 1\right)\right]}{1 + \left(\frac{r}{r_e} - \kappa\right)^n} - \frac{B \exp\left[-\beta\left(\frac{r}{r_e} - 1\right)\right]}{1 + \left(\frac{r}{r_e} - \lambda\right)^n} \quad (2.3)$$

where $A, B, f_e, \alpha, \beta, \kappa$, and λ are the fitting parameters. The embedding energy function is expressed by three equations, which works well over a wide range of electron density ranges,

$$F(\rho) = \sum_{i=0}^3 F_{ni} \left(\frac{\rho}{\rho_n}\right)^i, \quad \rho < \rho_n, \quad \rho_n = 0.85\rho_e \quad (2.4)$$

$$F(\rho) = \sum_{i=0}^3 F_i \left(\frac{\rho}{\rho_e}\right)^i, \quad \rho_n \leq \rho < \rho_0, \quad \rho_0 = 1.15\rho_e \quad (2.5)$$

$$F(\rho) = F_e \left[1 - \ln\left(\frac{\rho}{\rho_s}\right)^\eta\right] \left(\frac{\rho}{\rho_s}\right)^\eta, \quad \rho_0 \leq \rho \quad (2.6)$$

where F_0, F_1, F_2 , and F_3 are the fitting parameters, and these equations must match values and slopes at their junctions for a smooth variation of the energy and force. Besides, the equilibrium electronic dentistry ρ_e is calculated from a reference structure and modified by those parameters in the red region of Fig. 2.1.

2. Voter's and Onat's potential functions

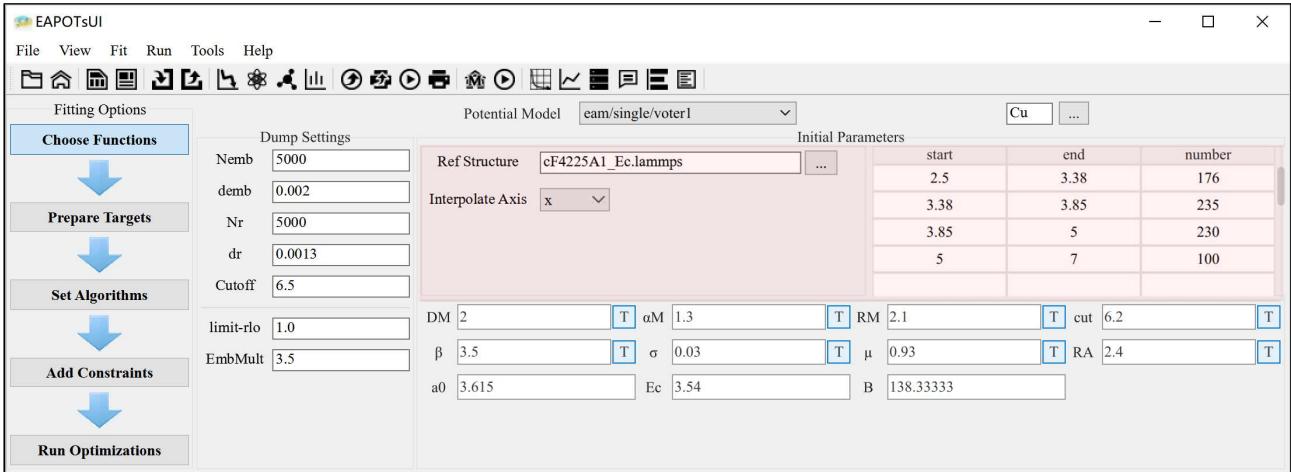


Fig. 2.2 Screenshot of the potential model by Voter et al.

The potential form proposed by Onat et al. [5], which follows the scheme presented by Voter et al. [6], can also produce reliable predictions for the structural properties and energetics as well. Following their scheme, the charge density is given by:

$$\rho_{ij}(r) = \tanh(20r^2) \left\{ r^6 (e^{-\beta r} + 2^9 e^{-2\beta r}) + \frac{\delta}{\mu R_A} e^{-\frac{1}{2}[\mu(r-R_A)]^2} + 0.1\delta e^{-\frac{1}{2}[\mu(r-R_A-0.5)]^2} \right\} \quad (2.7)$$

where β , σ , μ , R_A are the fitting parameters. The pair interaction, on the other hand, was taken as a Morse type potential

$$\phi_{ij}(r) = D_M [1 - e^{\alpha_M(r-R_M)}]^2 - D_M \quad (2.8)$$

here D_M , α_M , and R_M are fitting parameters. $\phi(r)$ and $\rho(r)$ have a smooth cut-off at $r = r_{cut}$ using the function

$$f_{smooth}(r) = h(r) - h(r_{cut}) + \left(\frac{r_{cut}}{m}\right) \left[1 - \left(\frac{r_{cut}}{m}\right)^m \right] \left(\frac{dh}{dr}\right) \Big|_{r=r_{cut}} \quad (2.9)$$

where $m = 20$. The embedding function was determined using the scheme introduced by Foiles et al. [3], where the total energy of fcc crystal is defined through the Rose universal equation of state (EOS) [7]:

$$E(x) = -E_{coh}(1 + x)e^{-x} \quad (2.10)$$

where x is a function of the lattice spacing and given by

$$x(a) = \left(\frac{a}{a_0} - 1 \right) \left(\frac{E_{coh}}{9\Omega_0 B} \right)^{-\frac{1}{2}} \quad (2.11)$$

Here, the parameters a_0 , Ω_0 , B and E_{coh} are the lattice constant, the atomic volume, the bulk modulus, and the cohesive energy, respectively, for the equilibrium fcc crystal at 0 K. To tabulate the embed energy (by interpolation), EAPOT need to sample the EOS equation based on a reference structure, as controlled by those parameters in the red region of Fig. 2.2, and the input format of the interpolation point is [start₁, end₁, point number₁], [start₂, end₂, point number₂] ... [start_n, end_n, point number_n].

2.2. FS/TB model

Besides of aforementioned EAM model, the pair functional models also include the tight-binding (TB) [8-10] and Finnis-Sinclair (FS) [11] model. In the TB model [9], the bonding energy in a metal system is approximatively proportional to the average bandwidth of the local electron density of the state. In particular, the second moment of the electron density of state can be written as a sum of squares of hopping integrals between the atoms and their neighbors [8, 10]:

$$E_{total} = - \sqrt{\sum_{i \neq j}^N \rho_{ij}(r_{ij}) + \frac{1}{2} \sum_{i \neq j}^N \phi_{ij}(r_{ij})} \quad (2.12)$$

where

$$\phi_{ij}(r_{ij}) = A \exp \left[-p \left(\frac{r}{r_0} - 1 \right) \right] \quad (2.13)$$

$$\rho_{ij}(r_{ij}) = \xi^2 \exp \left[-2q \left(\frac{r}{r_0} - 1 \right) \right] \quad (2.14)$$

here, the four parameters, p, q, A, and B, are assumed to depend only on the interacting atomic species i and j, and can be determined by fitting them to the physical properties. Although the TB potential initially seemed to apply to half-filled band metals, Ackland and Finnis have generalized the validity of the approach for band-filling metals [12]. The smoothly truncated potential form proposed by Li et al. [13, 14] was also implemented in EAPOT, which cut the long tail of the interatomic interaction by incorporating a truncation function into the pairwise and local electronic density function.

In the FS model [11], the total energy of an assembly of N atoms can be empirically written as

$$E_{total} = - \sqrt{\sum_{i \neq j}^N \rho_{ij}(r_{ij}) + \frac{1}{2} \sum_{i \neq j}^N \phi_{ij}(r_{ij})} \quad (2.15)$$

where

$$\phi_{ij}(r_{ij}) = (r - c)^2 (c_0 + c_1 r_{ij} + c_2 r_{ij}^2) \quad (2.16)$$

$$\rho_{ij}(r_{ij}) = (r - d)^2 \quad (2.17)$$

where c_0 , c_1 , and c_2 are the fitting parameters, c and d represent the cut-off radii for each function. The FS potential has been used successfully to calculate the properties in several bcc metals. However, as clarified by Ackland et al. [15], the FS potential appears to give unphysical results for properties involving small interatomic separation, e.g., at high pressure, where the calculated energy is much lower than the experimental

observation or theoretical prediction when the atoms are forced to be close together. To overcome this problem, Ackland et al. [15] added a product of a cubic and an exponential term to the pairwise function as an additional repulsive term:

$$\phi_{ij}(r_{ij}) = (r - c)^2(c_0 + c_1 r_{ij} + c_2 r_{ij}^2) + B(a_0 - r_{ij})^3 e^{-\alpha r_{ij}} \quad (2.18)$$

The adjusted potentials are shown to predict a more realistic pressure-volume relationship for the V, Nb, Ta, Mo, and W metals [15].

2.3. BOP model

For covalent materials, such as semiconductors, it is necessary to introduce those cluster potentials, e.g., bond order potential (BOP), that contains the angular item for directional forces. One of the most commonly used empirical interatomic potentials in covalent materials is the Tersoff BOP model [16, 17], where the total configuration energy of the center atom is:

$$E_i = \frac{1}{2} \sum_{i \neq j}^N f_C(r_{ij})[f_R(r_{ij}) + b_{ij}f_A(r_{ij})] \quad (2.19)$$

and

$$f_R(r_{ij}) = A \exp(-\lambda_1 r_{ij}) \quad (2.20)$$

$$f_A(r_{ij}) = -B \exp(-\lambda_2 r_{ij}) \quad (2.21)$$

$$f_C(r) = \begin{cases} 1 & , r < R - D \\ \frac{1}{2} - \frac{1}{2} \sin\left(\frac{\pi}{2} \frac{r - D}{D}\right) & , R - D < r < R + D \\ 0 & , r > R + D \end{cases} \quad (2.22)$$

$$b_{ij} = (1 + \beta^n \zeta_{ij}^n)^{-\frac{1}{2n}} \quad (2.23)$$

$$\zeta_{ij} = \sum_{k \neq i,j} f_C(r_{ik}) g(\theta_{ijk}) \exp[\lambda_3^m (r_{ij} - r_{ik})^m] \quad (2.24)$$

$$g(\theta) = \gamma_{ijk} \left(1 + \frac{c^2}{d^2} - \frac{c^2}{[d^2 + (\cos\theta - h)^2]} \right) \quad (2.25)$$

where $f_R(r_{ij})$ is a two-body term, $f_A(r_{ij})$ represents the three-body interactions, $h = \cos\theta_0$, $g(\theta)$ is the modified angular-dependent term, and $f_C(r)$ is the cut-off function. Here i, j , and k label the atoms of the system, r_{ij} is the length of the ij bond, and θ_{ijk} is the bond angle between bonds ij and ik . The summations in the formula are over all neighbors j and k of atom i within a cut-off distance equal to $R + D$, and it has been shown that the simulation results with Tersoff BOP potential reproduce well the experimental data [18], including defect structures, activation energies for defect motion, and coupling to strain.

The Brenner BOP potential [19] is a reformulation of the Tersoff BOP potential [16, 17], and the difference lies in the two-body parameters A, B, λ_1 , and λ_2 . These parameters have been rewritten in terms of the more physical formats, r_0, D_0, μ , and S , which relate to dimer properties and the Pauling relation:

$$f_R(r_{ij}) = \frac{D_0}{S-1} \exp[-\mu\sqrt{2S}(r_{ij} - r_0)] \quad (2.26)$$

$$f_A(r_{ij}) = \frac{SD_0}{S-1} \exp\left[-\mu \sqrt{\frac{2}{S}}(r_{ij} - r_0)\right] \quad (2.27)$$

2.4. SW model

The Stillinger-Weber (SW) potentials [20] have been extensively used to investigate semiconductor materials, which can also reproduce the buckling structure

and lattice thermal conductivity. For particles with no net charge, the potentials are based on the idea that the force is repulsive if the atoms are very close and attractive, vanishing smoothly to zero when the distance between them increases. Compared with the Tersoff [16, 17] BOP potential, the Stillinger-Weber potential has a simpler form and fewer parameters to fit, so it is faster and more appropriate for parameter fitting:

$$E_i = \sum_i \sum_{j>i} \phi_2(r_{ij}) + \sum_i \sum_{j\neq i} \sum_{k>j} \phi_3(r_{ij}, r_{ik}, \theta_{ijk}) \quad (2.28)$$

$$\phi_2(r_{ij}) = A_{ij}\epsilon_{ij} \left[B_{ij} \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{p_{ij}} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{q_{ij}} \right] \exp \left(\frac{\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}} \right) \quad (2.29)$$

$$\begin{aligned} \phi_3(r_{ij}, r_{ik}, \theta_{ijk}) &= \lambda_{ijk}\epsilon_{ijk} [\cos\theta_{ijk} - \cos\theta_{0ijk}]^2 \\ &\exp \left(\frac{\gamma_{ij}\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}} \right) \exp \left(\frac{\gamma_{ik}\sigma_{ik}}{r_{ik} - a_{ik}\sigma_{ik}} \right) \end{aligned} \quad (2.30)$$

Notice that the potential favors pairs of bonds with the desired tetrahedral angle when $\cos\theta_0 = \frac{1}{3}d$ the other parameters were chosen to ensure that the diamond structure is the most stable at low pressure.

2.5. MEAM model

The pair functional models mentioned above are all radially symmetric, which do not reflect the directional nature of the bonding. To improve the performance of these potentials in this aspect, Baskes [21] and Lee [22, 23] et al. have proposed the modified EAM (MEAM) potentials, where the local electron density is computed by considering the directionality of the bonding. The total MEAM configuration energy is a sum of direct contributions from all atoms:

$$E_i = F(\bar{\rho}_i) + \frac{1}{2} \sum_{i\neq j} \phi_{ij}(r_{ij}) \quad (2.31)$$

The first term corresponds to the embedding energy of the center atom, and the second term is the pair potential between atoms. The embedding function has the form

$$F(\bar{\rho}_i) = A_i E_i^0 \bar{\rho}_i \ln \bar{\rho}_i \quad (2.32)$$

and

$$\bar{\rho}_i = \frac{\bar{\rho}_i^{(0)}}{\rho_i^0} G_i \left(\sum_{k=1}^3 t_i^{(k)} \left(\frac{\bar{\rho}_i^{(k)}}{\bar{\rho}_i^{(0)}} \right)^2 \right) \quad (2.33)$$

$G_i(x)$ can be chosen as $\sqrt{1+x}$ or $\exp\left(\frac{x}{2}\right)$, and ρ_i^0 is the electronic scale factor.

Each partial electron density term has the following form:

$$\left(\bar{\rho}_i^{(0)} \right)^2 = \left[\sum_{i \neq j} \rho_j^{a(0)}(r_{ij}) \right]^2 \quad (2.34)$$

$$\left(\bar{\rho}_i^{(1)} \right)^2 = \sum_{\alpha}^3 \left[\sum_{j \neq i} \frac{r_{ij\alpha}}{r_{ij}} \rho_j^{a(1)}(r_{ij}) \right]^2 \quad (2.35)$$

$$\begin{aligned} \left(\bar{\rho}_i^{(2)} \right)^2 &= \sum_{\alpha}^3 \sum_{\beta}^3 \left[\sum_{j \neq i} \frac{r_{ij\alpha} r_{ij\beta}}{r_{ij}^2} \rho_j^{a(2)}(r_{ij}) \right]^2 \\ &\quad - \frac{1}{3} \left[\sum_{j \neq i} \rho_j^{a(2)}(r_{ij}) \right]^2 \end{aligned} \quad (2.36)$$

$$\begin{aligned} \left(\bar{\rho}_i^{(3)} \right)^2 &= \sum_{\alpha}^3 \sum_{\beta}^3 \sum_{\gamma}^3 \left[\sum_{j \neq i} \frac{r_{ij\alpha} r_{ij\beta} r_{ij\gamma}}{r_{ij}^3} \rho_j^{a(3)}(r_{ij}) \right]^2 \\ &\quad - \frac{3}{5} \sum_{\alpha}^3 \left[\sum_{j \neq i} \frac{r_{ij\alpha}}{r_{ij}} \rho_j^{a(3)}(r_{ij}) \right]^2 \end{aligned} \quad (2.37)$$

Here, $\bar{\rho}_i^{(k)}$ represent atomic electron densities from j atom at a distance r_{ij} from the center atom and $r_{ij\alpha}$ is the α component of the distance vector. The specific forms

above are chosen so that the partial background electron densities are invariant to lattice translation and rotation, scale simply with atomic electron density for homogeneous deformation, and they are computed as

$$\rho_i^{a(k)}(r_{ij}) = \rho_{i0} \exp\left[-\beta\left(\frac{r_{ij}}{r_i^0} - 1\right)\right] f_c\left(\frac{r_c - r_{ij}}{\Delta r}\right) \prod_{j \neq i} S_{ijk} \quad (2.38)$$

where

$$S_{ijk} = f_c\left(\frac{C_{ijk} - C_{min,ijk}}{C_{max,ijk} - C_{min,ijk}}\right) \quad (2.39)$$

$$C_{ijk} = 1 + 2 \frac{r_{ij}^2 r_{ik}^2 + r_{ij}^2 r_{jk}^2 + r_{ij}^4}{r_{ij}^4 - (r_{ik}^2 - r_{jk}^2)^2} \quad (2.40)$$

As clarified by Baskes et al. [21], the specific forms above can provide an improved description of systems where directional bonding is essential, such as semiconductors and elements from the middle of the transition metal series, where the origin EAM potentials have several shortcomings. Here we only introduce a short description of the inclusion of the angular items, and the complete MEAM expression can be accessed in Ref. [21-23]. Considering that the original MEAM model is not quite successful for bcc metals, the second nearest-neighbor MEAM proposed by Lee et al. [22, 23] was also implemented in EAPOT, which can reasonably reproduce the surface properties of the bcc transition metals.

