

MedeA Forcefield Optimizer: Efficient Forcefield Parameter Optimization using First-Principles Data

Contents

- [Introduction](#)
- [Supported Forcefield forms](#)
- [Working with the Forcefield Optimizer](#)
- [Running the Forcefield Optimizer Job](#)
- [Parameter Definition and Combining Rules](#)

1 Introduction

The *Forcefield Optimizer* employs first-principles computed properties to create and refine forcefields for use in energy minimization and molecular dynamics simulations. You can use the *Forcefield Optimizer* to expand the scope of atomistic simulations to systems with thousands of atoms and simulations with millions of configurations. This expansion in system length and timescales is obtained through the use of empirically parameterized energy functions designed to reproduce quantum mechanically derived information with substantially reduced computational complexity. Naturally, parameterized forcefields are less flexible and general than quantum mechanical approaches. However, when combined with first-principles validation, forcefield methods can be highly effective. The *Forcefield Optimizer* allows you to create, optimize, and validate forcefields based on first-principles derived training information.

Training sets for the *Forcefield Optimizer* typically consist of trajectories from VASP molecular dynamics, geometry optimizations, or single point energy calculations. Such sources of information are chosen to sample the desired configuration space for which the forcefield will be employed. The *Forcefield Optimizer* automates the process of selecting parameters for optimization, the handling of constraints, the details of the optimization process, and the analysis of optimization results. A variety of reports of the quality of fit obtained are provided, including graphical plots of training set properties, such as comparison of target and computed energies and forces, permitting direct comparison of the forcefield and first-principles results.

The *Forcefield Optimizer* supports, Buckingham, embedded atom method (EAM), Tersoff, Stillinger-Weber, COMB3, ReaxFF, and PCFF+ forcefield forms. Each of these forcefield forms in the context of the Forcefield Optimizer is briefly reviewed below.

2 Supported Forcefield forms

2.1 Buckingham

As discussed briefly in section Forcefields for materials simulations of this User Guide, for ionic systems the following simple functional form provides an effective description of interatomic interactions:

$$E = \sum \left(f \frac{q_i q_j}{r_{ij}} + A_{ij} e^{\frac{r_{ij}}{\rho_{ij}}} - C_{ij} r_{ij}^6 \right) \quad (1)$$

Here, E is the total energy of a system. The adjustable parameters are the atomic charges q_i and the pair-parameters A_{ij} , r_{ij} , and C_{ij} . The factor f converts the Coulomb term into appropriate energy units when the charges are given in atomic units and r_{ij} in Å. The effective charges q_i and q_j play a critical role since in compounds such as oxides electrostatic interactions represent the major bonding component of interatomic interactions. The exponential term, with the two pair-wise parameters A_{ij} and ρ_{ij} , captures i repulsion as charge clouds overlap. The van der Waals term of the form $-C/r^6$ describes attractive interactions due to dispersion between polarizable ions. The summation extends over all possible pairwise interactions in the systems. Typically for ionic systems, a long-range convergence acceleration algorithm such as the PPPM or Ewald method will be used to ensure the accurate evaluation of Coulombic interactions.

2.2 Embedded Atom Model (EAM)

For metallic systems the embedded atom model (EAM) forcefield form has proved effective in describing many properties with low computational cost and accurate property predictions.

In the EAM, the total energy is expressed as two terms, as illustrated in equations 4.27 and 4.28 in section III of this User Guide. These energy terms represent pairwise interactions between an atom and its neighbors and an embedding function that depends on the electron density present at a given atomic location within the simulation.

The density each atomic location is evaluated based on a function of the form of Equation 2, where the parameters ρ_0 and α are obtained from first-principles simulations.

$$\rho(r) = \rho_0 e^{-\alpha r} \quad (2)$$

The electron density, ρ_i , at a given site is given by the local sum of adjacent atom density functions, using Equation 2.

The embedding function, $F(\rho_i)$, for a given site may be represented by a variety of functional forms. A convenient variant, employed by the *Forcefield Optimizer* is:

$$F(r_i) = ar_i + br_i^2 + c(e^{dr_i} - 1) \quad (3)$$

Each element has its embedding parameters a , b , c , and d .

