MEAMfit User Guide

Andrew Ian Duff and Marcel H. F. Sluiter

Contents:

1. Citation

The code is free to use, but if you use it for publication then cite:

A. I. Duff, et al, Comp. Phys. Comm. (201…)

2. Features

The code is intended for fitting interatomic potentials to DFT data.

Features:

- Fits MEAM and EAM potentials [Baskes, 1999] in the reference-free format [Timonova, 2011];
- Fitting to atomic forces and/or energies read directly from VASP output files;
- Potential output directly to LAMMPS (EAM) or Camelion (MEAM) formats;
- Existing potentials can be read in and used as basis for further optimization.

3. Compiling

Unpack MEAMfit.tgz, and enter MEAMfit/src via the command-line. *MEAMfit* is installed by typing 'csh installmeam'. Note, by default the intel Fortran compiler is supported. To instead use the gnu Fortran compiler, please change the appropriate "FC=" and "FFLAGS=" flags in 'installmeam'. For example, if you are using a Linux system then search for "Linux", and then change the lines below this to: " set FC='gfortran' "; and " set FFLAGS='-O3' ". Once you have an executable, which will be named MEAMfitcode, it is convenient to alias it for easy use. For the remainder of this manual we will assume that the executable has been aliased to the command: MEAMfit.

4. Running

4.1) Setting up the fitting set

Place the vasprun.xml file/s containing the energies/forces which you would like to fit to in a new directory and run MEAMfit in that directory. MEAMfit will automatically detect these vasprun.xml files, but make sure that any labelling (e.g., 1, 2, etc) is placed after the word 'vasprun' and before '.xml' (e.g.: vasprun1.xml). MEAMfit will generate a file 'fitdbse' and then terminate. For the case where a single vasprun.xml file is supplied for example, 'fitdbse' will look like:

```
1  # Files | Configs to fit | Quantity to fit | Weights
vasprun1.xml   1-660   Free-Energy   1
```

The number on the first line is the number of vasprun.xml files found in the directory. A subsequent line is then included for each vasprun.xml file in the directory. These lines must contain no more than 80 characters each. The first column of these lines contains the name of the vasprun.xml file, in this case 'vasprun1.xml'.

The second column shows the number of configurations to be fit to from within this file. By default, all configurations are included in the 'fitdbse' file, so that in the present example 660 configurations were found in the vasprun file. The configurations to be fit to can be

adjusted by the user however, so that for example '1-100;102-160s2' will fit configurations 1-100 and 102, 104, …, 160 (where 's' denotes the step between adjacent configurations, and where ';' rather than a ',' should be used to separate ranges). The overall number of vasprun.xml files to fit to can also be adjusted by changing the number on the first line.

The third column determines the property in the vasprun file to be fit to, and can take the values:

'E0': to the total energy (specifically the E0, sigma->0 value in the vasprun file);

'Free-Energy': will fit to the free energy (the F value in the vasprun file);

'Force': will fit to atomic forces.

Note that only the first two letters are in fact read in by *MEAMfit*, so that it is sufficient to write 'Fr' or 'Fo' for the second and third cases respectively.

IMPORTANT: if a force-fit is being commenced, be aware that by default, *MEAMfit* runs with the FASTFORCE tag equal to true (see Section 6 for more details). This is very memory intensive, so if you encounter a memory error, or in general if *MEAMfit* crashes, add the tag "FASTFORCE=false" to the settings file and try again.

A simultaneous fit to the energies and forces of a particular file would require two separate lines in the fitdbse file.

The fourth column specifies the weight, w_i, of Equation 9 of the *MEAMfit* paper, used when computing the optimization function for the energies/forces read from that file. For energies, energy differences rather than absolute energies are included in the optimization function, with the differences taken with respect to the first configuration read in from the fitdbse file (in the above example this is configuration 1 of vasprun1.xml).

Fitting selected forces:

To fit to only a selection of forces within a vasprun file, go to the appropriate lines in the file (search for "forces"), and place a "t" character at the start of those lines containing forces which you would like to fit to. Note however that other vasprun.xml files specified in the 'fitdbse' file for force-fitting will still have all of their forces fit to unless they too have been amended with the addition of "t" characters.

4.2) Choosing the potential

Run *MEAMfit* again to create a 'settings' file, which is used to define the type of potential to be optimized as well as the parameters to be used in the optimization. The 'settings' file uses a tag-based system for specifying input parameters. A full description of these tags is given in Section 6, however for the present purposes a brief introduction is given only of those tags necessary to get the code running. The 'settings' file will read:

```
TYPE=EAM
# CUTOFF_MAX=
NTERMS=3
```

NTERMS_EMB=3

TYPE determines the type of potential to be optimized, chosen here to be EAM.