Reference

- [1] M.S. Daw, M.I. Baskes, Physical Review B, 29 (1984) 6443-6453.
- [2] S.M. Foiles, Physical Review B, 32 (1985) 7685-7693.
- [3] S.M. Foiles, M.I. Baskes, M.S. Daw, Physical Review B, 33 (1986) 7983-7991.
- [4] X.W. Zhou, R.A. Johnson, H.N.G. Wadley, Physical Review B, 69 (2004) 144113.
- [5] B. Onat, S. Durukanoğlu, Journal of Physics: Condensed Matter, 26 (2013) 035404.
- [6] A.F. Voter, S.P. Chen, MRS Proceedings, 82 (1986) 175.
- [7] J.H. Rose, J.R. Smith, F. Guinea, J. Ferrante, Physical Review B, 29 (1984) 2963-2969.
- [8] F. Cleri, V. Rosato, Physical Review B, 48 (1993) 22-33.
- [9] F. Ducastelle, F. Cyrot-Lackmann, Journal of Physics and Chemistry of Solids, 31 (1970) 1295-1306.
- [10] V. Rosato, M. Guillope, B. Legrand, Philosophical Magazine A, 59 (1989) 321-336.
- [11] M.W. Finnis, J.E. Sinclair, Philosophical Magazine A, 50 (1984) 45-55.
- [12] G.J. Ackland, M.W. Finnis, V. Vitek, Journal of Physics F: Metal Physics, 18 (1988) L153-L157.
- [13] J.H. Li, X.D. Dai, T.L. Wang, B.X. Liu, Journal of Physics: Condensed Matter, 19 (2007) 086228.
- [14] J.H. Li, Y. Dai, X.D. Dai, T.L. Wang, B.X. Liu, Computational Materials Science, 43 (2008) 1207-1215.
- [15] G.J. Ackland, R. Thetford, Philosophical Magazine A, 56 (1987) 15-30.
- [16] J. Tersoff, Physical Review B, 37 (1988) 6991-7000.

- [17] J. Tersoff, Physical Review B, 38 (1988) 9902-9905.
- [18] J. Tersoff, Physical Review Letters, 64 (1990) 1757-1760.
- [19] D.W. Brenner, Physical Review B, 42 (1990) 9458-9471.
- [20] F.H. Stillinger, T.A. Weber, Physical Review B, 31 (1985) 5262-5271.
- [21] M.I. Baskes, Physical Review B, 46 (1992) 2727-2742.
- [22] B.-J. Lee, M.I. Baskes, Physical Review B, 62 (2000) 8564-8567.
- [23] B.-J. Lee, M.I. Baskes, H. Kim, Y. Koo Cho, Physical Review B, 64 (2001) 184102.

3. Potentials of EAPOTc

3.1. EAM cross model

The details and output of the EAM model for single elemental solids are described in the previous chapter. In the generation of the potential of compound solids, the cross part between different elements must be defined and fitted to the properties of compounds. For that, two types of transformations are performed on the pure element potential functions, i.e., shifting operation by

$$F'_i(\bar{\rho}) = F_i(\bar{\rho}) + G_i \rho, \phi'_{ii}(r) = \phi_{ii}(r) - 2G_i \bar{\rho} \quad (3.1)$$

and scaling operation by

$$\rho'_i(r) = S_i \rho_i(r), F'_i(\bar{\rho}) = F_i(\bar{\rho}/S_i) \quad (3.2)$$

where i stands for one single elemental potentials, S_i and G_i are fitting parameters, as shown in

Fig. 3.1. Equations (3.1) and (3.2) ensure that the single elemental energies do not change during the transformations while the compound energy does, and thus helps increase the quality of the compound potential in the fitting. It is also important to note that the transformations should be used in sequence, in which the scaling operation is applied after the shifting operation.

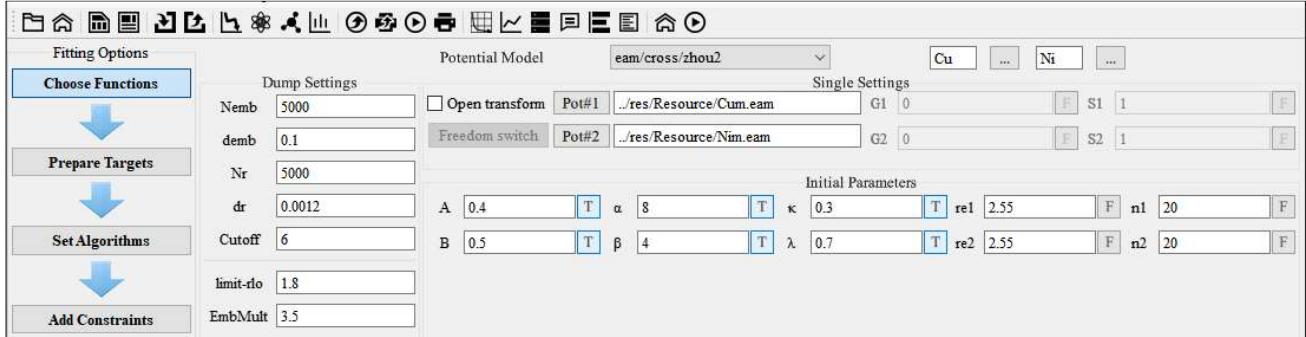


Fig. 3.1 Screenshot of EAM cross model

As shown in the "Single Settings" group in

Fig. 3.1, two single elemental potentials must be selected at first, including the following supported models:

1. The Morse function

The cross potential was taken to be a Morse type potential

$$\phi_{ij}(r) = D_M \left[1 - e^{\alpha_M(r-R_M)} \right]^2 - D_M \quad (3.3)$$

here D_M , α_M , and R_M are fitting parameters with a smooth cut-off at $r = r_{cut}$ using the function

$$f_{smooth}(r) = h(r) - h(r_{cut}) + \left(\frac{r_{cut}}{m} \right) \left[1 - \left(\frac{r_{cut}}{m} \right)^m \right] \left(\frac{dh}{dr} \right) \Big|_{r=r_{cut}} \quad (3.4)$$

2. The 1st version of Zhou's function

The cross potential [1] is expressed as

$$\begin{aligned} \phi_{ij}(r_{ij}) = & \frac{A_1 \exp \left[-\alpha_1 \left(\frac{r}{r_{e11}} - 1 \right) \right]}{1 + \left(\frac{r}{r_{e1}} - \kappa_1 \right)^{n_{11}}} - \frac{B_1 \exp \left[-\beta_1 \left(\frac{r}{r_{e12}} - 1 \right) \right]}{1 + \left(\frac{r}{r_{e2}} - \lambda_1 \right)^{n_{12}}} \\ & + \frac{A_2 \exp \left[-\alpha_2 \left(\frac{r}{r_{e1}} - 1 \right) \right]}{1 + \left(\frac{r}{r_{e21}} - \kappa_2 \right)^{n_{12}}} - \frac{B_2 \exp \left[-\beta_2 \left(\frac{r}{r_{e2}} - 1 \right) \right]}{1 + \left(\frac{r}{r_{e22}} - \lambda_2 \right)^{n_{22}}} \end{aligned} \quad (3.5)$$

3. The 2nd version of Zhou's function

The cross potential [1] is expressed as

$$\phi_{ij}(r_{ij}) = \frac{A \exp\left[-\alpha\left(\frac{r}{r_{e1}} - 1\right)\right]}{1 + \left(\frac{r}{r_{e1}} - \kappa\right)^{n1}} - \frac{B \exp\left[-\beta\left(\frac{r}{r_{e2}} - 1\right)\right]}{1 + \left(\frac{r}{r_{e2}} - \lambda\right)^{n2}} \quad (3.6)$$

4. The 3rd version of Zhou's function

The cross potential is expressed as

$$\phi_{ij}(r_{ij}) = D\{\beta \exp[-\alpha(r_0 - r_0)] - \alpha \exp[-\beta(r_0 - r_0)]\}\psi(r) \quad (3.7)$$

$$\psi(r) = \begin{cases} 1 & r < r_c - r_s \\ \frac{1}{2}[1 + \cos(\pi(r - r_c + r_s)/r_s)] & r_c - r_s < r \leq r_c \\ 0 & r_c \leq r \end{cases} \quad (3.8)$$

5. Ackland's function

The cross potential [2] is expressed as

$$\phi_{ij}(r_{ij}) = (r - c)^2(c_0 + c_1r_{ij} + c_2r_{ij}^2) + B(a_0 - r_{ij})^3e^{-\alpha r_{ij}} \quad (3.9)$$

6. Dai's function

The cross potential is expressed as

$$\phi_{ij}(r_{ij}) = (r - r_c)^m(c_0 + c_1r_{ij} + c_2r_{ij}^2 + c_3r_{ij}^3 + c_4r_{ij}^4) \quad (3.10)$$

7. Johnson's function

The cross potential in "direct" mode is expressed as

$$\phi_{ij}(r_{ij}) = \frac{1}{2}[w_1\phi_{ii}(r_{ij}) + w_2\phi_{jj}(r_{ij})] \quad (3.11)$$

The cross potential in "weight" mode [3] is expressed as

$$\phi_{ij}(r_{ij}) = \frac{1}{2}\left[w_1 \frac{\rho_{jj}(r_{ij})}{\rho_{ii}(r_{ij})} \phi_{ii}(r_{ij}) + w_2 \frac{\rho_{ii}(r_{ij})}{\rho_{jj}(r_{ij})} \phi_{jj}(r_{ij})\right] \quad (3.12)$$

If "Dof" is 1, then $w_2 = 2 - w_1$ as shown in Fig. 3.2.

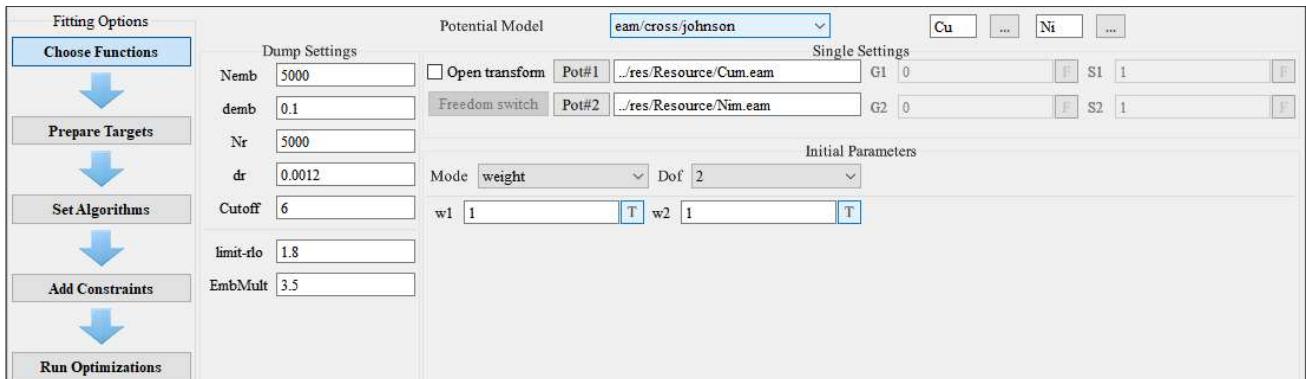


Fig. 3.2 Screenshot of Johnson's cross model

3.2. FS/TB cross model

This model has not yet been implemented in the UI, and one can achieve it by Python scripts.

3.3. BOP cross model

The details and output of BOP models in Tersoff's and Brenner's format for single elemental solid are described in the previous chapter. Table 3.1 and Table 3.2 show the default parameters of Tersoff's BOP model [4, 5] and Brenner's BOP model [6], respectively (Note: Python script supports custom mapping).

Table 3.1 Default available parameters of Tersoff's BOP model

i	j	k	m	γ	λ_3	c	d	h	n	β	λ_2	B	R	D	λ_1	A
A	A	A	m	γ	λ_3	c(A)	d(A)	h(A)	n(A)	$\beta(A)$	$\lambda_2(A)$	B(A)	R(A)	D(A)	$\lambda_1(A)$	A(A)
A	A	B	m	γ	λ_3	c(A)	d(A)	h(A)					R(A)	D(A)		
A	B	A	m	γ	λ_3	c(AB)	d(AB)	h(AB)					R(AB)	D(AB)		
A	B	B	m	γ	λ_3	c(AB)	d(AB)	h(AB)	n(AB)	$\beta(AB)$	$\lambda_2(AB)$	B(AB)	R(AB)	D(AB)	$\lambda_1(AB)$	A(AB)
B	A	A	m	γ	λ_3	c(AB)	d(AB)	h(AB)	n(AB)	$\beta(AB)$	$\lambda_2(AB)$	B(AB)	R(AB)	D(AB)	$\lambda_1(AB)$	A(AB)
B	A	B	m	γ	λ_3	c(AB)	d(AB)	h(AB)					R(AB)	D(AB)		

B	B	A	m	γ	λ_3	c(B)	d(B)	h(B)		R(B)	D(B)
B	B	B	m	γ	λ_3	c(B)	d(B)	h(B)	n(B) $\beta(B)$ $\lambda_2(B)$ B(B)	R(B) D(B)	$\lambda_1(B)$ A(B)

Table 3.2 Default available parameters of Brenner's BOP model

i	j	k	m	γ	λ_3	c	d	h	n	β	μ	S	R	D	R0	D0
A	A	A	m	γ	λ_3	c(A)	d(A)	h(A)	n(A) $\beta(A)$ $\mu(A)$	S(A)	R(A) D(A)	R0(A) D0(A)				
A	A	B	m	γ	λ_3	c(A)	d(A)	h(A)			R(A) D(A)					
A	B	A	m	γ	λ_3	c(AB)	d(AB)	hAB)			R(AB) D(AB)					
A	B	B	m	γ	λ_3	c(AB)	d(AB)	h(AB)	n(AB) $\beta(AB)$ $\mu(AB)$	S(AB)	R(AB) D(AB)	R0(AB) D0(AB)				
B	A	A	m	γ	λ_3	c(AB)	d(AB)	h(AB)	n(AB) $\beta(AB)$ $\mu(AB)$	S(AB)	R(AB) D(AB)	R0(AB) D0(AB)				
B	A	B	m	γ	λ_3	c(AB)	d(AB)	h(AB)			R(AB) D(AB)					
B	B	A	m	γ	λ_3	c(B)	d(B)	h(B)			R(B) D(B)					
B	B	B	m	γ	λ_3	c(B)	d(B)	h(B)	n(B) $\beta(B)$ $\mu(B)$	S(B)	R(B) D(B)	R0(B) D0(B)				

3.4. SW cross model

The details and output of the Stillinger-Weber [7] model for single elemental solids is described in the previous chapter. Table 3.3 shows the default parameters of the SW model (Note: Python script supports custom mapping).

Table 3.3 Default available parameters of SW model

			ε	σ	a	λ	γ	cos00	A	B	p	q	tol
A	A	A	$\varepsilon(A)$	$\sigma(A)$	a(A)	λ	γ	cos00(A)	A	B	p	q	tol
A	A	B	$\varepsilon(ijk)$			λ	γ	cos00(A)	A	B	p	q	tol
A	B	A	$\varepsilon(ijk)$			λ	γ	cos00(A)	A	B	p	q	tol
A	B	B	$\varepsilon(ijj)$	$\sigma(AB)$	a(AB)	λ	γ	cos00(A)	A	B	p	q	tol
B	A	A	$\varepsilon(ijj)$	$\sigma(AB)$	a(AB)	λ	γ	cos00(B)	A	B	p	q	tol

B	A	B	$\epsilon(ijk)$		λ	γ	$\cos\theta(B)$	A	B	p	q	tol
B	B	A	$\epsilon(ijk)$		λ	γ	$\cos\theta(B)$	A	B	p	q	tol
B	B	B	$\epsilon(B)$	$\sigma(B)$	a(B)	λ	γ	$\cos\theta(B)$	A	B	p	tol

3.5. MEAM cross model

The details and output of the MEAM model [8-10] for single elemental solids are described in the previous chapter. The MEAM parameter file contains settings that override or complement the library file settings. Examples of such parameter files are in the potentials directory with a “.meam” suffix, and a MEAM parameter editor was introduced to edit these parameters, as shown in Fig. 3.3.