The form of the Morse-like, universal binding energy curve [1] is used for the pair-wise interactions:

$$V = D_0 \left(e^{-2b(r-r_0)} - 2e^{-b(r-r_0)} \right) F_{cut} \quad (4)$$

$$F_{cut} = \frac{1}{2} (1 + \cos(p(r - r_{cut}))) \quad (5)$$

with three parameters, namely D_0 , r_0 , b . Each pair of elements has its own set of these three parameters. The additional function F_{cut} ensures a smooth quenching of the interaction function to zero at $r=8$ Å, here $r_{cut} = 7$ Å and is not modified by the *Forcefield Optimizer*.

Note: The *Forcefield Optimizer* handles EAM embedding and pair interaction functions by interpreting the functional form supplied in the *.frc* file. Hence the functional forms that are represented in the template EAM forcefield file may be adjusted for specific applications by modifying the appropriate function line in the *.frc* file. The syntax employed in specifying the function is that employed by the `tcl expr` command.

2.3 Tersoff and Stillinger-Weber Forcefields

Tersoff and Stillinger-Weber forcefields are well-established forcefields for semiconductor materials. Tersoff forcefield has the form:

[1] J H Rose, John Ferrante, and John R Smith, "Universal Binding Energy Curves for Metals and Bimetallic Interfaces," *Physical Review Letters* 47, no. 9 (August 31, 1981): 675-678.

$$E = \frac{1}{2} \sum_i \sum_{j \neq i} V_{ij}(r_{ij}) \quad (6)$$

$$V_{ij}(r_{ij}) = V_{ij}(r_{ij})^R + b_{ij} V_{ij}(r_{ij})^A \quad (7)$$

Tersoff forcefield has 14 parameters per element.

Stillinger-Weber forcefield has the form:

$$E = \frac{1}{2} \sum_i \sum_{j \neq i} \Phi_2(r_{ij}) + \sum_i \sum_{j \neq i} \sum_{k > j} \Phi_3(r_{ij}, r_{ik}, \theta_{ijk}) \quad (8)$$

Stillinger-Weber forcefield has 11 parameters per element.

2.4 COMB3 Forcefields

The 3rd generation charge-optimized many-body (COMB3) [2] forcefields are improvements over the previous generations of COMB forcefields. COMB3 contains an advanced bond order term for describing complex chemical reactions (bond breaking and formation), Coulomb with charge density described with Slater-type orbitals and variable charge equilibration (atomic charges automatically assigned based on atomic surroundings).

2.5 ReaxFF Forcefields

Reactive Forcefields (ReaxFF) [5] is a family of well-established forcefields that simulate complex chemical reactions and charge transfer. It includes advanced bond terms over valence terms, shielded Coulomb, and variable charge equilibration.

2.6 Class II Forcefields

Organic systems, Class II forcefields, of which PCFF+ is a leading example, provide high accuracy in reproducing experimental properties. The functional form of Class II forcefields has been described by Maple and coworkers [3].

The forcefield is composed of valence terms that describe the distortion of the covalently bonded framework of organic molecules, electrostatic interactions modeled using partial atomic charges, and non-bond interaction terms characterized by r^{-6} attractive terms and r^{-9} repulsive terms.

As is the case for Buckingham, EAM, Tersoff, Stillinger-Weber, and COMB3, the Forcefield Optimizer only presents parameters for optimization for Class II forcefields that are relevant to the description of the current system.

3 Working with the Forcefield Optimizer

The creation and optimization of forcefields typically proceed via three discrete stages. Firstly a 'training set' or first-principles properties are obtained. Secondly, optimization calculations are undertaken whereby initial parameters are adjusted to minimize the deviation between forcefield and first-principles derived properties. Finally, validation is conducted to assess the ability of the forcefield to reproduce properties not directly included in the training set.