CUTOFF_MAX defines the maximum cut-off radius (units: Å) to be used for the pairwise terms in the fitting of the potential. This line needs to be commented back in (remove the # symbol), with an appropriate value put in. To help decide on a value for this parameter, *MEAMfit* can be run with the –noopt (no optimization) flag to generate a distribution of interatomic separations read in from the vasprun.xml files. To do this, put in a starting value of CUTOFF_MAX (large enough to survey all possible separations of interest) and type:

MEAMfit –noopt

to produce the file sepnHistogram (alternatively add the line "NOOPT=true" to the settings file). This file contains histogram data, with the first column containing the interatomic separation bins, and the remaining columns showing the numbers of separations (per species pair) falling into these bins. The file can be visualized straightforwardly with gnuplot, for example as in Figure 1. Here for example, one sees that setting CUTOFF_MAX=4.4 would generate a potential for which interactions up to 3rd nearest neighbour are included.

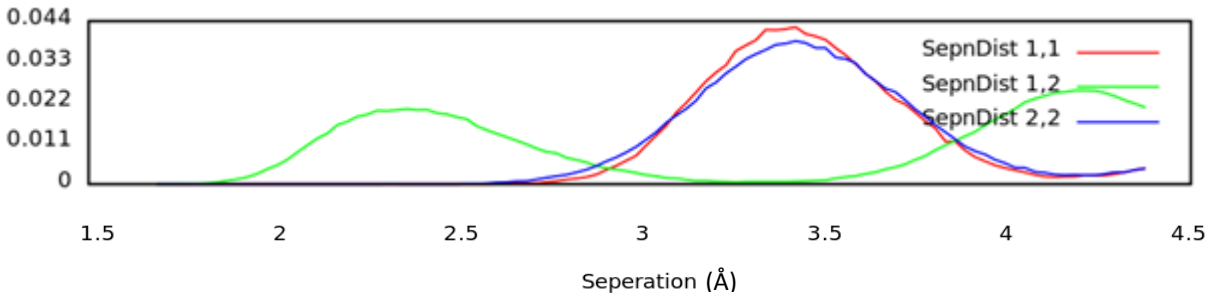

Figure 1: Histogram of interatomic separations for a ZrC rocksalt MD-run at 3805K. Here, one sees choosing CUTOFF_MAX=4.4 would result in a potential with up to 3rd nearest neighbour interactions included.

NTERMS determines the number of pairwise terms used in the expansion of the electron densities and pair-potentials (equations 7 and 8 of the *MEAMfit* paper). In this example, 3 pairwise terms are selected for each of the electron densities and pair-potentials (with each pairwise term containing two parameters). This would correspond to 12 pair-wise parameters for a unary (6 parameters for both $\rho_X^{a(0)}$ and $V_{X,X}^{p.p}$), and 30 pair-wise parameters for a binary (6 parameters for each of $\rho_X^{a(0)}$, $\rho_Y^{a(0)}$, $V_{X,X}^{p.p}$, $V_{X,Y}^{p.p}$ and $V_{Y,Y}^{p.p}$).

NTERMS_EMB determines the number of terms to be included in the embedding function (equation 2 of the article). A value of 1 will optimize a potential containing only a_α terms, 2 will optimize potentials containing a_α and b_α, and 3 will optimize potentials containing a_α, b_α and c_α.

In addition to the above, the parameter: OPTFUNCCG, may need to be tuned. By default this is takes the value 10, but may require adjusting if the initial random phase of sampling is too fast or too slow. A full description of this parameter, as well descriptions of all settings available in *MEAMfit* will be presented in Section 6.

<u>Starting from an existing potential</u>

To use an existing potential to supply some or all of the potential parameters one can use the POTFILEIN command (see Section 6 for details). A potential can be supplied either to: i)

use as a starting point from which to optimize a new potential; or ii) define a fixed part of a new potential (e.g., to define the X-X part of a binary potential). For (i), the parameters read in will be allowed to optimize, starting from their values as specified in the input potential file. This is the default mode. For (ii), one must add the line "FIXPOTIN=true" to the 'settings' file, which will fix the values of the parameters read in from the input potential file and not allow them to optimize during the subsequent optimization.

Note: only the non-zero parameters of a potential will be read in from an input potential file. If other parameters have been specified to be optimized (using either the NTERMS or PARASTOOPT tags in the 'settings' file) then these will be randomly generated as usual.