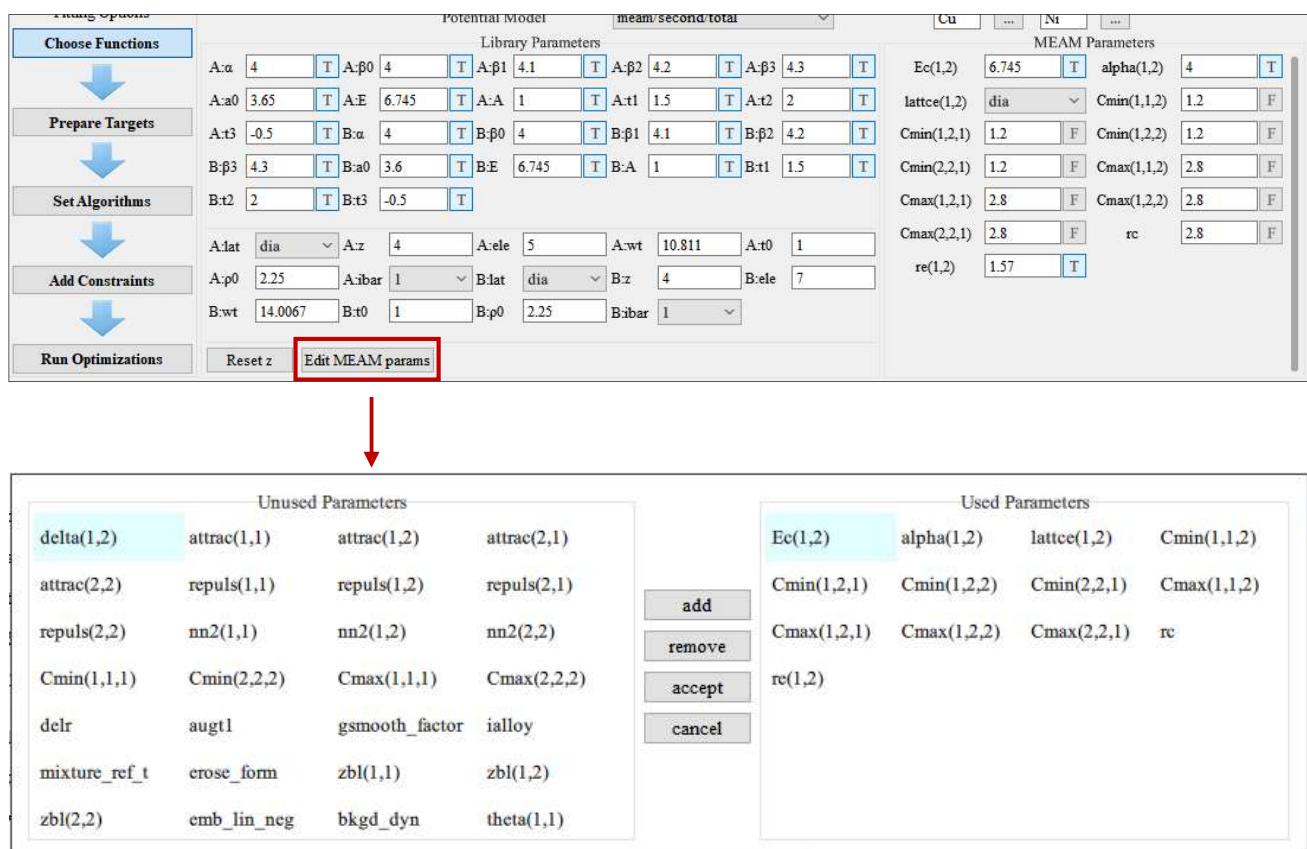


Fig. 3.3 MEAM parameter editor

Reference

- [1] X.W. Zhou, R.A. Johnson, H.N.G. Wadley, Physical Review B, 69 (2004) 144113.
- [2] G.J. Ackland, R. Thetford, Philosophical Magazine A, 56 (1987) 15-30.
- [3] R.A. Johnson, Physical Review B, 39 (1989) 12554.
- [4] J. Tersoff, Physical Review B, 37 (1988) 6991-7000.
- [5] J. Tersoff, Physical Review B, 38 (1988) 9902-9905.
- [6] D.W. Brenner, Physical Review B, 42 (1990) 9458-9471.
- [7] F.H. Stillinger, T.A. Weber, Physical Review B, 31 (1985) 5262-5271.
- [8] M.I. Baskes, Physical Review B, 46 (1992) 2727-2742.
- [9] B.-J. Lee, M.I. Baskes, Physical Review B, 62 (2000) 8564-8567.
- [10] B.-J. Lee, M.I. Baskes, H. Kim, Y. Koo Cho, Physical Review B, 64 (2001) 184102.

4. Potential format

As regards EAM, FS, and TB models, EAPOT will output the potential in a specific tabulated format, as controlled by those parameters in the “Dump Settings” frame as shown in Fig. 4.1 (marked by cyan color).

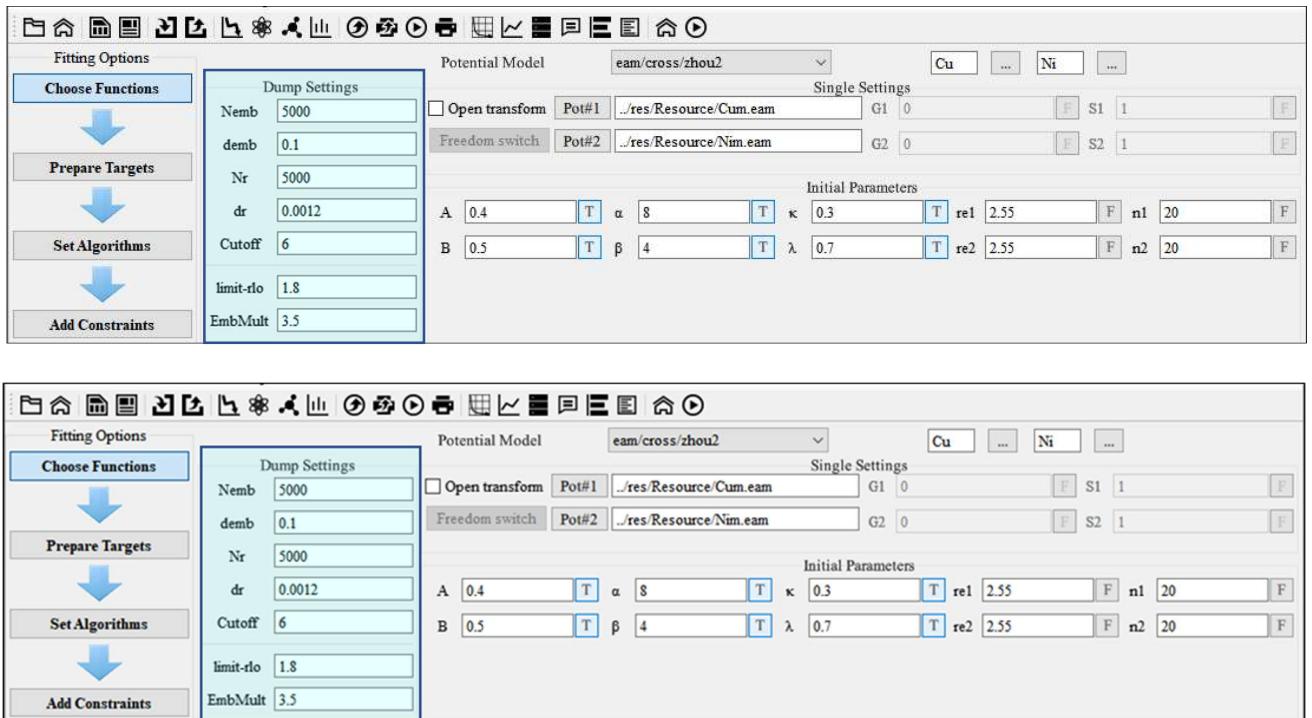


Fig. 4.1 EAM, FS, and TB Dump Settings

For EAM potentials, the potential functions are tabulated in the LAMMPS *eam/alloy* format, which is the same as the DYNAMO multi-element setfl format. As an example, the res/Resource/FeCuNi.eam file has tabulated EAM values for three elements and their alloy interactions: Fe, Cu, and Ni and formatted as follows:

- lines 1,2,3 = comments (ignored)
- line 4: Nelements Element1 Element2 ... ElementN
- line 5: Nrho, drho, Nr, dr, cut-off

In an *eam/alloy* file, the element name (Fe, Cu, Ni) of each element must be added to the line in the order, the elements appear in the file. On line 5, Nrho and Nr are the number of tabulated values in the subsequent arrays, drho and dr are the spacing in density and distance space for the values in those arrays, and the cut-off (in Angstroms) is a global value, valid for all pairwise interactions for all element pairings.

Following the five header lines are Nelements sections, one for each element, each with the following format:

- line 1 = atomic number, mass, lattice constant, lattice-type (e.g. FCC)
- embedding function F(rho) (Nrho values)
- density function rho(r) (Nr values)

The lattice constant is in the unit of Angstroms, and the F and rho arrays are unique to a single element. Following the Nelements sections, Nr values for each pair potential phi(r) array are listed for all i, j element pairs in the same format as other arrays. Since these interactions are symmetric ($i, j = j, i$) only phi arrays with $i \geq j$ are listed, in the following order: $i, j = (1,1), (2,1), (2,2), (3,1), (3,2), (3,3), (4,1), \dots, (N_{elements}, N_{elements})$. The tabulated values for each phi function are listed as $r^*\phi$ (in units of eV-Angstroms) since they are for atom pairs.

For FS and TB potentials, the tabulated files include more information than the *eam/alloy* format files, in that the i, j density functionals for all pairs of elements are included as required by the Finnis-Sinclair formulation, formatted as follows:

- lines 1,2,3 = comments (ignored)
- line 4: $N_{elements}$ Element1 Element2 ... ElementN

- line 5: Nrho, drho, Nr, dr, cut-off
- The 5-line header section is identical to an EAM file.

Following the header are Nelements sections, one for each element, each with the following format:

- line 1 = atomic number, mass, lattice constant, lattice-type (e.g. FCC)
- embedding function F(rho) (Nrho values)
- density function for the element at element 1 (Nr values)
- density function for the element at element 2
- ...
- density function for the element at element N

The units of these quantities in line 1 are the same as those for EAM files. Note that the rho(r) arrays in Finnis-Sinclair can be asymmetric $\rho_{ij}(r) \neq \rho_{ji}(r)$ so there are $N_{elements}^2$ of them listed in the file. Following the Nelements sections, Nr values for each pair potential phi(r) array are listed in the same manner ($r^*\phi$, units of eV-Angstroms) as in EAM files. Note that in Finnis-Sinclair potentials, the phi(r) arrays are still symmetric, so only phi arrays for $i \geq j$ are listed.

For Stillinger-Weber potential, the analytic bond-order potential (BOP) in the Tersoff or Brenner formats, and the modified EAM (MEAM) models, EAPOT will directly output potential parameters formatted as a series of entries, and more details of these potentials can be found in the LAMMPS [1] manual.

References:

- [1] S. Plimpton, Journal of Computational Physics, 117 (1995) 1-19.

5. Potential fitting

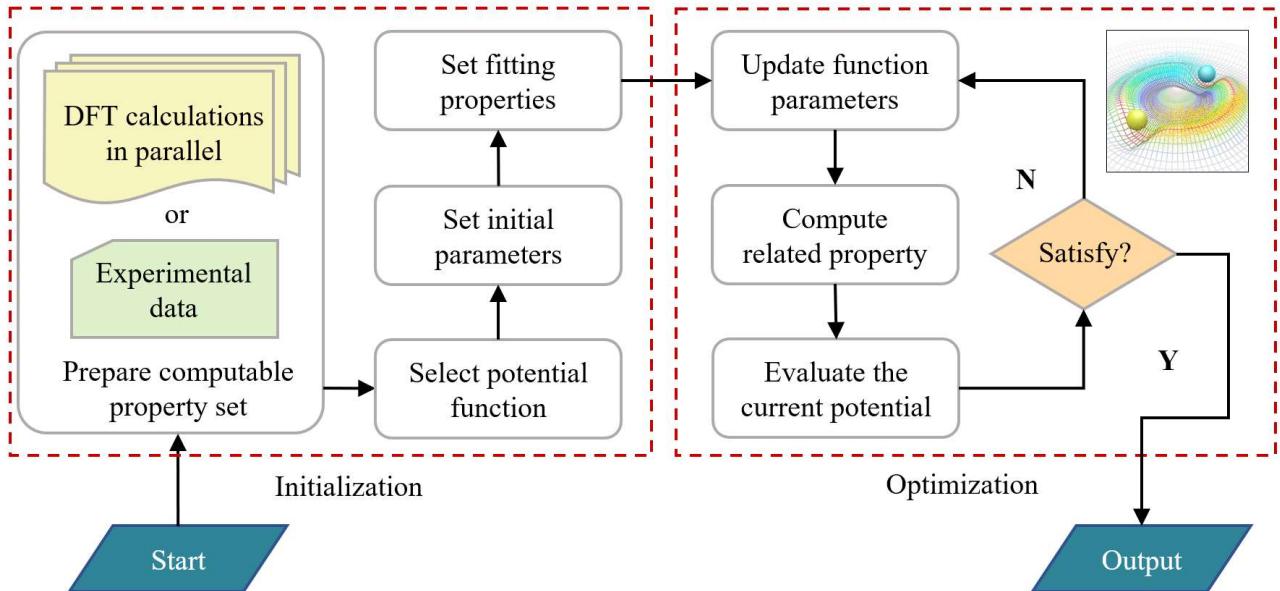


Fig. 5.1 Workflow of the core fitting process in EAPOT Studio

Generally, the potential fitting can be divided into three steps: initialization, optimization, validation, and output, where the related workflow is illustrated in Fig. 5.1. At the initialization step, the potential parameters and training targets should be provided or prepared, albeit some default parameters are already provided by EAPOT. During each optimization step, all fitting parameters of the potential functions vary according to the minimization strategy and then: (i) update the potential parameters at each loop step; (ii) compute the related target properties, and (iii) evaluate the predefined cost function. In the validation and output step, the optimized potential parameters are validated by reproducing several predefined properties, and then the final potential is saved in a tabulated potential format adapted to atomistic simulation codes, such as LAMMPS [1]. In EAPOT, the fitting workflow is divided into five steps: (1)

Choose potential functions, (2) Prepare fitting targets, (3) Set minimization algorithms, (4) Add other constraints, and (5) Run optimizations.

The five steps can be switched by the fit menu or left buttons split by arrows, as shown in Fig. 5.2 marked by orange color. In the following sections, we shall provide a full description of the five steps.