[2] T. Liang, T.-R. Shan, Y.-T. Cheng, B. D. Devine, M. Noordhoek, Y. Li, Z. Lu, S. R. Phillpot, and S. B. Sinnott, *Mat. Sci. & Eng: R* 74, 255-279 (2013).

[5] A.C.T. van Duin, S. Dasgupta, F. Lorant, and W. A. Goddard, ReaxFF: A reactive force field for hydrocarbons, *Journal of Physical Chemistry A* 105, 9396-9409 (2001); Chenoweth, A.C.T. van Duin, and W.A. Goddard, ReaxFF reactive force field for molecular dynamics simulations of hydrocarbon oxidation, *Journal of Physical Chemistry A* 112, 1040-1053 (2008)

[3] J R Maple, U Dinur, and A T Hagler, "Derivation of Force Fields for Molecular Mechanics and Dynamics From Ab Initio Energy Surfaces," *Proceedings of the National Academy of Sciences* 85, no. 15 (August 1, 1988): 5350-5354; J R Maple, M J Hwang, T P Stockfisch, U Dinur, M Waldman, et al., "Derivation of Class II Force Fields. I. Methodology and Quantum Force Field for the Alkyl Functional Group and Alkane Molecules," *Journal of Computational Chemistry* 15, no. 2 (February 1994): 162-182.

Here we illustrate the use of the *Forcefield Optimizer* to develop a Buckingham forcefield for magnesium oxide, MgO.

A Buckingham forcefield can be employed to describe the crystalline and molten forms of MgO and the training set is developed with this objective in mind.

3.1 Before Carrying Out Forcefield Optimization

First, an appropriate forcefield file must be read into *MedeA*. In this case, read the file `../MD/data/Forcefields/custom/MgO_template.frc` using `Tools/Forcefields/Read...`. This file contains starting values for the parameters Buckingham forcefield parameters, $A_{\text{Mg-O}} = 1000$ eV and $\rho_{\text{Mg-O}} = 0.3 \text{ \AA}^{-1}$. The atomic charges are initialized with the formal charges of +2 and -2, respectively. The other parameters A_{ij} are set to 0 and the inverse radii ρ_{ij} for the interactions Mg-Mg and O-O are kept at a non-zero value, in this case, 0.3 \AA^{-1} . The coefficients C_{ij} for all the dispersion terms are set to zero. These parameter settings are contained within the supplied `custom/MgO/template.frc`, and these parameters serve as a starting point for optimization.

3.2 Preparing a First-Principles Based Training Set

The next step is the creation of a training set. To this end, a conventional MgO unit cell is loaded from *InfoMaticA* and a 2x2x2 supercell is created. By default, the symmetry of this supercell is P1, which is necessary for the following molecular dynamics run. Before starting the VASP calculation, the forcefield template of MgO is loaded from the Forcefield directory `../MD/data/Forcefields/custom` using `Tools >> Forcefields >> Read...` into *MedeA* and the atom types and charges are assigned to the MgO supercell.

Now a VASP molecular dynamics run is computed for the MgO supercell. In the present example, it is recommended that this is done with a GGA-PBESol functional, a soft O PAW potential, one k-point, normal precision and a real-space integration. The VASP molecular dynamics simulation is performed with temperature scaling at 2000 K for a total of 100 fs with a time step of 5 fs, resulting in a trajectory with 20 configurations.

Using this *ab initio* molecular dynamics result, a structure-list file (`.list`) is created, which contains the cell parameters, atomic coordinates, total energy, forces on each atom, and the stress tensor for each of the 20 configurations, using the `Builders/Structure List file...` command and adding structures from the appropriate VASP job.

3.3 Setting up a Forcefield Optimizer Calculation

After the selection of an appropriate `.frc` template and the preparation of the `.slst` file containing the training set, the Forcefield Optimizer capability can be invoked within a *MedeA* Flowchart. This is achieved using the `Jobs >> New Job...` command and adding a *Forcefield Optimizer* stage to the resulting flowchart.