If only some of the supplied potential parameters are to be optimized (with the rest held fixed), the flexible PARASTOOPT tag can be used, which enables one to choose for each potential parameter in turn whether it is to be read-in from file or randomly generated and whether it is to be allowed to optimize or be held fixed. A full description of the syntax is provided in Section 6.

4.3) Optimization

Once the 'fitdbse' and 'settings' files have been set-up *MEAMfit* can be run again to start the optimization. The program will begin by writing to the screen information about the fitting-database, general settings from the 'settings' file and information concerning the potential parameters which are to be randomly generated. This information should be carefully checked by the user to ensure it is consistent with expectation. Sample output can be found in the Examples folder.

After this, optimization will commence. The first stage of optimization involves generating potentials from random seeds. Potentials are randomly sampled until one of them yields an optimization function less than the value specified by the OPTFUNCCG tag in the settings file (see Section 4.2). Output will look like:

```
Random potential parameters sampling... (current optfunc:      35.3293971358      , lowest optfunc:
35.3293971358   >  20.0000000000   )

 Random potential parameters sampling... (current optfunc:      74.0714594560      , lowest optfunc:
35.3293971358   >  20.0000000000   )

 Random potential parameters sampling... (current optfunc:      23.0792594179      , lowest optfunc:
23.0792594179   >  20.0000000000   )

 Random potential parameters sampling... (lowest optfunc:   15.9909549686   <  20.0000000000   )
```

In this case, only four attempts were needed to achieve a suitably small optimization function (although in general many more will be needed, see Section 4.2). The potential is then minimized using a conjugate gradient scheme:

```
   it   nf      f        reldf     preldf     reldx   stppar   d*step   npreldf
    0    1  0.160D+02
    1    4    0.146D+02    0.838D-01    0.178D+01    0.436D-O2    0.3083D+04    0.960D-
01    0.274D+04
...
```

where ‘it’ and ‘nf’ keep track of the number of iterations and function evaluations respectively, and where ‘f’ shows the current value of the optimization function. This process of randomly generating and optimizing potentials is repeated, and the values of the corresponding optimization functions are recorded both in the standard-output and in the ‘bestoptfuncs’ file. The latter will look like:

```
1 :  8.563364725131709E-002  time:   11.0000000000000      hours
2 :  8.605524477835252E-002  time:   16.0000000000000      hours
3 :  9.072688201965821E-002  time:   22.0000000000000      hours
4 :  9.638533881176091E-002  time:   5.00000000000000      hours
5 :  0.103553338844121       time:   27.0000000000000      hours
6 :  0.144635662422224       time:   22.0000000000000      hours
Total time taken    27h 54m 11s.
```

Once ten potentials have been generated in this way (this default value can be changed, see Section 6), the next stage in the optimization will commence. In this stage, the ten potentials generated from the first stage are optimized further by a genetic algorithm scheme: Two potentials are randomly selected and mixed (and occasionally mutated) and then optimized using the conjugate-gradient minimization scheme. If the resulting optimization function is smaller than any of the other ten potentials then the new potential is retained in the ‘top 10 list’ (with the other potentials ‘shuffled down’ and the previous occupier of the number 10 position being dropped). The process is then repeated until either a convergence criterion is satisfied or a certain length of time has passed (see Section 6).

The potentials are written out to file as they are generated (and so can be used or tested as and when they are generated) and are named as ‘potparas_bestN’, where N=1-10. The corresponding fitted energies/forces are also stored in the corresponding files ‘datapnts_bestN’. Be careful however: as the potentials are re-ordered during the running of the code (as more favourable potentials are found), so these potential files will also be re-ordered. The ‘potparas’ files contain all of the parameters used to specify a given potential (see Appendix A for the exact layout), and files of this type can be read into *MEAMfit* for the purposes of evaluating the performance of an optimized potential within a testing-set (see Section 7), or for performing a continuation job (see Section 6).

Potentials are also written out in the eam.alloy format of LAMMPS (e.g. ZrC.eam.alloy_N, where N=1-10) and the … format of Camelion (e.g.: _N, where N=1-10), enabling immediate use of these potentials with these codes.

5. Worked example

5.1) Introduction

To demonstrate energy and force-fitting using *MEAMfit*, a worked example is provided in which EAM and MEAM potentials are fit to ZrC DFT-MD data generated using VASP. This

worked example should be carried out in the MEAMfit/SampleCalculation/Input directory (or for convenience, in a copy of this directory). Sample output is provided in MEAMfit/SampleCalculation/Output.