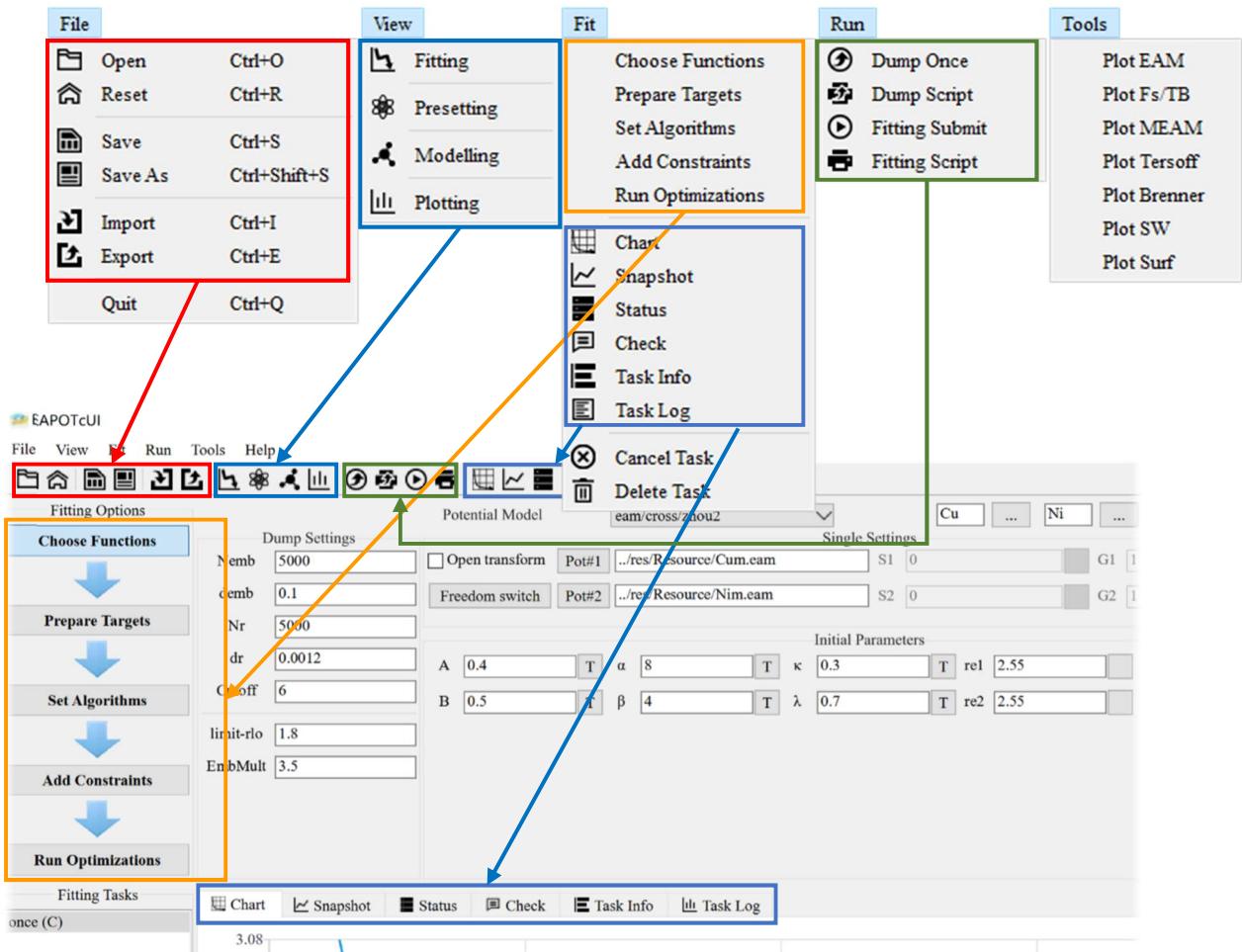


Fig. 5.2 Fitting workflow and main menu options

5.1. Choose potential functions

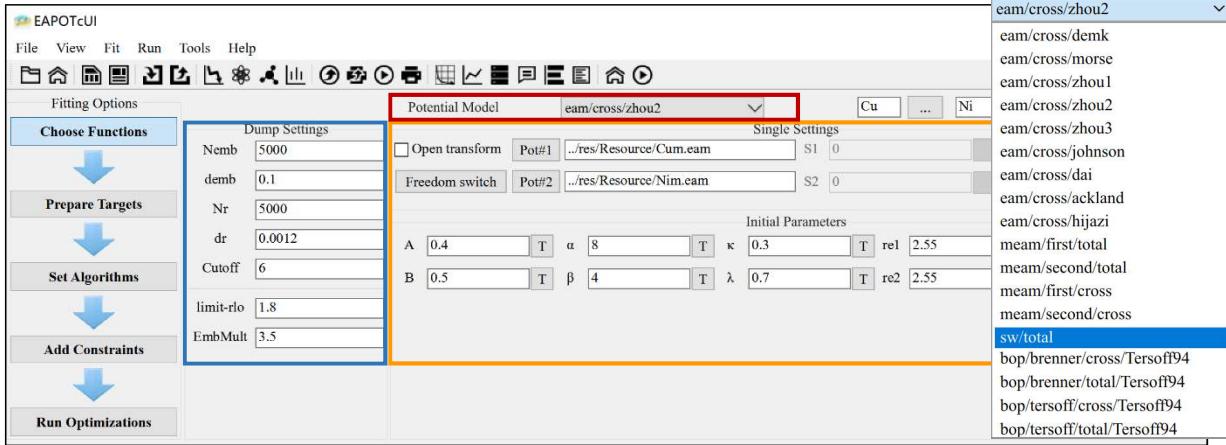


Fig. 5.3 Screenshot of potential function settings

In EAPOT, the following potential models are implemented: i) the embedded-atom method (EAM) [2-4], tight-binding (TB) [5-7], and Finnis-Sinclair (FS) [8] potentials; ii) the Stillinger-Weber (SW) potential [9] and the analytic bond-order potential (BOP) in the Tersoff [10, 11] and Brenner formats [12]; and iii) the modified EAM (MEAM) and second-nearest neighbor MEAM potentials [13-15].

As shown in Fig. 5.3, one can choose the potential function form in the “Function Type” Combobox (marked by red color), set the potential parameters in the “Initial Parameters” frame (marked by orange color), and other properties in this step.

5.2. Prepare fitting targets

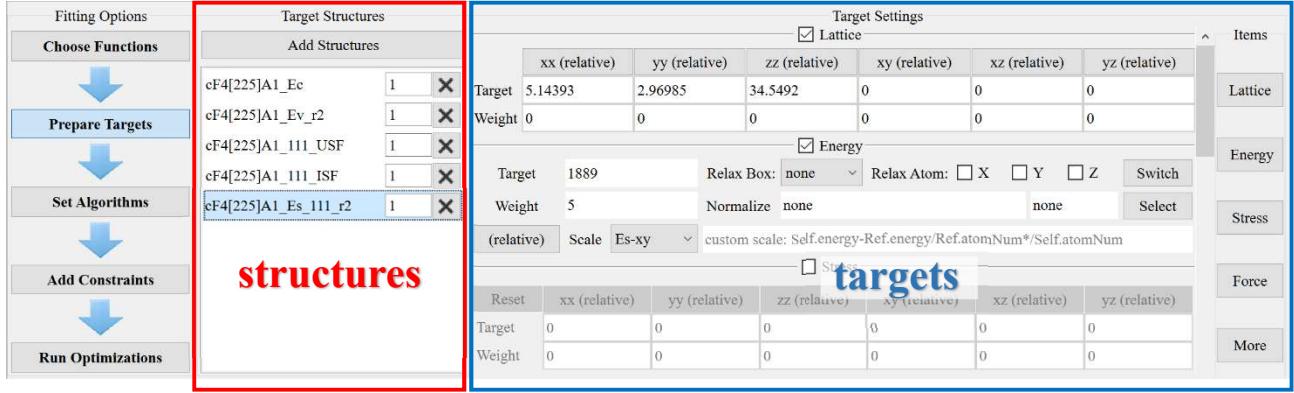


Fig. 5.4 Screenshot of fitting targets settings

The optimization quality depends on the minimization process for a cost function that quantifies the difference between the predicted and trained sets. The cost function used in EAPOT is the well-known least-squares method by minimizing the sum of squares of the deviations for evaluation, which is expressed in the general form:

$$Z_{\text{tot}} = \sum_i^N u_i Z_i, \quad (3.1)$$

where N is the number of structures, u_i is the weight factor for the i th structure, and Z_i represents the corresponding difference between the predicted and trained targets, given by:

$$Z_i = \sum_{k=0}^N w_k \delta(A_k^{\text{predict}}, A_k^{\text{target}})^2 + \frac{1}{3n} \sum_{j=0}^n \sum_{\alpha=0}^3 w_{i,\alpha} \delta(F_{j,\alpha}^{\text{predict}}, F_{j,\alpha}^{\text{target}})^2, \quad (3.2)$$

In which the deviation function is,

$$\delta(x^{\text{predict}}, x^{\text{target}}) = \begin{cases} \frac{x^{\text{predict}} - x^{\text{target}}}{x^{\text{target}}} & , x^{\text{target}} \neq 0 \\ x^{\text{predict}} - x^{\text{target}} & , x^{\text{target}} = 0 \end{cases}. \quad (3.3)$$

Here, A_k^{predict} and A_k^{target} are the k th global prediction and target, respectively,

and $F_{j,\alpha}^{\text{predict}}$ and $F_{j,\alpha}^{\text{target}}$ values are the α component of the force prediction and target for the atom j , respectively. Moreover, w_k and $w_{i,\alpha}$ are the corresponding weight factors, which are necessary since several specific targets are usually more critical than others during the fitting procedure. It is helpful to measure the relative deviations rather than the absolute deviations from the reference data, except for the zero values and atomic forces, where the absolute deviations are used by default due to the large magnitude difference in atomic force values. (The deviation function of forces can be set manually, as shown in Fig. 5.5 marked by red color).

To meet the demands of different simulation scenarios, EAPOT realizes multiple combinations of "energy-stress-force-elasticity" and multi-level objective optimization schemes, where both experimental and ab initio data can be introduced as training targets in the optimization. The inner relaxation is supported in EAPOT for a given structure, enabling the training to the experimental equilibrium values, e.g., the lattice constant, cohesive energy, elastic constants, stacking fault energies (SFEs), surface energies, or other equilibrium properties. The inner relaxation can be divided into two types: box relaxation and atomic relaxation, as shown in Fig. 5.5.

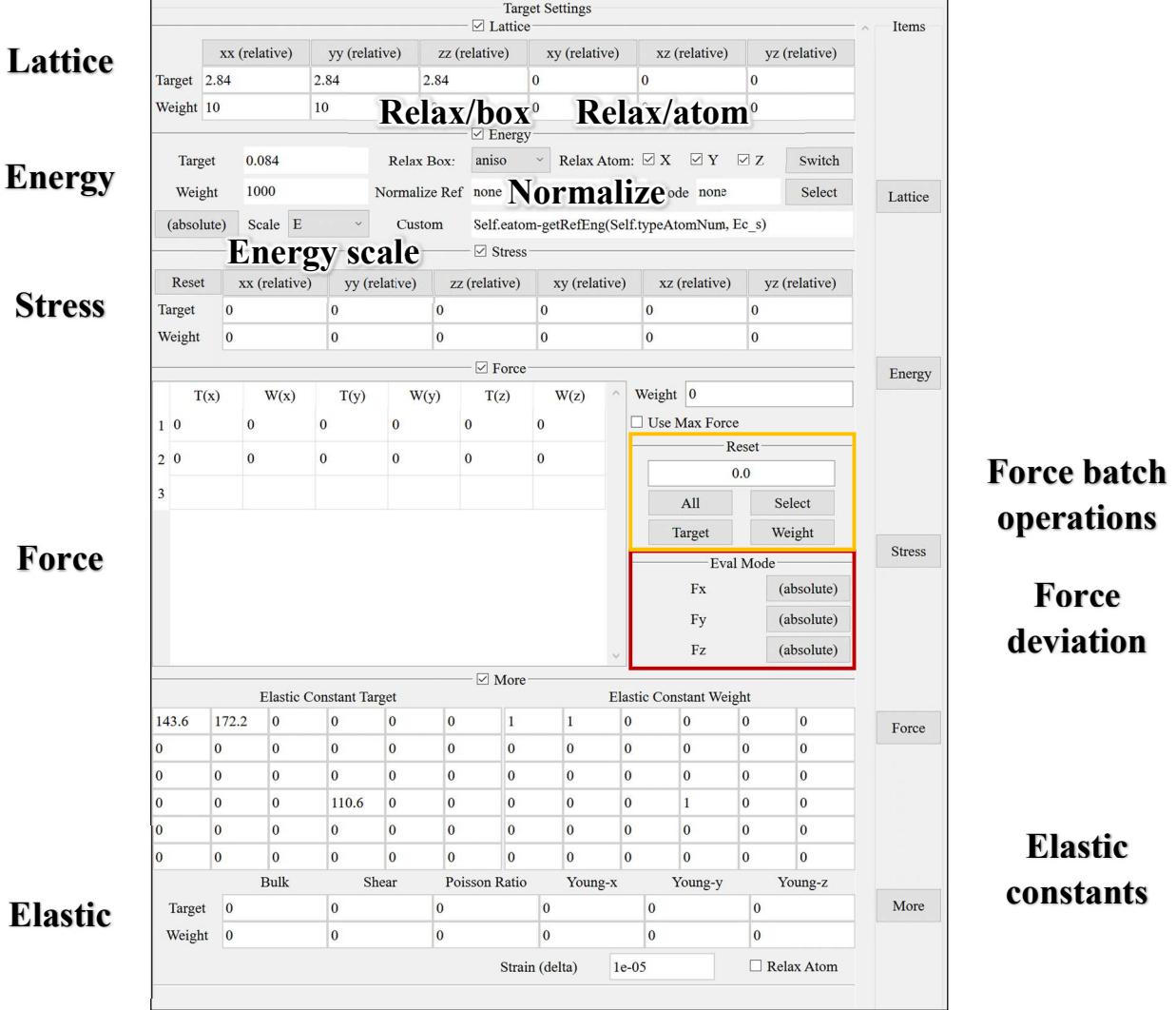


Fig. 5.5 Complete training target setting window

Relax Box: The keyword *none* (as shown in Fig. 5.6) means no box relaxation in the inner loop. The keyword *iso* means couple all three diagonal components together when pressure is computed (hydrostatic pressure). The keyword *aniso* means x, y, and z dimensions are controlled independently using the Pxx, Pyy, and Pzz components of the stress tensor as the driving forces. The keyword *full* means x, y, z, xy, xz, and yz dimensions are controlled independently using their individual stress components as the driving forces.

Relax Atom: Check to activate atomic relaxation in the corresponding axis.

Reference: The choice of this parameter has two effects: (1) as a reference structure to get relaxation ratio (as shown in Fig. 5.6), (2) as the reference energy to calculate the energy difference if “energy scale” is selected.

Normalize: Normalize the lattice constants according to the relaxation ratio of the reference structure.

Energy scale: A series of predefined expressions, as shown in Table 5.1.

For more details on energy settings, please refer to Fig. 5.6.

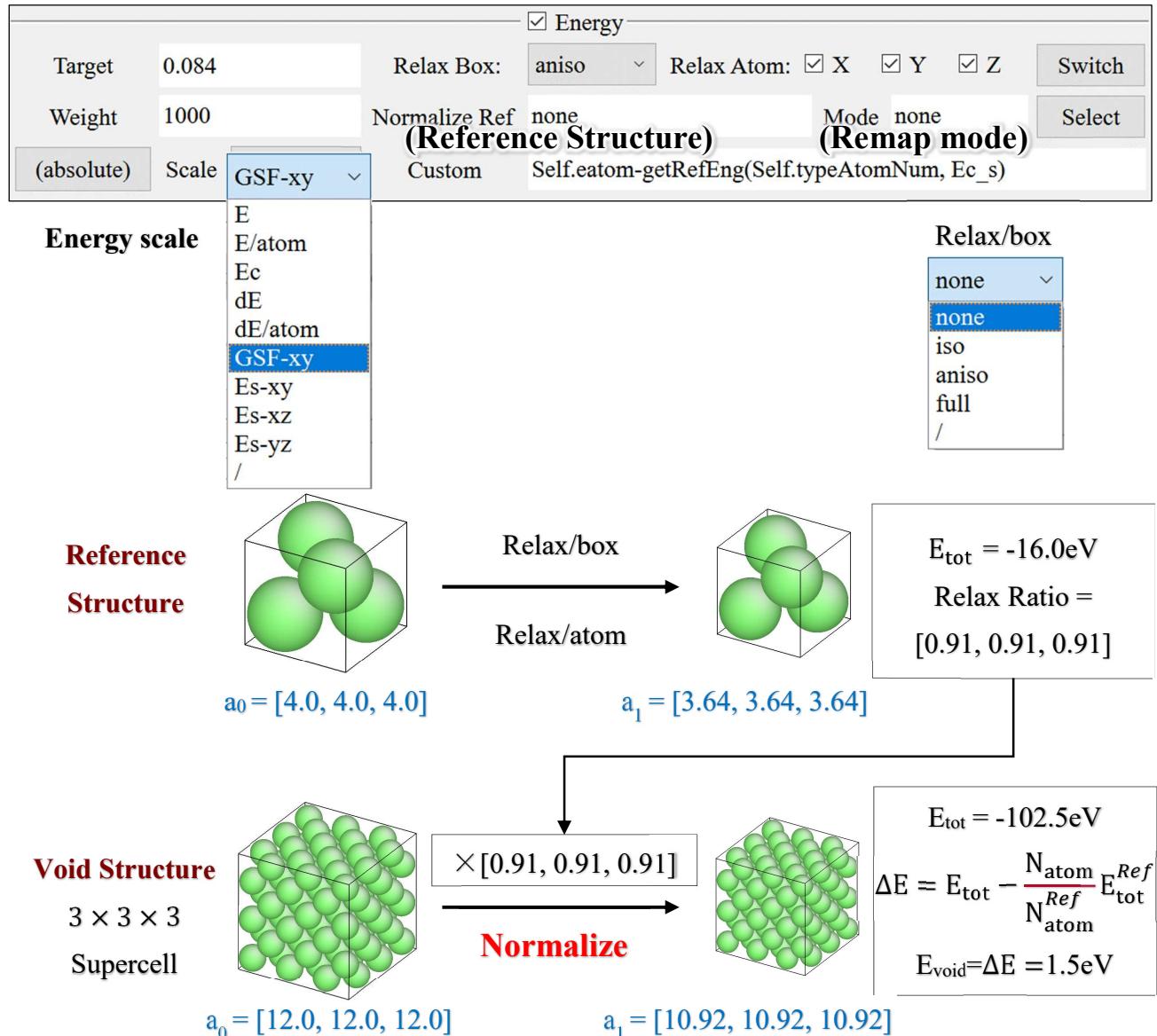


Fig. 5.6 Energy calculation workflow

Table 5.1 Energy scale expressions

	Unit	expression
E	eV	E_{tot}
E/atom	eV	$E_{\text{tot}}/N_{\text{atom}}$
dE	eV	$E_{\text{tot}} - N_{\text{atom}} N_{\text{atom}}^{\text{Ref}} / N_{\text{atom}}^{\text{Ref}}$
dE/atom	eV	$E_{\text{tot}}/N_{\text{atom}} - N_{\text{atom}}/N_{\text{atom}}^{\text{Ref}}$
E_c	eV	$-E_{\text{tot}}/N_{\text{atom}}$
GSF-xy	mJ/m^2	$\frac{16020}{S_{xy}} (E_{\text{tot}} - N_{\text{atom}} E_{\text{tot}}^{\text{Ref}} / N_{\text{atom}}^{\text{Ref}})$
$E_s\text{-xy}$	mJ/m^2	$\frac{8010}{S_{xy}} (E_{\text{tot}} - N_{\text{atom}} E_{\text{tot}}^{\text{Ref}} / N_{\text{atom}}^{\text{Ref}})$
$E_s\text{-xz}$	mJ/m^2	$\frac{8010}{S_{xz}} (E_{\text{tot}} - N_{\text{atom}} E_{\text{tot}}^{\text{Ref}} / N_{\text{atom}}^{\text{Ref}})$
$E_s\text{-yz}$	mJ/m^2	$\frac{8010}{S_{yz}} (E_{\text{tot}} - N_{\text{atom}} E_{\text{tot}}^{\text{Ref}} / N_{\text{atom}}^{\text{Ref}})$

Note: S represents the area size, and 16020 is the unit conversion factor.