In the Flowchart, double-click on the *Forcefield Optimizer* stage to open a dialog to load the `.frc` template. Here the `MgO_template.frc` is selected. Clicking on the "Import" box enables the loading of the `.slst` file, which is the `MgO.slst` file created above. Once the `.slst` file is loaded, *Forcefield Optimizer* displays a window where the terms from the training set to be included in the optimization can be selected. For the present example selected just 'Forces' in the Fitting Data tab of the *Forcefield Optimizer* stage.

3.4 Setting Forcefield Parameters

The Forcefield tab of the *Forcefield Optimizer* stage can be used to set the parameters to be refined, as well as setting the lower and upper limits for their optimization. In this example, select the charge factor and the parameters A and Rho for the Mg-O pairwise interactions for optimization.

3.5 Setting Optimization Options

For a rapid illustration of the use of the *Forcefield Optimizer*, the evolutionary algorithm parameters **Population Size** and the **Number of Generations** may be set to zero. This will accelerate parameter optimization calculations. A maximum of 20 iterations in the subsequent conjugate gradient optimization should also be selected. (A larger maximum may be necessary for more complex cases.)

4 Running the Forcefield Optimizer Job

Now the forcefield optimization can be initiated using the Run button on the Flowchart dialog box. (Give the Job a suitable title such as 'MgO parameter optimization'). The progress of the optimization can be monitored by inspecting the file optimization.log in *Stage_1* in the corresponding Job on the JobServer. On completion of the optimization (which should take approximately 5 minutes on a typical laptop), the *Job.out* file will report details of the least-squares fit obtained.

Note: Note that the optimization procedure results in an effective charge on the ions that is smaller in magnitude than the formal charges.

The optimized forcefield parameters are written to the file "optimized.frc" which is stored in the current Job folder. This newly created forcefield file is available for use with *MedeA* LAMMPS. You may find it convenient to copy this file to your */MD/data/Forcefields/custom* folder, and perhaps rename the file for future reference using your preferred file manager.

4.1 Assessing the Fit Obtained by the Forcefield Optimizer

The *Job.out* file of the Forcefield Optimizer calculation provides information on the course of the evolutionary algorithm search of parameters space (if this option is employed) and on the quality of the overall agreement between forcefield and first-principles fitting variables. After the listing of the final parameters obtained, you will find a set of reports of the deviations obtained between forcefield and first-principles derived properties, with sections for the energies, forces, and stresses. In each case the initial root mean squared deviation for this fitting target will be reported, and additionally the parameters of a least-squares regression between target and fitted values will be reported. In the ideal case, such regressions will show unit slopes (i.e. $a=1.0$), intercepts of zero (i.e. $b=0.0$), and R values of unity (i.e. $R=1.0$). However, except in simple cases, these values will in practice deviate from ideality, and it will be necessary to choose a compromise description, which focuses on a particular set of target observables, for example. The reported fitting parameters enable appropriate selections and decisions about fitting strategies to be made.

In addition to the summary statistical information contained in *Job.out*, the graphic output is also provided, in .png files named training_energy_pat_vs.png, for example. Here the naming shows that this particular plot compares training and fitted energies on a per-atom basis, for example, with similar naming conventions for additional comparisons. As noted above, such plots should show a linear relationship, with a unit gradient and zero intercepts, if the fit obtained is perfect. The graphical output allows you to determine if there is an outlying data point or group of data points, for example, and modify your fitting strategy if appropriate.

The standard estimated errors for fitting parameters are reported in the *Job.out* file. This quantity for each parameter is useful for assessing the quality of fit obtained. It shows the precision with which the least-squares method can estimate that parameter, and should be a fraction of the reported parameter itself. Estimated errors that exceed the parameter values themselves are a cause for concern and show that these forcefield parameters are only weakly determined by the available first-principles information.

Additional report files, written to the stage folder for the Forcefield Optimizer job, contain additional details of the least-squares fitting, including the correlation matrix. This information can help identify situations where multiple variables provide a similar effect on the sum of squared deviations between the forcefield and first-principles properties, and cannot be readily separated by the available target information. This can occur, where there are relatively few fitting points and parameters have similar impacts on the energy description, an

example being the A and ρ repulsive parameters of the Buckingham potential, which both affect the general repulsion between species and a variation in one can be compensated by a change to the other.