5.2) EAM energy optimization

Enter Input/EAM and run *MEAMfit*. A 'fitdbse' file will be generated and the code will stop. This file lists all the vasprun.xml files, along with the number of ionic configurations in each, and should be edited to define your 'fitting-set', i.e. the data you want to fit your potential to. In this worked example we will fit to 670 configurations, so replace the limit 1-362 on the third line with 1-322 and change the number on the first line from 6 to 2 (the number of vasprun.xml files to read-in). The resulting 'fitdbse' file should look like Input/EAM/fitdbse.save, except that in the latter, the 'un-used' configurations in this file have been moved below a header "!Testing set:" (this will not affect the present run, and is only for convenience later on).

Run *MEAMfit* again to generate the 'settings' file. Uncomment the CUTOFF_MAX line (remove the #) and specify a maximum cutoff radius of 4.4. In addition, add the line "STOPTIME=24", which specifies the approximate length of time (in hours) for which the program should run for. Check that the 'settings' file is the same as: Input/EAM/settings.save.

Run *MEAMfit* again, this time to perform an optimization. Once completed, check the 'bestoptfuncs' file. This file contains the ten smallest values of the optimization function, *R* (Equation 9 of the article) as evaluated using the fitting-set (henceforth we shall denote this as R^{fit}). The corresponding potentials are stored in potparas_best# (where # runs from 1-10). Check that the smallest R^{fit} is ~ 0.12.

5.3) Check EAM potential using testing-set

Enter the directory Input/EAM/Test and copy over your best performing potential from your EAM optimization:

```
Input/EAM/Test$ cp ../potparas_best1 potparas_EAM
```

Compare the 'fitdbse' to that in the previous directory and notice that it has been re-organised so that *MEAMfit* will now read in those vasprun.xml files that were not used in the fit (although the 1st configuration of the vasprun1.xml file *has* been included to ensure that the same reference structure is used in the calculation of the optimization function as was used during the optimization).

Note also that the 'settings' file in this directory is set up to verify, rather than optimize a potential through the tag "NOOPT=true". The filename containing the potential to be verified, 'potparas_EAM', is supplied by the "POTFILEIN=potparas_EAM" tag. Run *MEAMfit*. The optimization function will be printed to standard output in the line: "Optimization function=". Check that this value is in the range R^{test} = 0.12-0.13. This indicates comparable performance of the potential in the testing-set as in the fitting-set.

5.4) MEAM optimization

The MEAM fitting capability of *MEAMfit* is now demonstrated. Enter the directory Input/MEAM/. Notice that the settings file here differs from that in EAM/ only by the line 'TYPE=MEAM'.

Run MEAMfit. Once completed, check the 'bestoptfuncs' file and note the smallest optimization function, R^{fit}. Check that $R^{fit} < 0.12$. The optimal value for this optimization function is $R^{fit} \sim 0.06$, although it is unlikely that your value will be this low. To reach this optimal value one should allow the optimization to run for longer or use multiple instances of *MEAMfit*. Our test calculations (using an Intel Fotran compiled version of the code) found that running *MEAMfit* for 24 hours across 10 parallel instances provided a 90% chance of reaching $R^{fit} \sim 0.06$-0.07.

5.5) Check MEAM potential using testing-set

As with the EAM potential, we can now test the optimized MEAM potential by evaluating the optimization function for a 'testing-set' of energies. Enter the directory Input/MEAM/Test and copy over your best performing potential from your MEAM optimization:

```
Input/MEAM/Test$ cp ../potparas_best1 potparas_MEAM
```

Run *MEAMfit* and search for the line "Optimization function=" in the standard output. Typically for the MEAM, one will find $R^{test} \approx R^{fit} + 0.015$.

5.6) Force-fitting

So far the potentials optimized in this worked example have been fit solely to DFT energies. In the present sub-section the force-fitting capability of *MEAMfit* is demonstrated. Enter Input/EAM/ForceFit. Check the 'fitdbse' file and notice the 'f' in place of the 'E' on the second line. This specifies that the forces are to be fit for the configurations 1-5. In addition, the 'vasprun1.xml' file has been altered so that for the first configuration only the first and 33rd forces are to be fit to ('t' symbols have been added to the corresponding force-lines in this file – see section 4.1 'Fitting selected forces' for more details). The total number of force-components specified by this 'fitdbse' file is 774.