5.3. Set minimization algorithms

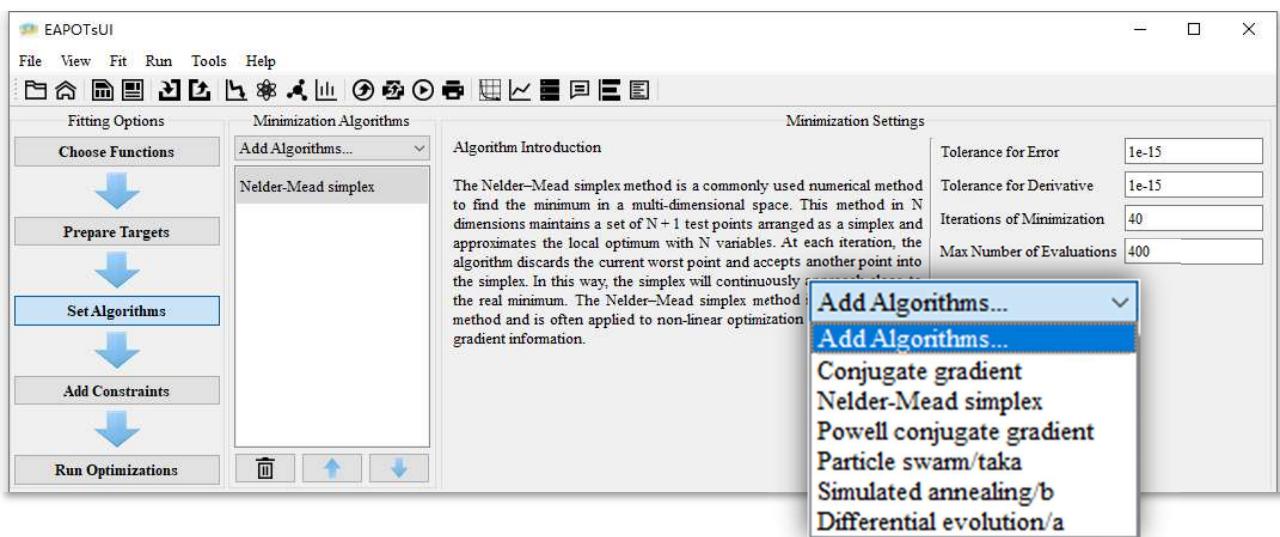


Fig. 5.7 Screenshot of minimization algorithms settings

In general, the minimization algorithms can be classified into local and global algorithms. While the local algorithms always find the closest local minimum, the global

algorithms are non-deterministic methods that sample a larger part of the parameter space and find the best minimum but have a computational burden that is several orders of magnitude larger.

As evaluating the non-linear cost function is relatively expensive, several efficient local minimization algorithms were implemented in EAPOT. The Nelder–Mead simplex method [16] is a commonly used numerical method to find the minimum in a multi-dimensional space. This method in N dimensions maintains a set of $N + 1$ test points arranged as a simplex and approximates the local optimum with N variables. At each iteration, the algorithm discards the current worst point and accepts another point into the simplex. In this way, the simplex will continuously approach close to the real minimum. The Nelder–Mead simplex method is a direct search method and is often applied to non-linear optimization problems without gradient information.

The conjugate gradient (CG) algorithm was implemented in the Polak-Ribiere version [17], where the next search direction must be conjugate to the previous search directions. This algorithm begins the optimization in the negative gradient direction, and at each iteration, the gradient is combined with the previous iteration information to create a new search direction. This method has the feature that it only requires the storage of two gradient vectors, which for large problems with many parameters, is a significant saving in storage versus Newton-type algorithms.

Powell algorithm [18] is a conjugate-gradient-like optimization method that is efficient in the number of calculations and does not require any gradient information. The Powell method first determines all search vectors at the starting point and then

transforms the multi-dimensional optimization into a line search problem. At each iteration, a new parameter guess is determined by a sequence of line searches. Afterward, the search direction which gives the largest impact is replaced by a new search direction, and the iterations are performed until convergence.

The particle swarm optimization (PSO) [19] is an efficient global searching method, where the optimization is carried out with typically dozens of particles, each treated as a parameter set that is initialized randomly. The basic idea of PSO is to evolve each particle towards the optimum using two contributions: (1) their best-fitted parameters in the past and (2) the best-fitted particle in the swarm. Each particle keeps track of its coordinates in the problem space associated with the best solution it has achieved so far. When a particle takes all the population as its neighbors, the global solution is achieved.

Differential evolution [20] algorithms perform optimization by mimicking natural selection with a population of candidate solutions to the problem. The population is then evolved from one generation to the next, using techniques inspired by biological evolution such as mutation or natural selection. Typically weaker members of a population are replaced by fitter offspring created by the combination of traits from other population members, thus increasing the population's overall quality.

The simulated annealing algorithm was also implemented [21, 22]. Here, jumps are made in potential-parameter space, with downhill moves (those that result in a decrease in Z_{cost}) always accepted and uphill moves accepted only with a probability $\exp^{-\frac{\Delta Z}{T}}$. Here, ΔZ is the increase in Z due to the proposed move, and T is a fictitious temperature providing an overall control on the probability of an uphill. By allowing

uphill moves in this way, a more exhaustive search of the potential parameter space is enabled, which leads to a real global minimum.

5.4. Add other constraints

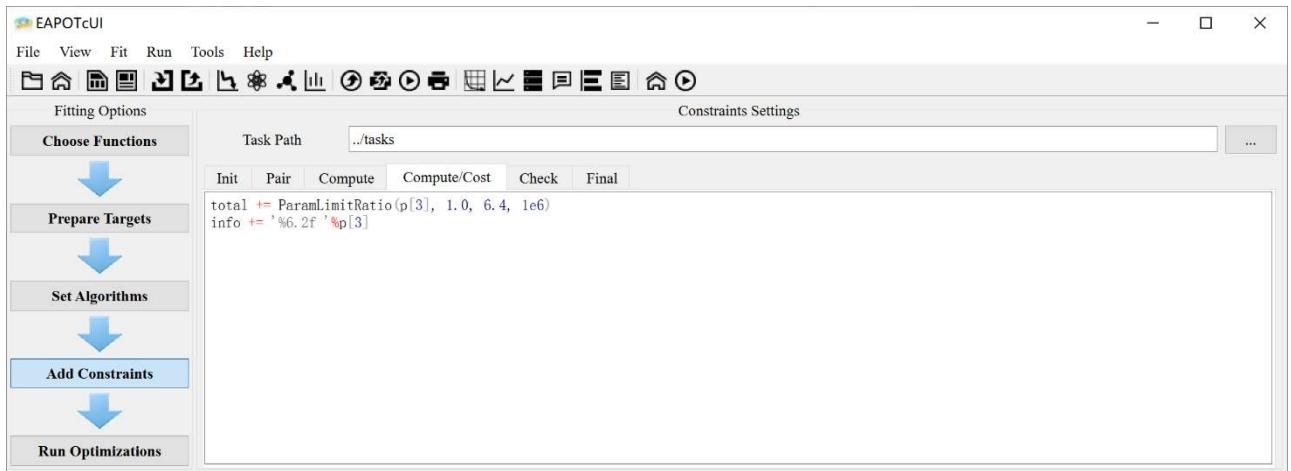


Fig. 5.8 Screenshot of other constraints settings

This window can be used to specify other program settings or enter additional custom codes. The optimization quality depends on the cost function that quantifies the difference between the predicted and trained sets, thus EAPOT will always generate a python script according to the settings in the fitting target window to get the cost function value. For example, settings in Fig. 5.4 will be converted to the following Python script. In addition, this window allows additional custom codes to modify certain EAPOT properties during the fitting procedure, for example, custom codes shown in Fig. 5.8 will be inserted into the cost function script (see those codes separated by comments):

```
sq2 = lambda w, t, v: w*(v-t)**2
fmt = lambda u: '%9.3g' %u if abs(u) > 1e5 else '%5.4g' %u
def ParamLimitRatio(x, a, b, c): return c*(x<a)*(x/a-1)**2 + c*(b<x)*(x/b-1)**2
def ParamlimitAbslo(x, a, b, c): return c*(x<a)*(x - a)**2 + c*(b<x)*(x - b)**2
weight = [ 1, 1, 1, 5, 3, 3, 5, 10, 10, 5 ]
target = [ 3.615, 3.615, 3.615, 3.54, 170, 122.5, 75.8, 1.27, 161, 45, 1889 ]
factor = [ 3.615, 3.615, 3.615, 3.54, 170, 122.5, 75.8, 1.27, 161, 45, 1889 ]
```

```

weight = tuple(map(lambda w, f: w/f**2, weight, factor))
targetMsg = ''.join(map(fmt, target))

def computeCostScript(caller, pArg):
    arg, v, p, f, c, res = EAPOT.parseComputeExternal(pArg)
    val, total, info = [0.0]*11, 0.0, ''

    Ec, Ela, Ev, USF, ISF, Es = v
    val[0:3], val[3], val[4:7] = Ec.lattice, Ec.Ec, Ela.cubic
    val[7] = Ev.energy-cF4225A1_Ec.energy*7.75
    val[8] = 16020/(USF.xx*USF.yy)*(USF.energy-Ec.energy*6)
    val[9] = 16020/(ISF.xx*ISF.yy)*(ISF.energy-Ec.energy*6)
    val[10] = 8010 /(Es.xx * Es.yy)*(Es.energy -Ec.energy*3)

    total += ParamlimitAbslo(p[3], 1.0, 6.4, 1e6)
    info += '%6.2f %p[3]

    res[0] = sum(map(sq2, weight, target, val)) + total
    return EAPOT.packMsg(''.join(map(fmt, val)) + info)

```

The cost function script will evaluate the difference between the predicted and trained sets and save it into the *res* variable. The return value of the cost function script is a short string message to give a summary of current fitting information. In this example, the custom script will limit the third fitting parameter, where “[ParamlimitAbslo\(p\[3\], 1.0, 6.4, 1e6\)](#)” will limit *p[3]* to below 6.4 to prevent it from exceeding the cut-off radius.

5.5. Run optimizations

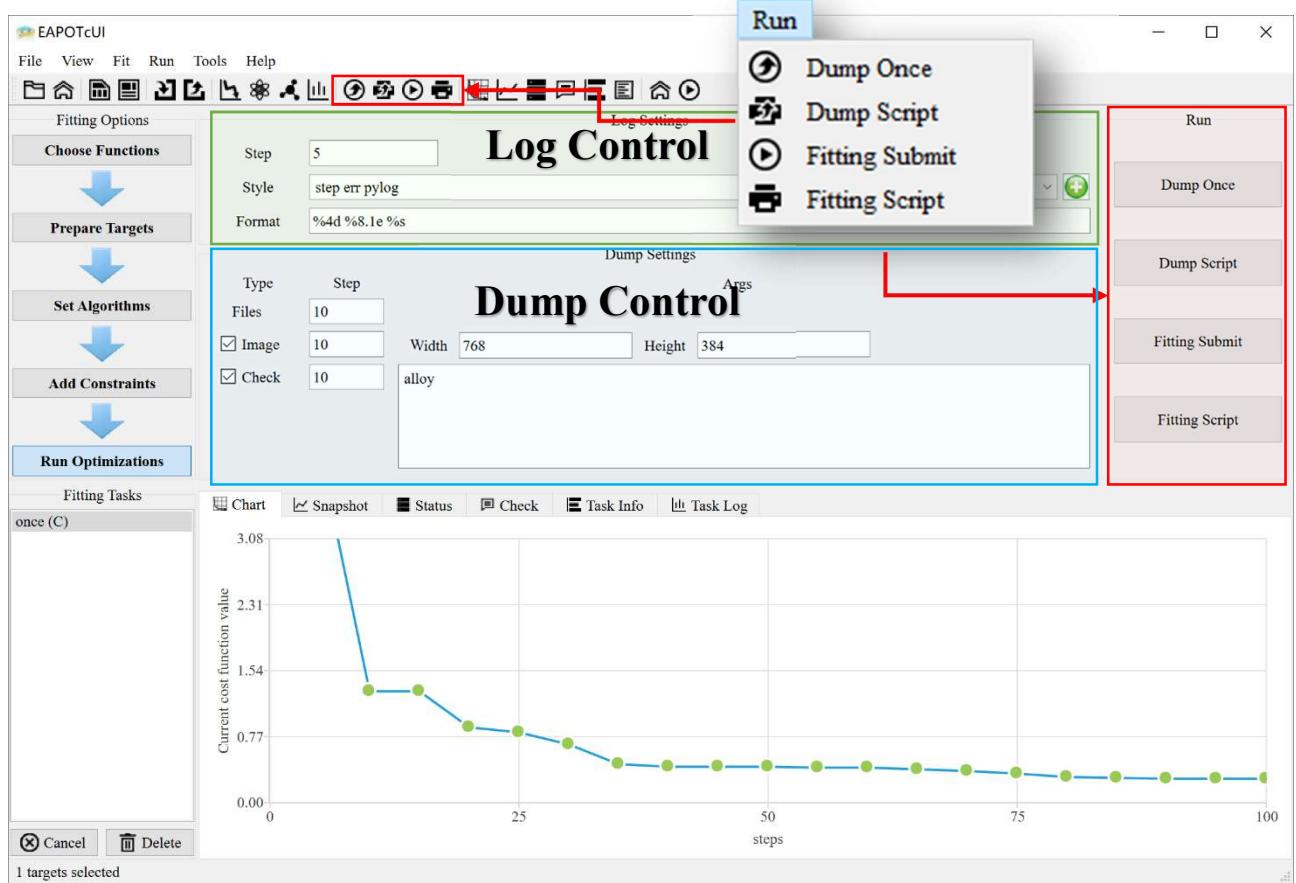


Fig. 5.9 Screenshot of other constraints settings

The log system print info (e.g., cost function value, timestep) on timesteps that are a multiple of N and at the beginning and end of a simulation. A step value of 0 will only print the log at the beginning and end. The content and format of what is printed are controlled by the “style Combobox” and “format Combobox”, as shown in Table 5.2.

Table 5.2 Log keywords

keyword	description
step	timestep
eval	cost function evaluation times
err	the current cost function value
pylog	summary of current fitting information from the cost function

The dumping system outputs a potential file, snapshot, or checking result to one or more files every N timesteps in one of several styles. After all the settings are provided, one can submit a task by clicking the right buttons in Fig. 5.9, and more details are listed in Table 5.3. Normally, the “Dump Once” button can be used to test and will not allocate a new folder, while the “Fitting Submit” button will submit a new task formally.

Table 5.3 Submit button content

button	content
Dump Once	Submit the task in a fixed folder: /tasks/once
Dump Script	Export the project file in a fixed folder: /tasks/once
Fitting Submit	Submit the task in a new folder
Fitting Script	Export the project file in a new folder

5.6. Potential fitting task manager

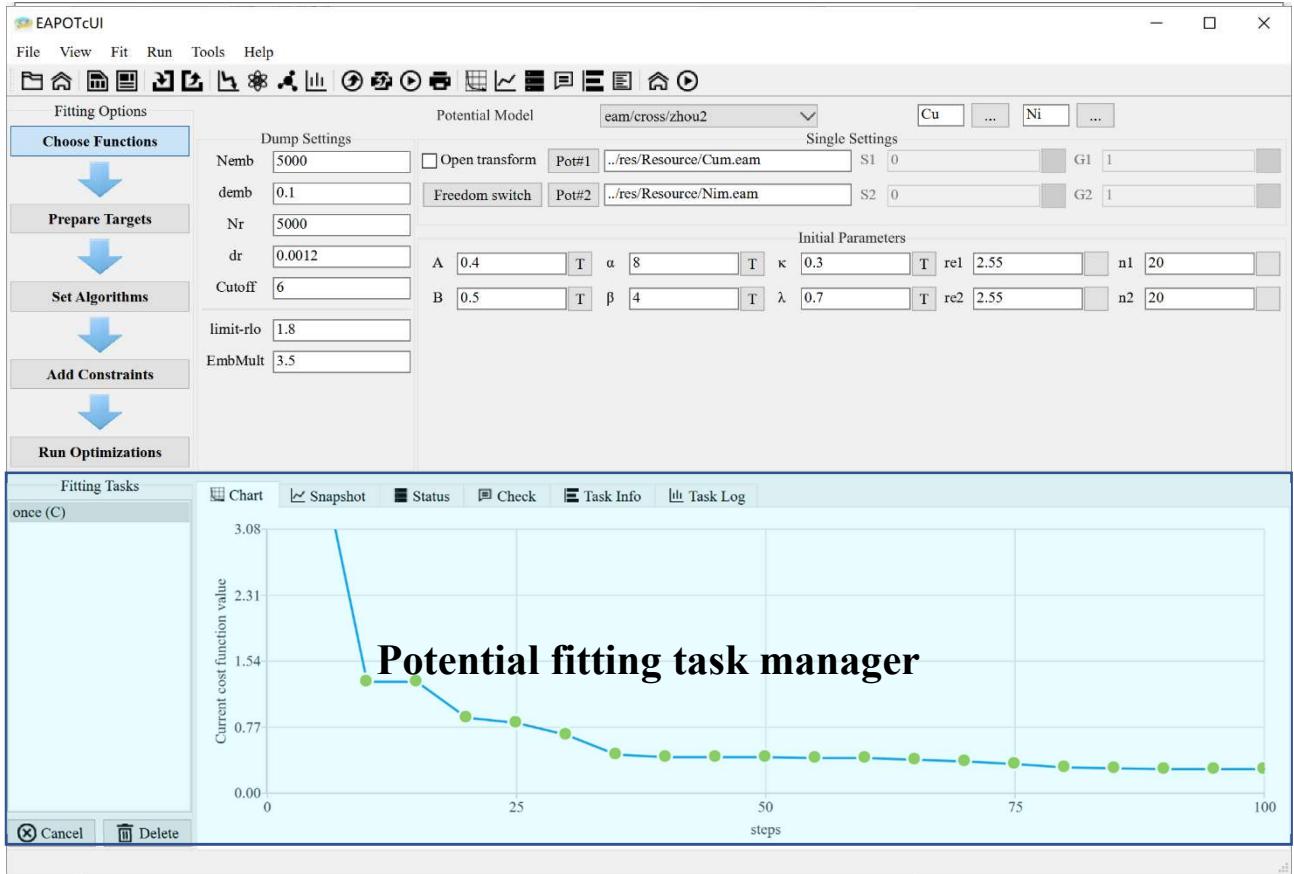


Fig. 5.10 Screenshot of Potential fitting task manager (Cost function)

Once a task is submitted, the potential fitting task manager will provide a dynamic real-time view of running tasks. It can display summary information of the selected task as well as a list of tasks currently being managed by the EAPOT, where one can double-click the list item to open the task folder in the file manager. The task summary information includes the potential snapshot, status, checking results, running information, and exception log, which records all exceptions during the running time. The fitting task manager provides an interactive interface for process manipulation, as more details are shown in Fig. 5.11.