Having assessed the performance of a given fitting calculation, you can decide how to proceed. You may be judge, for example, that the resulting forcefield achieves satisfactory agreement with the first-principles training set and you can proceed to validate the forcefield and to carry out production calculations using *MedeA* LAMMPS. Alternatively, further fitting can be carried out, for example, by adjusting the weights on target observables, and conducting additional fitting calculations.

4.2 Validating the Optimized Forcefield

You can use the optimized forcefield to compute, for example, the density of MgO at a defined temperature. In the present case, for example, you should find a value that is close to the PBEsol density value of 3.59 g/cm³.

The MgO example illustrates the general use of the *MedeA Forcefield Optimizer*. The procedures employed are entirely analogous for EAM and Class II forcefields. Bulk MgO is a relatively straightforward system, where the inherent approximations of the Buckingham forcefield are reasonable and therefore a good least-square fit can be obtained relatively easily. For more complex systems, for example in developing a forcefield capable of describing both bulk and surface structures, care must be taken in the construction of the training set that is employed as the target of the optimization process. Its components must adequately provide information on the energy, forces, and stresses of the system of interest, to enable the least-square optimization process to proceed.

One also needs to keep in mind that any DFT calculation carries errors, which are due to the approximations in the underlying many-body theories. One possible method to overcome these shortcomings is a calibration of *ab initio*-based forcefields by sensitive experimental data such as the melting temperature. This has recently been successfully demonstrated for the case of Li₂O. [4] Such a calibration procedure is by no means new and similar calibration has been employed in the development of Class II forcefields for organic molecules for several years.

5 Parameter Definition and Combining Rules

5.1 Buckingham Forcefields

The Buckingham forcefield form relies on the specific definition of the pairs to be included in the forcefield file. For example, in the case of MgO, you could specify Mg-Mg interaction parameters, though *MgO.template.frc* does not presently contain such terms. In practice though, cation-cation interactions are adequately described with Coulombic interactions so this is not typically necessary for forcefields for ionic systems. The *Forcefield Optimizer* will optimize only the non-bond interactions explicitly described in the supplied forcefield file.

5.2 EAM Forcefields

In contrast to the Buckingham Forcefield form, the Materials Design EAM forcefield form for use with the Forcefield Optimizer (exemplified by the *md_eam.template.frc* file) allows for either the use of a combining rule derivation or explicit inclusion of cross interaction terms. Hence, for a given forcefield to represent Cu/Zr alloys, the Cu-Zr interaction term may be either explicitly included in the supplied forcefield file, or if these terms are absent from the file, they are created by *MedeA* using a 'combining rule' where the D_0 , r_0 , and β of that particular interaction are obtained as the arithmetic mean of the corresponding terms for the pure element interactions.

[4] Ryoji Asahi, Clive M Freeman, Saxe, and Erich Wimmer, "Thermal Expansion, Diffusion and Melting of Li₂O Using a Compact Forcefield Derived From Ab Initio Molecular Dynamics," *Modelling and Simulation in Materials Science and Engineering* 22, no. 7 (September 25, 2014): 075009-16.

When the Cu-Zr terms are explicitly included in the fitting process, there will be more adjustable terms, and accordingly potentially improved agreement with first-principles data. When Cu-Zr terms are omitted from the *md_eam_template.frc* file, only the pure element parameters can be adjusted, simplifying the forcefield, and reducing the degrees of freedom available to the Forcefield Optimizer. The decision of whether to rely on combining rules or to explicitly include cross-term parameters will require analysis of the details of the systems to be simulated using the forcefield. A robust and typical approach is to employ as few variables as possible initially in the fitting process and to introduce additional variables as needed, with the expectation that additional variables may improve the fitting performance for a particular training set but may also lead to 'over fitting', manifest as difficulties in reproducing the properties of systems not directly included in the training set.