Before commencing a force-fit, let us see how well your previously fitted EAM potential already reproduces the forces. First copy over your potential:

```
TestCalcs/EAM/ForceFit$ cp ../potparas_best1 potparas_EAM
```

Now run *MEAMfit*, either using the -noopt and -i potparas_EAM flags or by temporarily adding "NOOPT=true" and "POTFILEIN=potparas_EAM" to the 'settings' file. The optimization function ("Optimization function=" in the standard output) should be in the range 0.15-0.18, indicating force-components errors within 15%-18% of their thermal fluctuations. This is already a nice agreement, given that none of these forces were included in your original fit.

Now, to commence a force-fit run *MEAMfit* as usual (remove the "POTFILEIN= potparas_EAM" and "NOOPT=true" lines from the 'settings' file if you added these previously). Once completed, one should find R^{fit} ~ 0.14-0.15 from the 'bestoptfuncs' file.

5.7) Check force-fitted potential against energy data

One can now test how well the force-fitted potential reproduces the energies. Enter the Input/EAM/ForceFit/Test/ and copy over your best performing potential from your MEAM optimization:

```
Input/EAM/ForceFit/Test$ cp ../potparas_best1 potparas_EAM
```

Run *MEAMfit* and note the value of the optimization function (search for the line "Optimization function=" in the standard output). The optimization function may be larger than that achieved with a pure energy fit: A rigid shift of the energies may be necessary to bring the energies in line with the DFT energies. To achieve this enter the Input/ForceFit/Test/ApplyShift directory and copy over your potential:

```
Input/ForceFit/Test/ApplyShift$ cp ../potparas_EAM ./
```

Upon completion (this part is very fast), you should find R^{test} = 0.13-0.15 (search for the line "Optimization function= …" in the standard output).

5.8) Testing potentials with LAMMPS

The files of the form ZrC.eam.alloy produced in your EAM run are compatible with LAMMPS. In this section we demonstrate that the energies produced in LAMMPS using the ZrC.eam.alloy potential are identical to those produced by *MEAMfit*.

Enter Input/EAM/LAMMPStest and copy over your EAM potential (both in *MEAMfit* and LAMMPS format):

```
Input/EAM/LAMMPStest$ cp ../potparas_best1 potparas_EAM
Input/EAM/LAMMPStest$ cp ../ZrC.eam.alloy_1 ZrC.eam.alloy
```

Now run *MEAMfit*. The 'fitdbse' file for this run contains only one configuration: take a note of the energy of this configuration as produced by *MEAMfit* (search for 'Energies:' in the standard output, and read off the 'fitdata' value given below). This value is equal to the energy of the second configuration of vasprun4.xml (the first configuration is not used because it corresponds to a perfect crystal), in particular it is equal to the a*bsolute* energy since "USEREF=false" has been specified in the settings file.

Now use LAMMPS to get the same value using the in_ZrC.eam input file. This will perform a static calculation and produce a total energy. Check that this energy is the same as that produced by *MEAMfit*.

5.9) Testing potentials with Camelion

To test your MEAM potential with Camelion, enter Input/MEAM/Test/CamelionTest and copy over your MEAM potential (both in *MEAMfit* and Camelion formats):

```
Input/MEAM/Test/CamelionTest/$ cp ../potparas_MEAM ./
Input/MEAM/Test/CamelionTest/$ cp ../ZrC_MEAM* ./
```

Now run *MEAMfit*. A single absolute energy will be produced as for the LAMMPS test, corresponding to configuration 2 of the vasprun4.xml file. Now we would like to check that Camelion produces the same energy. **First edit the goion_ZrC file by substituting in the lines provided at the end of the previous *MEAMfit* run.** To run Camelion, first ensure it is installed and then:

```
Input/EAM/MEAM/Test/CamelionTest/$ cp cn2.eam cn0.0
Input/EAM/MEAM/Test/CamelionTest/$ ./goion_ZrC 0 0 0 0
```

In the standard output, search for "** START OF DYNAMICS **", and then locate the sixth line down from this match. The energy is written out in the third value of this line in units of 0.0872eV per atom. To convert to eV per supercell, multiply by 64*0.0872, and check that the value is the same as that produced by *MEAMfit*.

6. Settings

This section contains a full list of the tags recognized in the settings file, including more detailed descriptions of some of those parameters already introduced:

TYPE: (no default, must be specified) controls the type of potential to be optimized, of which there are two choices: EAM or MEAM.