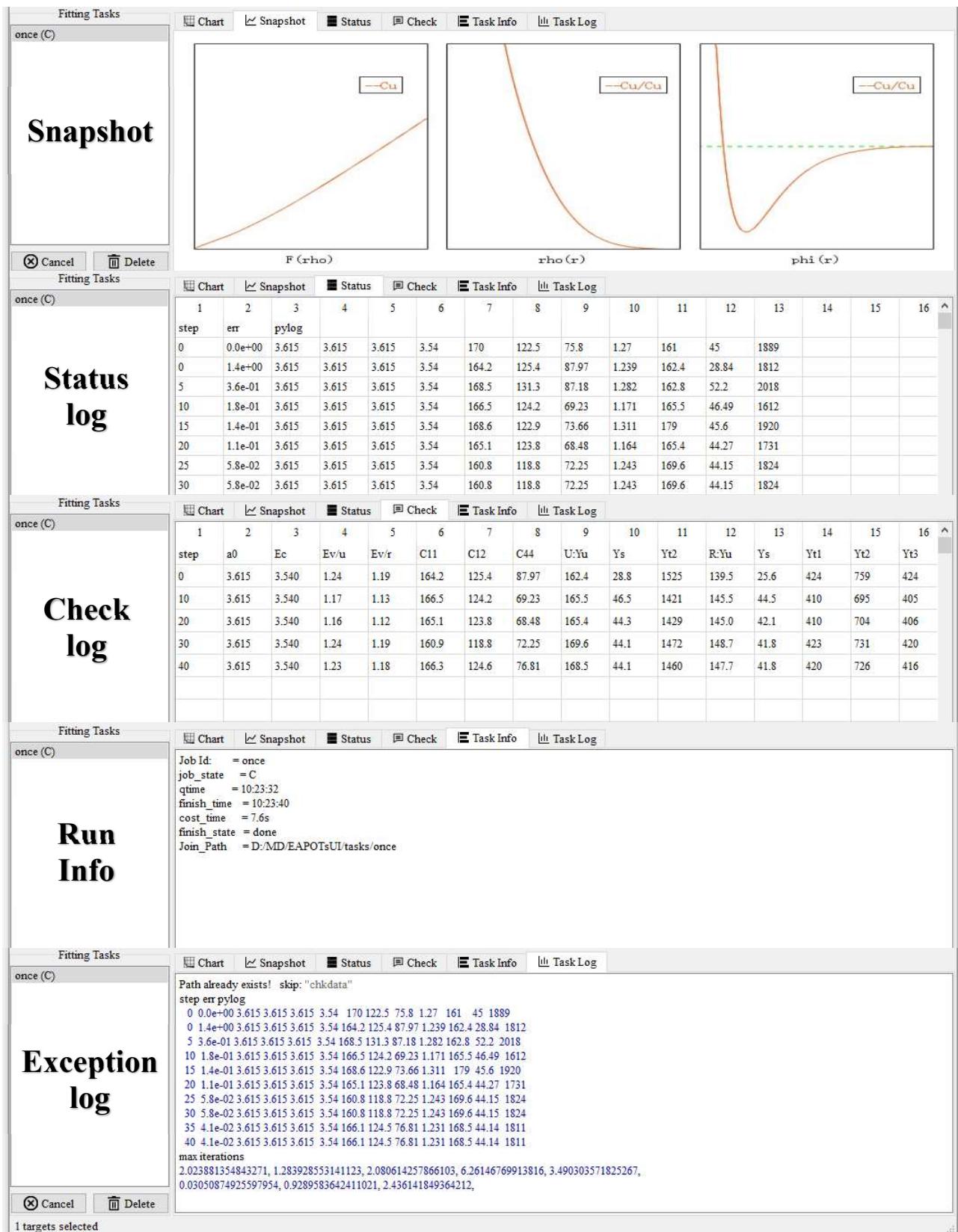


Fig. 5.11 Screenshot of Potential fitting task manager (All views)

Reference

- [1] S. Plimpton, Journal of Computational Physics, 117 (1995) 1-19.
- [2] M.S. Daw, M.I. Baskes, Physical Review B, 29 (1984) 6443-6453.
- [3] S.M. Foiles, Physical Review B, 32 (1985) 7685-7693.
- [4] S.M. Foiles, M.I. Baskes, M.S. Daw, Physical Review B, 33 (1986) 7983-7991.
- [5] F. Cleri, V. Rosato, Physical Review B, 48 (1993) 22-33.
- [6] F. Ducastelle, F. Cyrot-Lackmann, Journal of Physics and Chemistry of Solids, 31 (1970) 1295-1306.
- [7] V. Rosato, M. Guillope, B. Legrand, Philosophical Magazine A, 59 (1989) 321-336.
- [8] M.W. Finnis, J.E. Sinclair, Philosophical Magazine A, 50 (1984) 45-55.
- [9] F.H. Stillinger, T.A. Weber, Physical Review B, 31 (1985) 5262-5271.
- [10] J. Tersoff, Physical Review B, 37 (1988) 6991-7000.
- [11] J. Tersoff, Physical Review B, 38 (1988) 9902-9905.
- [12] D.W. Brenner, Physical Review B, 42 (1990) 9458-9471.
- [13] M.I. Baskes, Physical Review B, 46 (1992) 2727-2742.
- [14] B.-J. Lee, M.I. Baskes, Physical Review B, 62 (2000) 8564-8567.
- [15] B.-J. Lee, M.I. Baskes, H. Kim, Y. Koo Cho, Physical Review B, 64 (2001) 184102.
- [16] Jeffrey, C, Lagarias, James, A, Reeds, Margaret, H, Wright, Paul, SIAM Journal on Optimization, (1998).
- [17] E. Polak, G. Ribiere, Revue Francaise Information Recherche Operationnelle, 3 (1969) 35-43.
- [18] M.J.D. Powell, The Computer Journal, 7 (1965) 303-307.

- [19] Z. Zhan, J. Zhang, Y. Li, H.S. Chung, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39 (2009) 1362-1381.
- [20] R. Storn, K. Price, Journal of Global Optimization, 11 (1997) 341-359.
- [21] A. Corana, M. Marchesi, C. Martini, S. Ridella, 13 (1987) 262–280.
- [22] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by Simulated Annealing, in: M.A. Fischler, O. Firschein (Eds.) Readings in Computer Vision, Morgan Kaufmann, San Francisco (CA), 1987, pp. 606-615.

6. Potential presetting

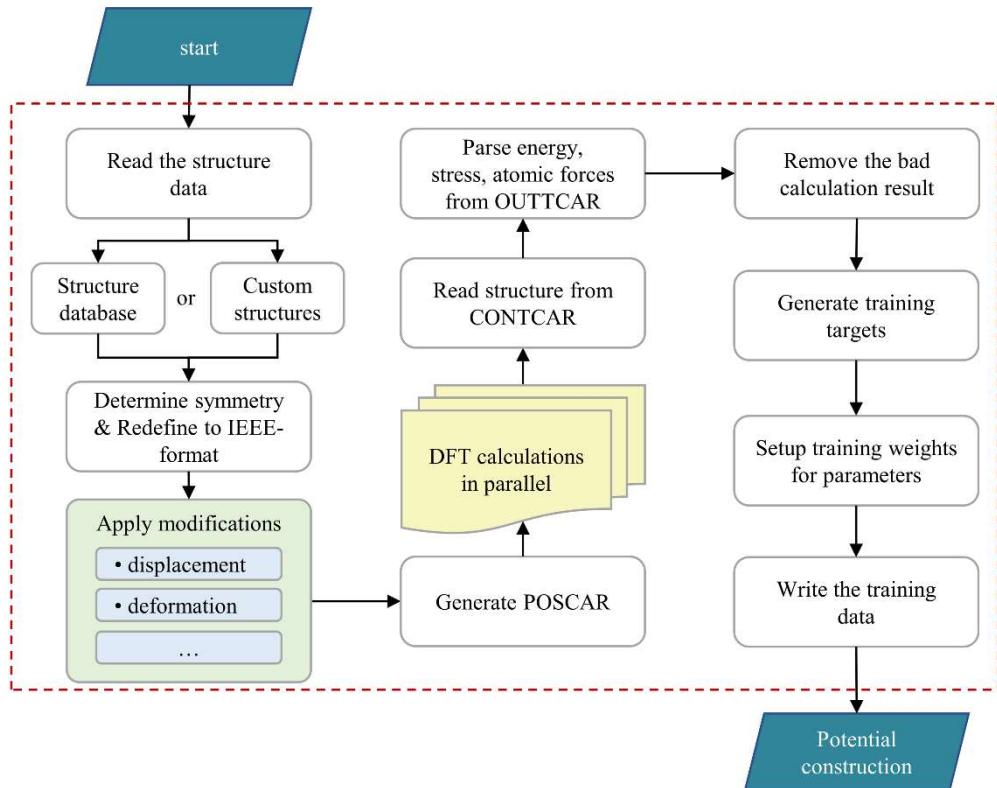


Fig. 6.1 The high-throughput workflow used in EAPOT Studio.

To address the deficiency of experimental data [1], Ercolelli and Adams [2] proposed a force-matching method via first-principles calculations, which provided a feasible solution to expand the search for a high-dimensional, potential energy space. Motivated by this advancement, EAPOT realizes the HT strategy to generate effective interaction potentials by matching microscopic quantities from first-principles methods, such as energies, stresses, elasticity, and forces. Fig. 6.1 shows the HT flowchart implemented in EAPOT through the automatic processing of model construction and data delivery, and a brief introduction of the two steps will be shown in the following.

6.1. Create structures for DFT calculations

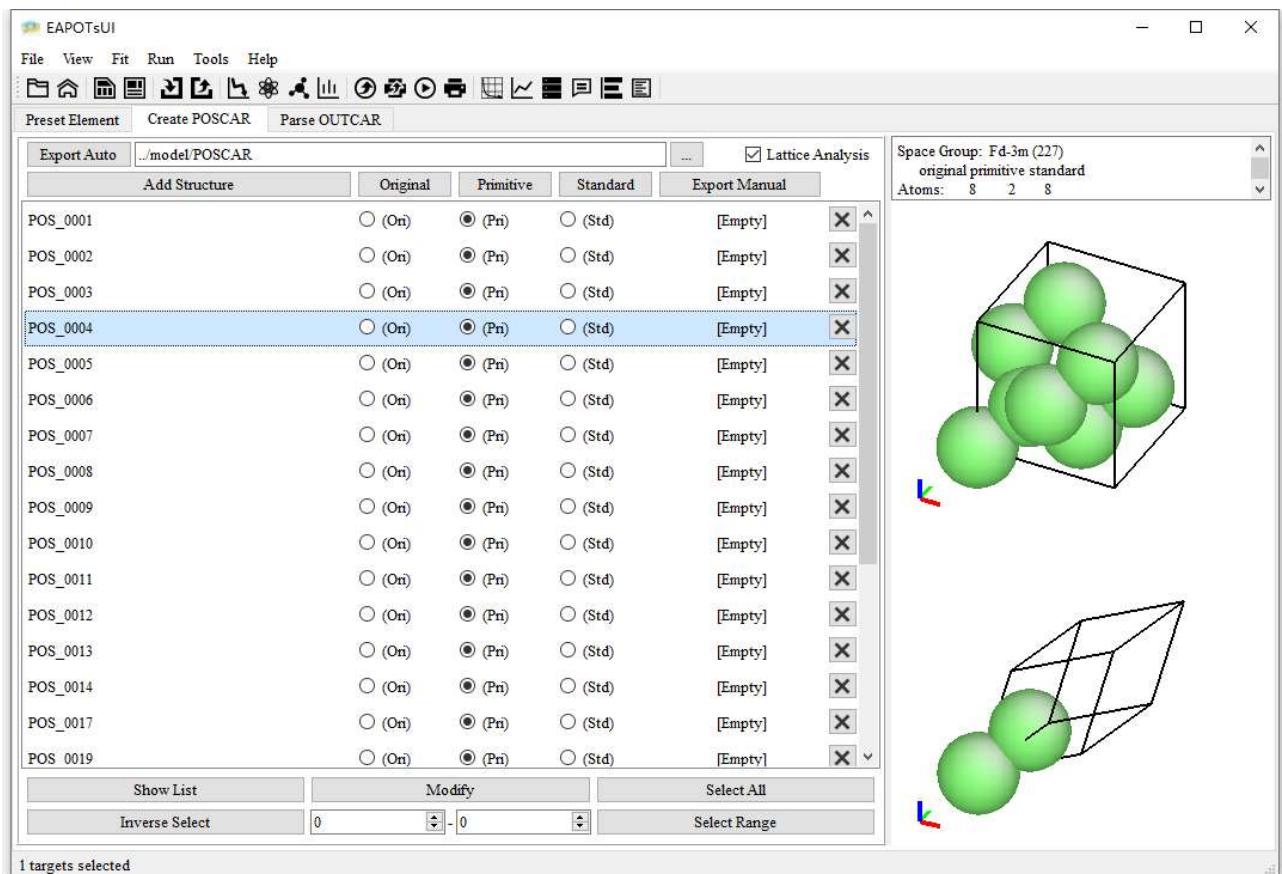


Fig. 6.2 The model creation and lattice symmetry analysis

First, EAPOT read structure data containing lattice vectors and atomic coordinates, and then the lattice symmetry is analyzed using the SPGLIB code [3], as shown in Fig. 6.2. Based on the symmetry, various lattice and atomic modifications are performed on the reduced primary or unit cells to cover various possible configurations that may be relevant to phase transformation, crystal plasticity, or diffusions, as shown in Fig. 6.3.

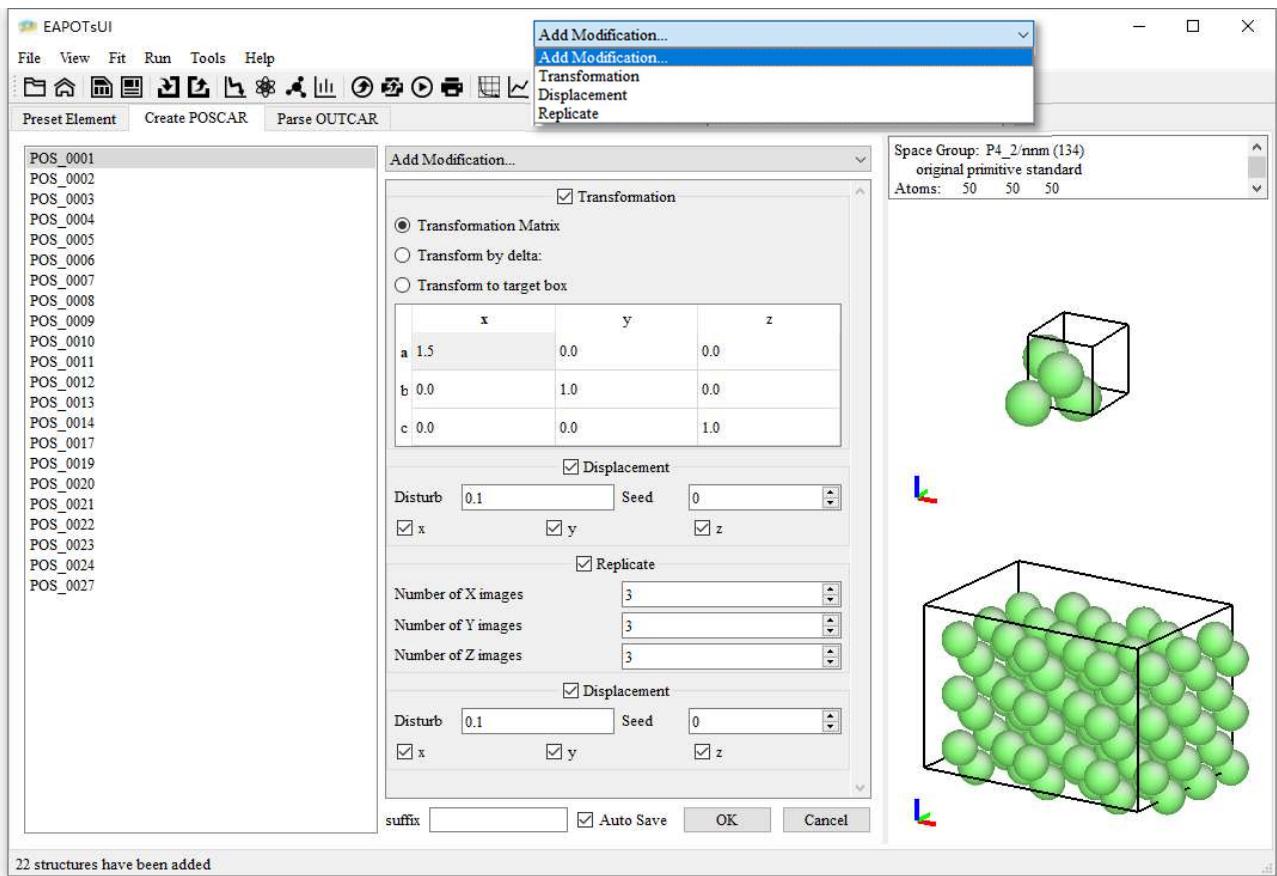


Fig. 6.3 Batch modifications of lattice cell and atomic coordination

Transformation modifier: This modifier applies an affine transformation to the system, which may be used to translate, scale, rotate, or shear the simulation cell. The transformation can be explicitly specified in terms of a 3×3 matrix or directly by prescribing a target geometry for the simulation cell.