OPTFUNCCG: (default 10) defines the threshold for conjugate-gradient minimization during the random-sampling phase. Once the optimization function for a randomly generated potential falls below this value, a full conjugate-gradient minimization for this potential will commence. As a rule of thumb, this should be adjusted so that *MEAMfit* takes approximately 5-10 seconds to find a potential suitable for conjugate-gradient minimization.

CUTOFF_MAX: (no default, must be specified) specifies the maximum value that any of the cut-off radii in the pairwise functions are allowed to take, and therefore controls the maximum range of the potential (units: Å). This is where you choose the number of nearest neighbour interactions to be taken into account by the potential (1^{st}, 2^{nd}, 3^{rd} nearest neighbour, etc). To help choose the value, *MEAMfit* produces a file 'sepnHistogram.out', containing the distribution of interatomic separations found from the POSCAR files specified in 'fitdbse'. From here one can see what an appropriate value for the value of rmax should be, given the number of nearest neighbours one wishes to include. For more than one species, this file will have multiple columns for separations between each type of species, e.g., for two species, three additional columns will be produced for the species pairs: (1,1); (1,2); and (2,2).

CUTOFF_MIN: (default 1.5 Å) specifies the minimum value that any of the cut-off radii in the pairwise functions are allowed to take (units: Å). This cannot be smaller than the default value, 1.5 Å (Note: pair-potentials below this separation are matched to Ziegler-Biersack-

Littmark functions, which are fully specified by the atomic numbers read in from the vasprun.xml input file/s).

STOPTIME (default = 168 hrs = 1 week): (approximately) the length of time *MEAMfit* will be allowed to run for. It will usually run for slightly longer, as this stop condition is only evaluated once a given CG optimization is completed.

NOPTFUNCSTORE (default = 10): The number of potentials generated in the initial random-sampling phase of the algorithm and retained during the genetic-algorithm phase. This should be adjusted to find the optimal balance between maintaining diversity between potentials and thus avoiding local minima (achieved by using larger values) and reducing the time taken to complete the optimization (smaller values).

NOOPT (default=F): When set to T, no optimization will be performed. This is useful in conjunction with POTFILEIN

CONT (default=F): When set to T, specifies that a continuation job should be performed. Such a job can be performed on a previous job provided that a full set of potparas_best files are available (I.e.., NOPTFUNCSTORE such files) along with the corresponding bestoptfuncs file. These files will be over-written once *MEAMfit* is run, so it is best to copy these files (along with the settings, fitdbse and vasprun files) into a new directory, and run the job there. Note: NOPTFUNCSTORE in the continuation job should be set the same as in the original job. *Alternatively use the flag: -cont*.

VERBOSE (default=F): controls the amount of output provided by the fitting code, and should be set either to T or F.

SEED (default=0): controls the seed used to generate the random numbers. Any finite positive value will be used directly as a seed by the random number generator, whereas a value of zero will result in the system clock being used to set up the initial seed.

OPTFUNC_ERR (default, for pure energy optimization=10^{-10}, for force/mixed energy-force optimizations=10^{-4}): tells the CG-optimizer the order of error it can expect in the calculated optimization function. This will help determine the termination condition for the CG-optimizer, and should be set smaller for force-fitting (due to the larger error in the computed forces – on account of the finite difference approach).

POTFILEIN (no default; cmd-line: -in <filename>): if this is specified, it should be set equal to the name of the file containing a potential. The contents of this file should be in the MEAMfit format (the format used by the files potparas_best#. For more details on how to use this command to read-in potentials for further optimization refer to Section 4.2, and for details on how to use this command to analyse a potential refer to Section 7.2.

POTFILEOUT (no default): if this is specified, it should be set equal to the name of a file, to which a sample potential file will be written. This can be useful for better understanding the format of the MEAMfit potential files.

FIXPOTIN (default: false): if an input potential file is to be read in, this parameter specifies whether the parameters supplied by this file are to be held fixed (=true) or allowed to

optimize (=false). The former is useful for, e.g., fitting a cross-potentials between two species of atoms, whilst keeping the single species parts of the potential equal to previously optimized potentials. The latter (the default setting) is useful when a potential is to be used as a starting point for further optimization (e.g., EAM to MEAM).

PARASTOOPT: use this tag to specify manually which parameters are to be read-in from a file/randomly generate; and which parameters are to be held fixed and which are allowed to varied over the course of the optimization. This is an advanced setting and normally should not be used. To use this parameter it is advised to set PARASTOOPT=Write. Upon running the code will then print out an example use of this tag. A set of values are presented, to be copied into the settings value. These values are in the same order as the parameters presented in the potparas_best# files, and for each parameter one can specify a value from 0-3:

0 – Read-in starting value from file (or set to zero if no file provided) and do not allow this parameter to be optimized.

1 – As above, but allow the parameter to be optimized.

2 – Randomly initialize the parameter, and allow the parameter to be optimized.

3 – As above, but do not allow the parameter to be optimized.

Note: this parameter cannot be used in conjunction with NTERMS nor with NTERMS_EMB (NTERMS and NTERMS_EMB are recommended unless a very specific application of *MEAMfit* is envisaged).

7. Analysis of potentials

7.1) Visualising potentials:

Once a potential has been fitted, it can be visualized using the 'plotpot.gp' gnuplot script. The pair-potential terms and atomic densities will be plotted side-by-side with a histogram showing the distribution of atomic separations (e.g. Figure ...). In addition, vertical lines are placed at the cutoff radii for each pairwise function.

7.2) Evaluating optimization functions in testing-set:

An optimized potential, should be tested by checking it's performance against an independent set of data which was not used in the fit (a so-called 'testing-set'). The 'fitdbse' file should be adjusted accordingly, with *MEAMfit* then run using the command MEAMfit –no –i *filename* (where *filename* is the potparas_best file you would like to test). Alternatively NOPT=TRUE and POTFILEIN=*filename* can be added to the settings file.

8. Using potential-files

8.1) Introduction

Potentials produced by *MEAMfit* are written in three different formats: *MEAMfit* format; LAMMPS format and Camelion format. The *MEAMfit* format contains all of the parameters used to specify the analytic form of the potential (those appearing in Eqns. 1-8 of the *MEAMfit* article). The LAMMPS and Camelion formats however contain radial- and density-grids, where the various pairwise functions and embedding function/s are evaluated across a range of values of the interatomic separation and the background-density respectively.

8.2) *MEAMfit* format

The *MEAMfit* format is that produced and read by the code itself, and is the format used in the 'potparas_best#' files (where '#' denotes a number). These files are produced as the code runs, and store the best performing potentials at any given point. By default, the top ten potentials are stored (with corresponding values of the optimization functions given in the 'bestoptfuncs' file). Caution: as the code gradually finds better potentials, these files will be overwritten.

'potparas_best#' files can be used either as further input to the code (by means of the –i flag, or the POTFILEIN= tag, see Section 6 for further details) either to test the quality of a potential in a testing set (see further Section 7.2) or to use for the purposes of further optimization (see further Section 4.2).

8.3) LAMMPS format (EAM potentials)

Potentials are written in the LAMMPS format to files of the form XY.eam.alloy_# (for binaries), which are the equivalents of the potparas_best# files but in LAMMPS format. Here, X and Y are the species of atoms appearing in the fit (for a unary the files will be named X.eam.alloy, for a ternary, XYZ.eam.alloy, etc). These files are ready for immediate use within the LAMMPS code (atomic numbers and atomic masses, which are a required part of this format, are automatically added to these files by *MEAMfit* so that no editing on the part of the user is required). LAMMPS files can also be generated by running *MEAMfit* in no-optimization mode using a potparas_best# file as an input (add "POTFILEIN= potparas_best1" and "NOOPT=true" to the 'settings' file for example). In this case a file of the form XY.eam.alloy (with no _#) will be produced.

8.4) Camelion format (MEAM potentials)

LAMMPS does not currently support reference-free MEAM potentials (only the original formulation of Baskes, and its 2nd nearest-neighbor generalisation are supported), and so compatibility with an alternative free to use molecular-dynamics software, Camelion, is provided [Thijsse, 2009].

To use your fitted potential in Camelion, enter MEAMfit –i potparas_best1 –no (or add "POTFILEIN=potparas_best1" and "NOOPT=true" to your settings file and run *MEAMfit* as usual). Here, potparas_best1 should be the name of the potential file you want to use in Camelion. This will produce files of the form XY_MEAM… which contain the pair-potentials, electron-densities and embedding functions of the potential in a radial grid format (X and Y are the species). *MEAMfit* will also write out a few lines which need to be substituted into

the ‘goion’ file (another file used by Camelion, a template for which is provided in MEAMfit/camelionInputs).

Using these files, as well as a suitable input file specifying the starting positons of the atoms (an example is provided: MEAMfit/camelionInputs/cn0.0), one can run Camelion by directly running the goion script (e.g.: goion 0 0 0 0 to produce a single energy for the atomic configuration specified). To perform a test to demonstrate that the energies produced by *MEAMfit* and Camelion are identical, please see Section 5. For further details on using Camelion please contact the author and download the full code [Thijsse, 2009].