1. Transformation Matrix: $\mathbf{Box} = \mathbf{Box} \cdot \mathbf{M}$,
2. Transform by delta: $\mathbf{Box} = \mathbf{Box} + \mathbf{C} \cdot \mathbf{M}$,
3. Transform to target box: $\mathbf{Box} = \mathbf{M}$,

where \mathbf{M} is the input matrix parameter and \mathbf{Box} represents the domain vectors:

$$\mathbf{Box} = \begin{pmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{pmatrix}$$

Displacement modifier: Apply random displacement /perturbation to atoms.

Replicate modifier: This modifier copies all particles multiple times to a system, and the numbers of images on each axis (X/Y/Z) specify how many times the system is copied in each direction.

6.2. Parse properties from DFT calculations

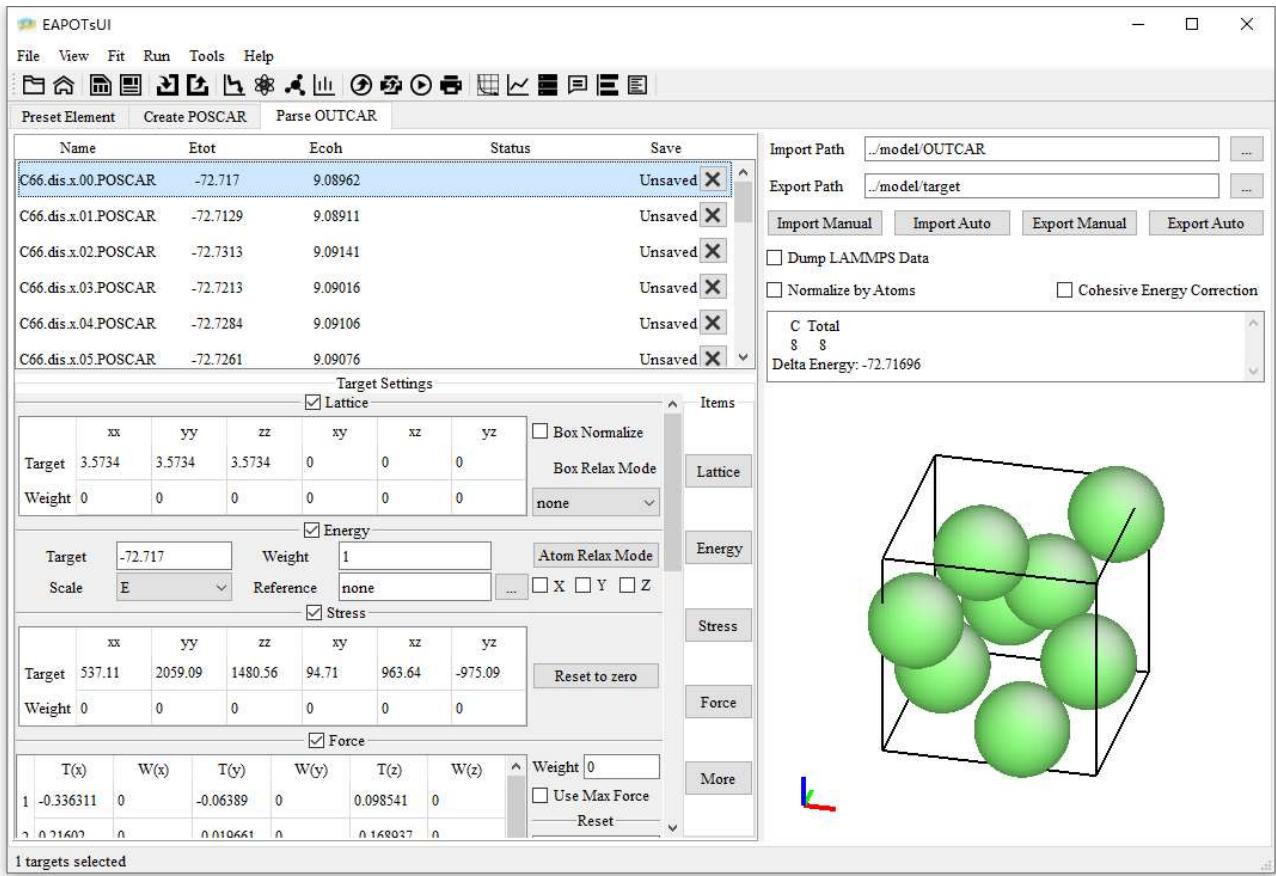


Fig. 6.4 The parsing interface from DFT calculations

Then, all structures are sent for parallel first-principle calculations to obtain their energies, stresses, and forces, and these calculated data are parsed from the generated files for further processing in EAPOT. Finally, the targets are generated according to their different combinations or derivations to other physical properties, such as the SFE,

lattice stability, or elasticity.

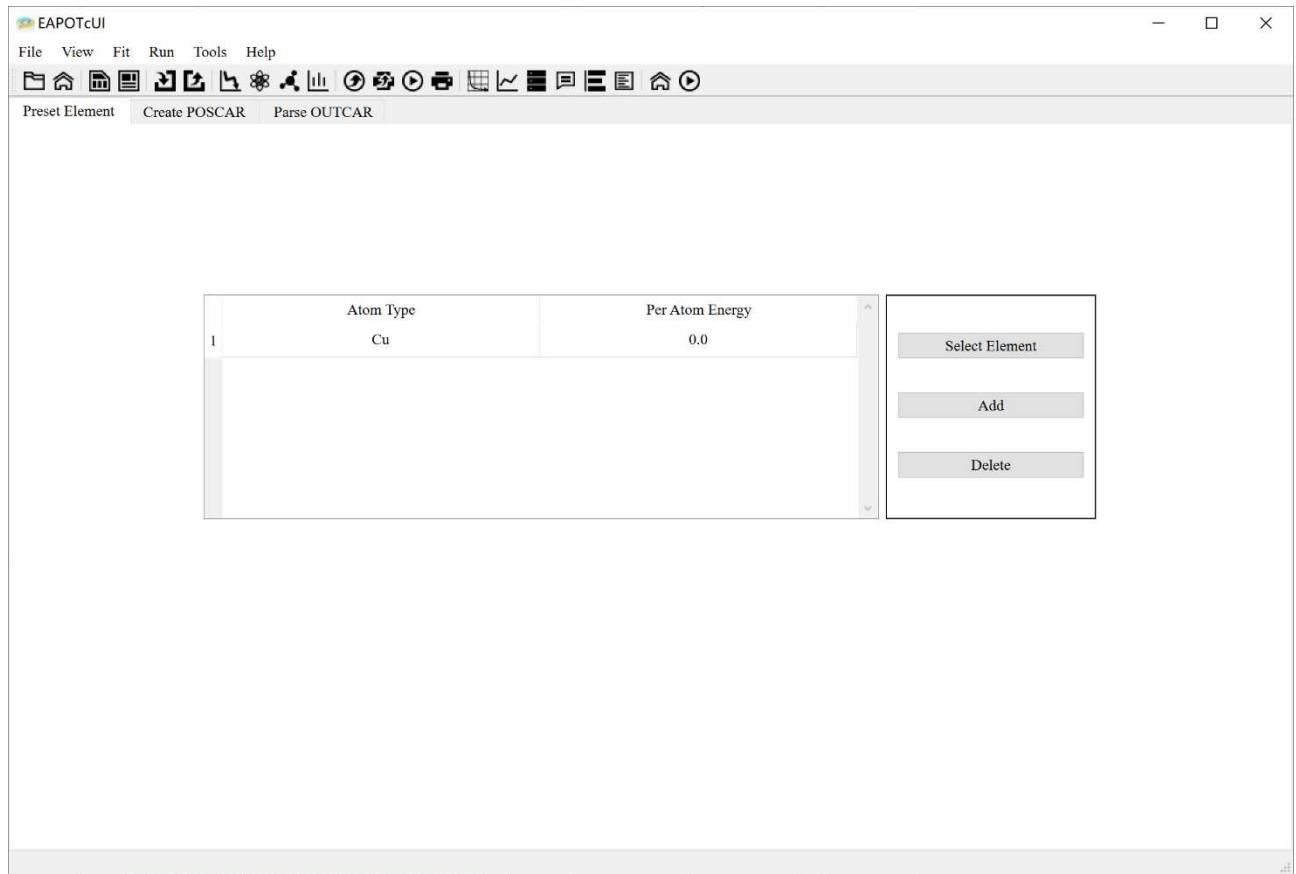


Fig. 6.5 The settings of elemental information

Besides, element type and energy for a single type atom (to calculate the cohesive energy) can be assigned in the first window, as shown in Fig. 6.5.

Reference

- [1] M. Wen, J. Li, P. Brommer, R.S. Elliott, J.P. Sethna, E.B. Tadmor, Modelling and Simulation in Materials Science and Engineering, 25 (2016) 014001.
- [2] F. Ercolessi, J.B. Adams, Europhysics Letters (EPL), 26 (1994) 583-588.
- [3] <http://atztogo.github.io/spglib>, (accessed 1 Feb 2021).

7. Structure modeling

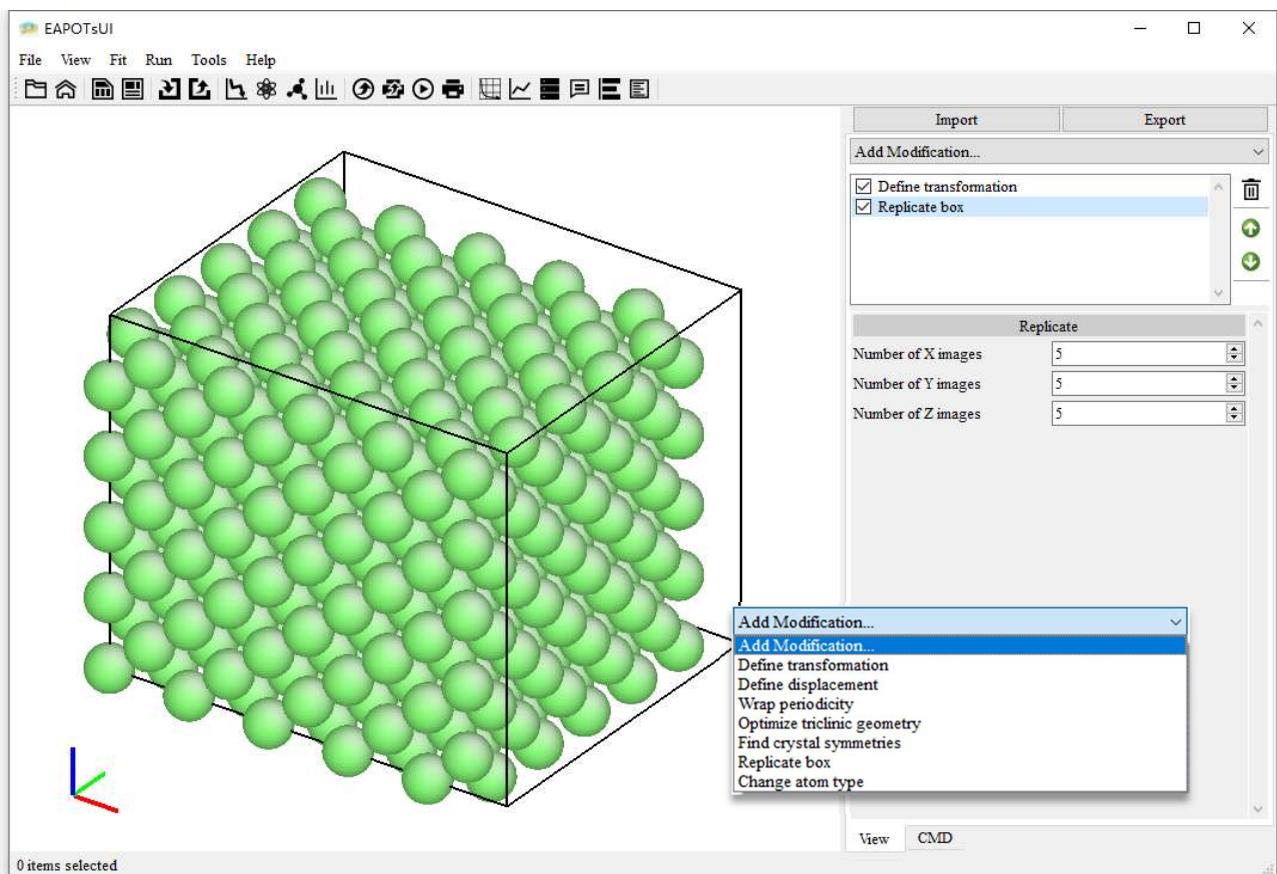


Fig. 7.1 Screenshot of the structure modeling window

Modifiers are the basic building blocks for creating a data pipeline in the structure modeling module, and each modifier implements a very specific, well-defined type of operation or computation, and typically one will need to combine several modifiers to accomplish more complex tasks, and more details are listed in Table 7.1.

Table 7.1 Structure modeling modifiers

Modifier name	Description
Define transformaiton	Applies an affine transformation to the system.

Define displacement	Apply random displacement /perturbation to atoms.
Wrap periodicity	Folds particles located outside of the periodic simulation box back into the box.
Optimize triclinic geometry	Limit tilt factors that skew the box less than half the distance of the parallel box length
Find crystal symmetries	Get the space group symbol, group number, or create the standardized primitive cell
Replicate box	Duplicates particles to periodic images of the system.
Change atom type	Assign the element type to atoms

8. Potential plotting

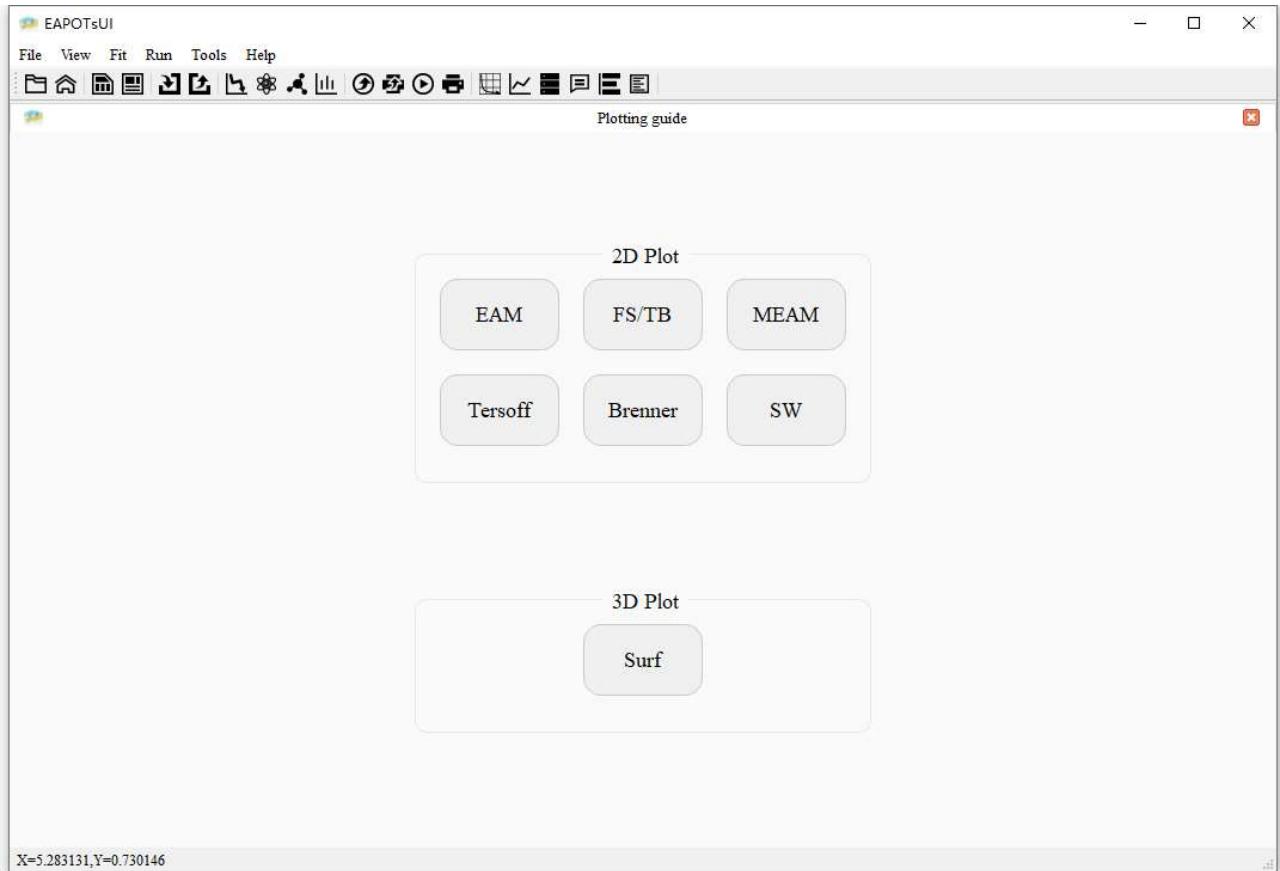


Fig. 8.1 Screenshot of the potential plotting window

Fig. 8.1 illustrates the potential plotting window, which is used for the visualization of potential function curves and potential energy surface.

8.1. Potential function curves

Posted below are two examples to show the potential function curves. The EAM potential function curves are shown in Fig. 8.2, which contains: (a) pair interaction function, (b) embedding energy function, and (c) electron charge density function.

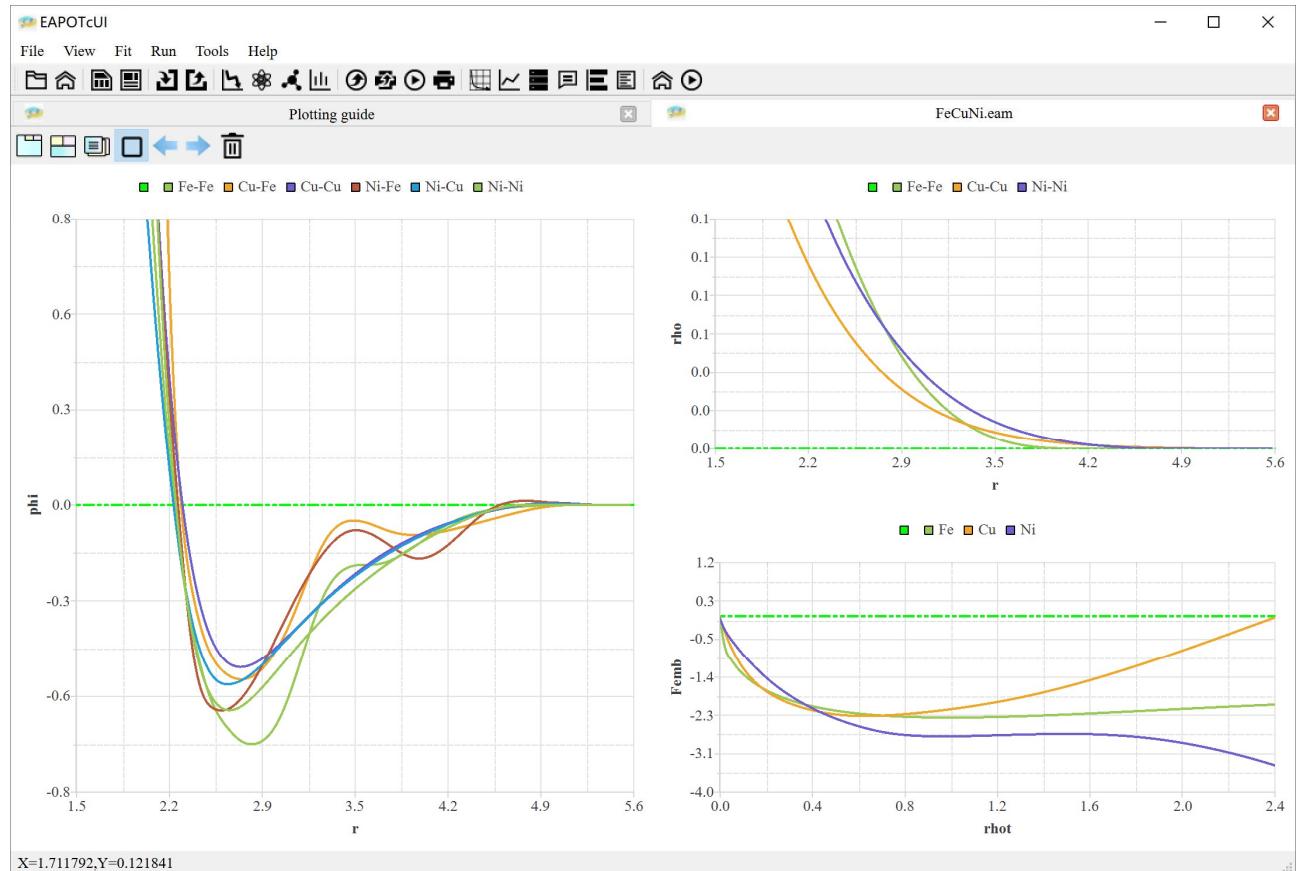


Fig. 8.2 Snapshot of EAM potential function curves

Fig. 8.3 shows the SW potential function curves, in which contains: (a) 3-body function, (b) angular function, and (c) 2-body function.

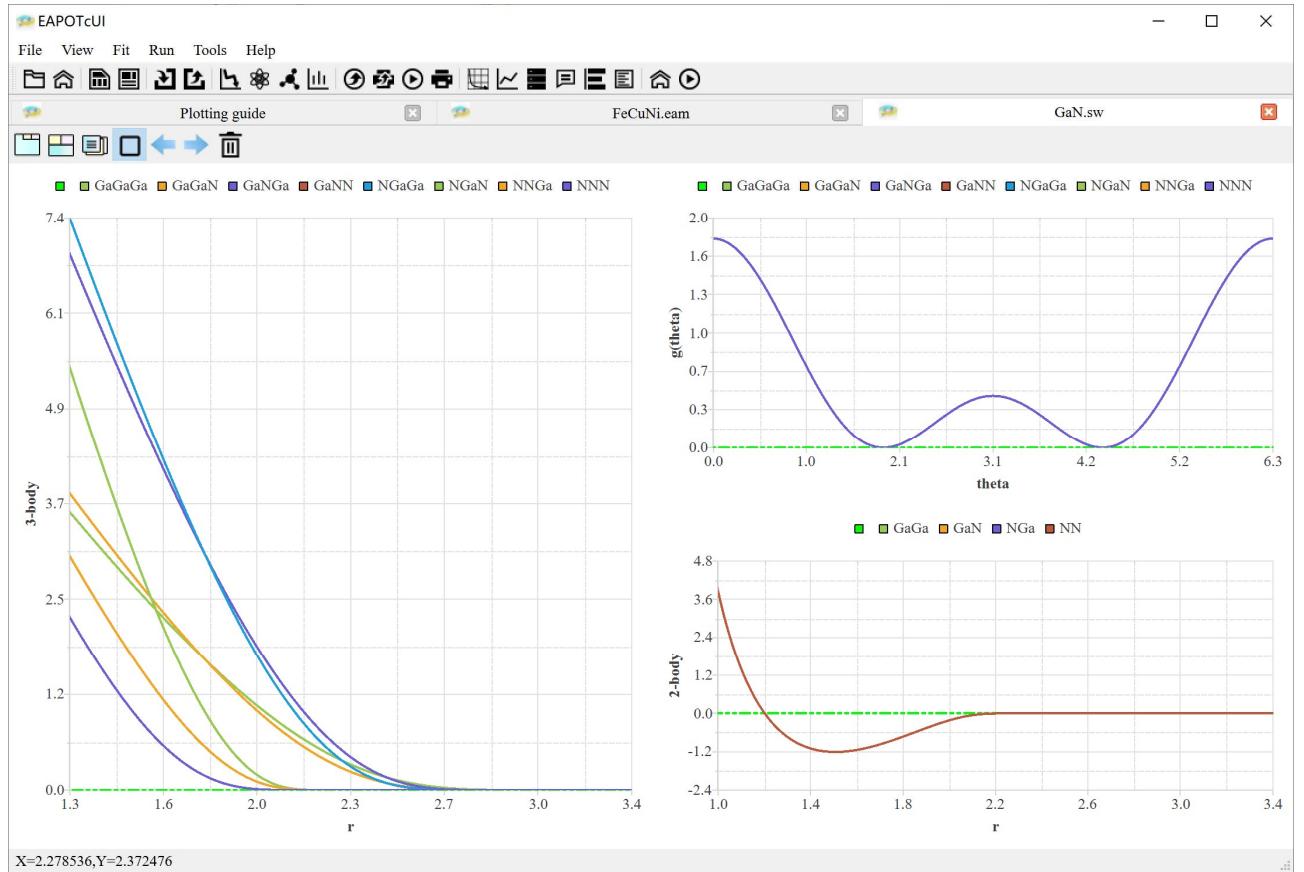


Fig. 8.3 Snapshot of SW potential function curves

8.2. Potential energy surface

For energy surface visualization, Fig. 8.4 and Fig. 8.5 exhibit the typical γ surface of the compact {111} plane for Cu metal and the generalized SFE curve along with the [112] directions with marked unstable stacking fault energy (USFE). Here, the USFE represents the direct energy barrier required for the sliding between two neighboring atomic planes, and the intrinsic stacking fault energy (ISFE) represents indirect resistance by restricting the width of the stacking fault.

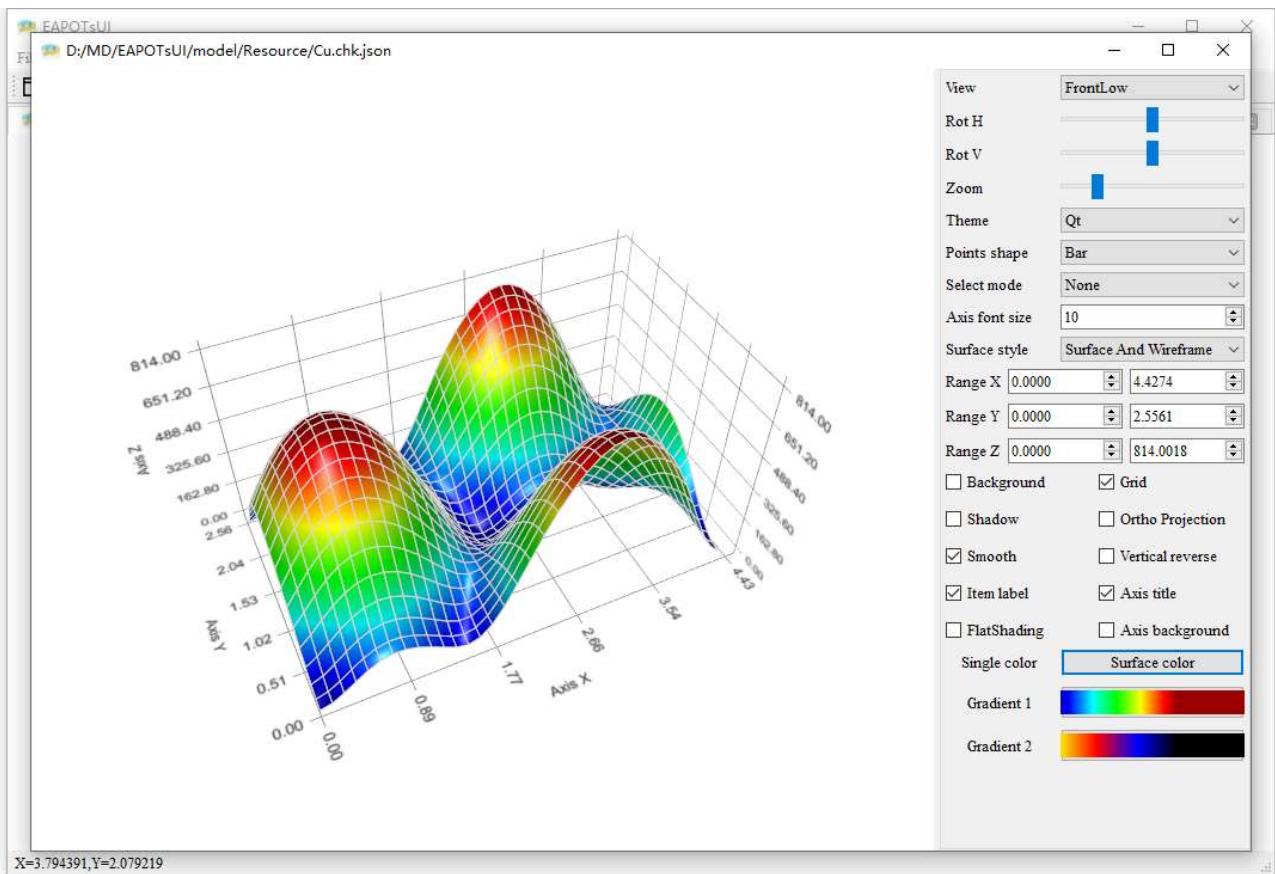


Fig. 8.4 The typical γ surface of the compact {111} plane for Cu metal

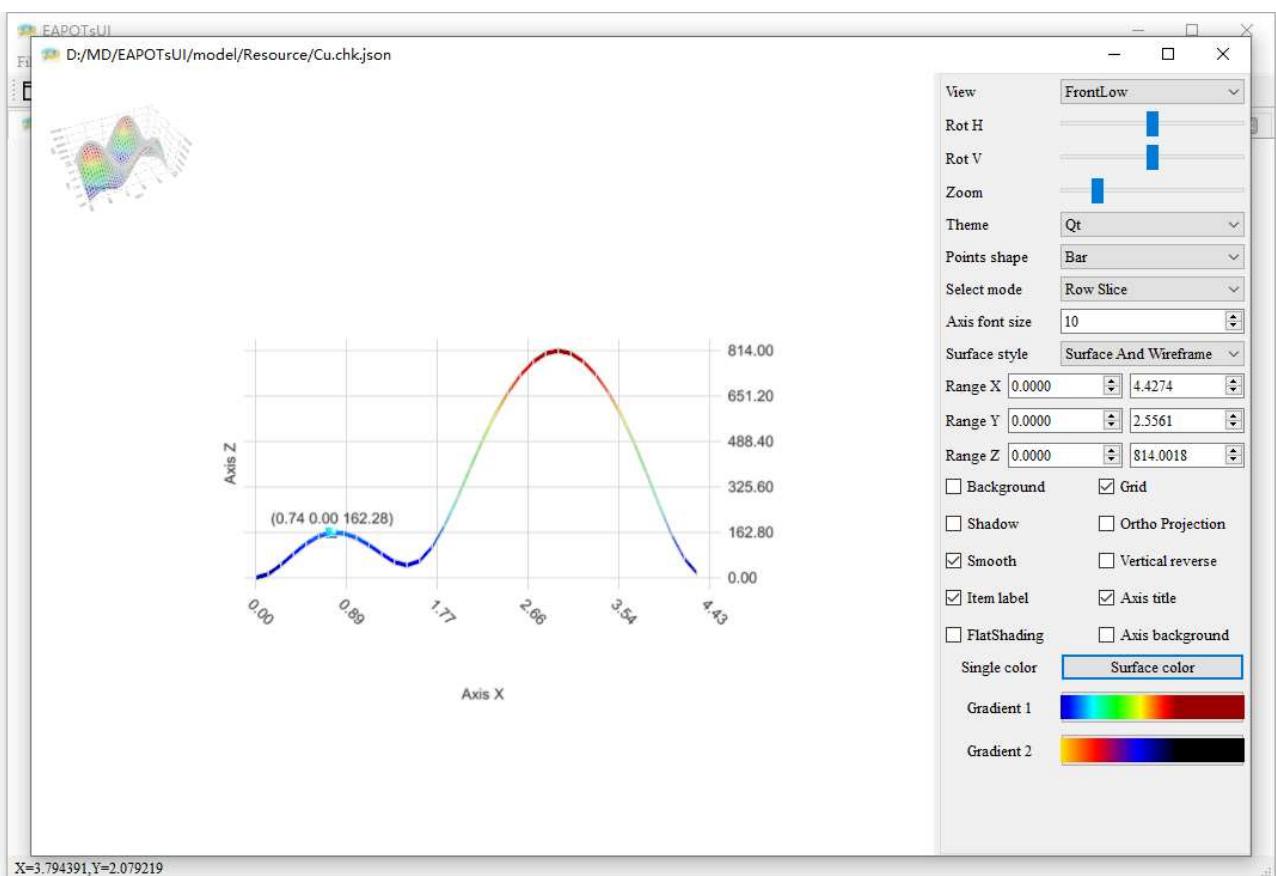


Fig. 8.5 The generalized SFE curve along with the [112] directions