9. Contact

Please contact me (A.I.Duff.99@cantab.net) if you have any suggestions for improving the code.

10. References

[Baskes, 1999] M. I. Baskes, Mater. Sci. Eng. A **261** 165

[Timonova, 2011] M. Timonova and B. J. Thijsse, Mod. Simul. Mat. Sci. Eng. **19** 015003 (2011)

[Thijsse, 2009] B. Thijsse, Camelion molecular dynamics software: https://dl.dropboxusercontent.com/u/54720544/Molecular%20Dynamics%20pack/Aug%202011%20v%2015Feb2013%20%2B%20Next%20Release.zip

Appendix A: the *MEAMfit* potential format

The ‘potparas_best’ files produced and read-in by *MEAMfit* contains all of the parameters necessary to fully specify a potential. The precise formatting will depend upon the total number of different types of atomic species to be used in the fit. For a single species of atom it will have the following form:

```
# cmin-cmax values, used for the screening terms in an MEAM potential
0.000000000000000E+000
0.000000000000000E+000
T # meam_tau values, used to define relative weights of angular terms in an MEAM potential
0.000000000000000E+000
0.000000000000000E+000
F
1 # Electron-density. Only first four lines are used for EAM (other lines are for l>0 MEAM-terms)
-7.31461506361180        2.80000000000000        -0.343018860189889
3.60000000000000        0.289666686098838        4.20000000000000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
```

```
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
1 # Embedding function
3.31245742276721
6.055778680792733E-002
7.651490560479200E-008
0.000000000000000E+000
1 # Pair-potential Note: need to change from 2 to 1 in code
1.97039738425846        2.80000000000000        -0.685336328799219
3.20000000000000        0.356367723929941        3.80000000000000
-6.935520711404397E-002   4.20000000000000        0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000
0.000000000000000E+000
-2.17447721699808 # Fixed energy constant counted, per atom, towards the total energy
```

For a pure pair-potential, only the yellow-highlighted values are relevant. The first yellow-highlighted line contains the number ‘1’, denoting that the pair-potential should consist of a sum over cubic terms (currently the only form implemented):

$$V^{p.p}_{\alpha,\alpha'}(r) = \sum_{n=1}^{16} a^n_{\alpha,\alpha'}(r - b^n_{\alpha,\alpha'})^3\theta(r - b^n_{\alpha,\alpha'})$$

where the α_i is the species of atom *i* and where $\theta(r - b^n_{\alpha,\alpha'})$ is a cut-off function; one for $r < b^n_{\alpha,\alpha'}$ and zero for $r > b^n_{\alpha,\alpha'}$. For the above ‘potparas_best’ file there is only one species and so we can drop this index.

The subsequent lines in ‘potparas_best’ correspond to pairs of a^n and b^n coefficients, so that for example a^1 = 1.970… and b^1 = 2.800… . Up to 16 pairs of coefficients (16 terms in the above expansion) can be specified in total, although if they are not all needed zeros can be put in their place. The above ‘potparas_best’ file corresponds to a pair-potential with 4 terms in the above expansion.

The green-terms should be set to non-zero values if an EAM potential is to be used. The first green-highlighted line should be given the value F (in a later update of the code, this parameter will determine whether an atomic density should also depend on the ‘acceptor’

atom). The next green-highlighted line tells *MEAMfit* to construct the atomic densities from sums over cubic terms (currently the only form implemented):

$$\rho_{\alpha,\alpha'}^{a(0)}(r) = \sum_{n=1}^{16} a_{\alpha,\alpha'}^{n}\,(r - b_{\alpha,\alpha'}^{n})^{3}\theta(r - b_{\alpha,\alpha'}^{n})$$

Up to six terms in this expansion can then be specified over the next four lines (12 parameters in total), using the same format as for the pair-potential (coefficient; cut-off; …).

For MEAM potentials, there are four more lines (highlighted in blue) for each of l=1, 2 and 3, to specify the functions $\rho_{\alpha,\alpha'}^{a(1)}(r)$, $\rho_{\alpha,\alpha'}^{a(2)}(r)$ and $\rho_{\alpha,\alpha'}^{a(3)}(r)$. In addition there lines three and four of the file specify *cmin* and *cmax* (the three-body angular range over which screening is switched on), however these have not yet been tested, so please do not use yet. Lines five and six contain the values of $\tau_i^{(l)}$ for l=0 to 3 (the l=0 value is redundant).